



リファレンス・マニュアル：プロシージャ

Adaptive Server[®] Enterprise

15.7 ESD #2

ドキュメント ID : DC36424-01-1572-01

改訂 : 2012 年 7 月

Copyright © 2012 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

Sybase の商標は、the Sybase trademarks page (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

このマニュアルに記載されている SAP、その他の SAP 製品、サービス、および関連するロゴは、ドイツおよびその他の国における SAP AG の商標または登録商標です。

Java および Java 関連の商標は、米国およびその他の国における Oracle およびその関連会社の商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

IBM および Tivoli は、International Business Machines Corporation の米国およびその他の国における登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

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

目次

第 1 章	システム・プロシージャ	1
	システム・プロシージャの概要	1
	システム・プロシージャの戻り値	2
	システム・プロシージャに対するパーミッション	2
	システム・プロシージャの監査	3
	システム・プロシージャの実行	3
	パラメータ値の入力	4
	メッセージ	5
	システム・プロシージャ・テーブル	5
	sp_activeroles	6
	sp_add_qpgroup	8
	sp_add_resource_limit	9
	sp_add_time_range	14
	sp_addalias	18
	sp_addauditrecord	20
	sp_addauditable	23
	sp_addengine	25
	sp_addexclass	27
	sp_addextendedproc	29
	sp_addexternlogin	31
	sp_addgroup	34
	sp_addlanguage	36
	sp_addlogin	39
	sp_addmessage	40
	sp_addobjectdef	43
	sp_addremotelogin	47
	sp_addsegment	51
	sp_addserver	54
	sp_addthreshold	58
	sp_addtype	63
	sp_addumpdevice	67
	sp_adduser	70
	sp_altermessage	72
	sp_audit	74
	sp_autoconnect	82

sp_autoformat	84
sp_bindcache	89
sp_bindefault.....	94
sp_bindexeclass.....	97
sp_bindmsg.....	101
sp_bindrule.....	103
sp_cacheconfig	106
sp_cachestrategy	117
sp_changedbowner.....	121
sp_changegroup	123
sp_checknames	125
sp_checkreswords	128
sp_checksouce	142
sp_chgattribute.....	145
sp_cleanpwdchecks	151
sp_clearpsexex	152
sp_clearstats	154
sp_client_addr.....	156
sp_cluster.....	158
sp_clusterlockusage.....	182
sp_cmp_all_qplans	183
sp_cmp_qplans	186
sp_commonkey	188
sp_companion.....	190
sp_compatmode.....	194
sp_config_dump.....	195
sp_configure.....	199
sp_copy_all_qplans.....	208
sp_copy_qplan	210
sp_countmetadata.....	212
sp_cursorinfo.....	215
sp_dbextend.....	218
sp_dboption.....	229
sp_dbrecovery_order	247
sp_dbremap	250
sp_defaultloc	252
sp_deletesmobj.....	255
sp_depends.....	258
sp_deviceattr.....	266
sp_diskdefault	269
sp_displayaudit	271
sp_displaylevel.....	275
sp_displaylogin.....	277
sp_displayroles	284

sp_downgrade.....	287
sp_dropalias.....	295
sp_drop_all_qplans.....	297
sp_drop_qpgroup.....	298
sp_drop_qplan.....	299
sp_drop_resource_limit.....	300
sp_drop_time_range.....	304
sp_dropdevice.....	305
sp_dropengine.....	307
sp_dropexclass.....	309
sp_dropextendedproc.....	310
sp_dropexternlogin.....	311
sp_dropglockpromote.....	313
sp_dropgroup.....	315
sp_dropkey.....	316
sp_droplanguage.....	318
sp_droplogin.....	320
sp_dropmessage.....	321
sp_dropobjectdef.....	322
sp_dropremotelogin.....	324
sp_droprowlockpromote.....	326
sp_dropsegment.....	328
sp_dropserver.....	330
sp_dropthreshold.....	332
sp_droptype.....	334
sp_dropuser.....	335
sp_dump_history.....	337
sp_dumpoptimize.....	340
sp_encryption.....	346
sp_engine.....	362
sp_errorlog.....	366
sp_estspace.....	368
sp_export_qpgroup.....	374
sp_extendsegment.....	376
sp_extengine.....	378
sp_extrapwdchecks.....	380
sp_familylock.....	381
sp_find_qplan.....	384
sp_fixindex.....	387
sp_flushstats.....	390
sp_forceonline_db.....	392
sp_forceonline_object.....	394
sp_forceonline_page.....	397
sp_foreignkey.....	399

sp_freedoml.....	401
sp_getmessage.....	403
sp_grantlogin.....	405
sp_ha_admin.....	407
sp_help.....	409
sp_help_resource_limit.....	420
sp_help_qpgroup.....	424
sp_help_qplan.....	426
sp_helpapptrace.....	428
sp_helppartition.....	429
sp_helpcache.....	434
sp_helpcomputedcolumn.....	436
sp_helpconfig.....	437
sp_helpconstraint.....	443
sp_helpdb.....	448
sp_helpdevice.....	453
sp_helpextendedproc.....	456
sp_helpexternlogin.....	458
sp_helpgroup.....	460
sp_helpindex.....	462
sp_helpjava.....	465
sp_helpjoins.....	468
sp_helpkey.....	470
sp_helplanguage.....	472
sp_helplog.....	474
sp_helpobjectdef.....	475
sp_helpremotelogin.....	477
sp_helpprotect.....	478
sp_helpsegment.....	484
sp_helpserver.....	489
sp_helpsort.....	491
sp_helpstext.....	493
sp_helpthread.....	500
sp_helpthreshold.....	502
sp_helpuser.....	504
sp_hidetext.....	507
sp_import_qpgroup.....	509
sp_indsuspect.....	511
sp_jreconfig.....	512
sp_ldapadmin.....	526
sp_listener.....	534
sp_listsuspect_db.....	538
sp_listsuspect_object.....	539
sp_listsuspect_page.....	541

sp_lmconfig	542
sp_lock	546
sp_locklogin	552
sp_logdevice	555
sp_loginconfig	558
sp_logininfo	560
sp_logiosize	562
sp_logintrigger	566
sp_maplogin	568
sp_merge_dup_inline_default	570
sp_metrics	573
sp_modify_resource_limit	575
sp_modify_time_range	579
sp_modifylogin	581
sp_modifystats	582
sp_modifythreshold	586
sp_monitor	591
sp_monitorconfig	600
sp_monitor_server	608
sp_object_stats	609
sp_objectsegment	613
sp_opt_querystats	614
sp_optgoal	617
sp_options	619
sp_passthru	629
sp_password	632
sp_passwordpolicy	633
sp_pciconfig	648
sp_placeobject	653
sp_plan_dbccdb	655
sp_poolconfig	658
sp_post_xpload	665
sp_primarykey	667
sp_processmail	669
sp_procxmode	672
sp_querysmobj	675
sp_recompile	678
sp_refit_admin	680
sp_remap	683
sp_remoteoption	684
sp_remotesql	687
sp_rename	690
sp_rename_qpgroup	693
sp_renamedb	695

sp_reportstats	698
sp_restore_system_role	700
sp_revokelgin	702
sp_role	704
sp_securityprofile	706
sp_sendmsg	710
sp_serveroption	713
sp_set_qplan	719
sp_setlangalias	721
sp_setpglockpromote	722
sp_setpsexex	726
sp_setrowlockpromote	728
sp_setsuspect_granularity	732
sp_setsuspect_threshold	736
sp_setup_table_transfer	738
sp_shmdumpconfig	739
sp_show_options	745
sp_showcontrolinfo	747
sp_showexeclass	750
sp_showoptstats	752
sp_showplan	760
sp_showpsexex	762
sp_spaceusage	764
sp_spaceused	778
sp_ssladmin	782
sp_syntax	787
sp_sysmon	789
sp_tab_suspectptn	796
sp_tempdb	797
sp_tempdb_markdrop	808
sp_thresholdaction	810
sp_tran_dumpable_status	813
sp_transactions	814
sp_unbindcache	821
sp_unbindcache_all	824
sp_unbindefault	826
sp_unbindexeclass	828
sp_unbindmsg	831
sp_unbindrule	832
sp_version	835
sp_volchanged	838
sp_webservices	842
sp_who	852
sp_xmlschema	856

第 2 章	カタログ・ストアド・プロシージャ	857
	概要	857
	オプション・パラメータの指定	859
	パターン一致	860
	システム・プロシージャ・テーブル	860
	ODBC のデータ型	860
	sp_column_privileges	862
	sp_columns	864
	sp_databases	867
	sp_datatype_info	868
	sp_fkeys	869
	sp_pkeys	871
	sp_server_info	872
	sp_special_columns	875
	sp_sproc_columns	877
	sp_statistics	879
	sp_stored_procedures	881
	sp_table_privileges	882
	sp_tables	883
第 3 章	システム拡張ストアド・プロシージャ	885
	概要	885
	システム ESP へのパーミッション	886
	システム ESP に対応する DLL	886
	システム ESP の使用	886
	xp_cmdshell	887
	xp_deletemail	890
	xp_enumgroups	891
	xp_findnextmsg	892
	xp_logevent	893
	xp_readmail	894
	xp_sendmail	897
	xp_startmail	901
	xp_stopmail	903
第 4 章	dbcc ストアド・プロシージャ	905
	概要	905
	オブジェクトの名前と日付の指定	906
	オブジェクト名の指定	907
	日付の指定	907
	sp_dbcc_alterws	908
	sp_dbcc_configreport	910
	sp_dbcc_createws	911

sp_dbcc_deletedb	913
sp_dbcc_deletehistory	915
sp_dbcc_differentialreport.....	917
sp_dbcc_evaluatedb	919
sp_dbcc_exclusions	921
sp_dbcc_faultreport.....	923
sp_dbcc_fullreport.....	929
sp_dbcc_help_fault	930
sp_dbcc_patch_finishtime	931
sp_dbcc_recommendations	933
sp_dbcc_runcheck	935
sp_dbcc_statisticsreport.....	937
sp_dbcc_summaryreport.....	941
sp_dbcc_updateconfig	945

索引	949
-----------------	------------

システム・プロシージャ

この章では、システム・テーブルからのレポート取得や、システム・テーブルの更新に使用される、Sybase® 提供のストアド・プロシージャであるシステム・プロシージャについて説明します。

この章では、次の項目について説明します。

トピック名	ページ
システム・プロシージャの概要	1
システム・プロシージャに対するパーミッション	2
システム・プロシージャの監査	3
システム・プロシージャの実行	3
パラメータ値の入力	4
メッセージ	5
システム・プロシージャ・テーブル	5

システム・プロシージャの概要

システム・プロシージャは、インストール時に `installmaster` によって作成されます。作成されたシステム・プロシージャは `sybssystemprocs` データベースに保管され、システム管理者が所有します。最後に実行された `installmaster` のバージョンを確認するには、`sp_version` を使用します。

システム・プロシージャには、特定のデータベースでしか実行できないものもありますが、システム・プロシージャの多くは、どのデータベースでも実行できます。任意のデータベースから実行できる独自のシステム・プロシージャを作成できます。詳細については、『システム管理ガイド』を参照してください。

すべてのシステム・プロシージャは、独立性レベル 1 で実行されます。

すべてのシステム・プロシージャは、リターン・ステータスをレポートします。たとえば、次の例は、プロシージャが正しく実行されたことを表します。

```
return status = 0
```

このマニュアルの例では、リターン・ステータスが省略されています。

Adaptive Server[®] バージョン 12.5.1 以降では、ストアド・プロシージャで最大 10,000 個の変数を宣言できます。それ以前のバージョンの Adaptive Server では、最大 2,000 個に限られていました。

システム・プロシージャの戻り値

ストアド・プロシージャは、プロシージャが正常に終了したかどうか、または失敗した場合はその理由を示す「リターン・ステータス」を返します。システム・プロシージャの戻り値の詳細については、『Transact-SQL ユーザーズ・ガイド』の「第 16 章 ストアド・プロシージャの使用」にある「戻り値」を参照してください。

システム・プロシージャに対するパーミッション

システム・プロシージャへのパーミッションは、`sysystemprocs` データベースに設定されます。

システム・プロシージャの中には、特定の権限または役割を持つユーザしか実行できないものがあります。システム・プロシージャのパーミッション・チェックは、細密なパーミッションの設定によって異なる場合があります。詳細については、各システム・プロシージャの「パーミッション」セクションを確認してください。細密なパーミッションの詳細については、『セキュリティ管理ガイド』の「細密なパーミッションの使用」を参照してください。

他のシステム・プロシージャ（たとえば、すべての `sp_help` プロシージャ）は、そのプロシージャに対する `execute` パーミッションが `sysystemprocs` で `public` に付与されていれば、どのユーザでも実行できます。

システム・プロシージャのユーザ・パーミッションを拒否するには、システム管理者がユーザを `sybsystemprocs..sysusers` に追加して、そのプロシージャに適用される `revoke` 文を記述する必要があります。ユーザ・データベースの所有者は、自分のデータベース内のシステム・プロシージャに対するパーミッションを直接制御できます。

システム・プロシージャの監査

一般には、監査オプション “`exec_procedure`” を有効にすると、ストアド・プロシージャの実行を監査できます。このオプションでは、ストアド・プロシージャの名前とパラメータを含む監査レコードが生成されます。

システム・プロシージャの実行

システム・プロシージャは、`sybsystemprocs` 以外のデータベースで実行されると、実行されたデータベースのシステム・テーブル上で動作します。たとえば、`pubs2` のデータベース所有者が `sp_adduser` を `pubs2` で実行すると、新しいユーザが `pubs2..sysusers` に追加されます。

特定のデータベース内でシステム・プロシージャを実行するには、次のいずれかを行います。

- そのデータベースを `use` コマンドでオープンして、プロシージャを実行する。
- プロシージャ名にデータベース名を修飾する。

たとえば、ユーザ定義のシステム・プロシージャである `sp_foo` は `db_name` システム関数を実行し、実行先のデータベース名を返します。したがって、`pubs2` データベースで実行すると、値 “`pubs2`” が返されます。

```
exec pubs2..sp_foo

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

`sybsystemprocs` で実行すると、値 “`sybsystemprocs`” が返されます。

```
exec sybsystemprocs..sp_foo
```

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

パラメータ値の入力

システム・プロシージャに対するパラメータ値に句読点や埋め込みスペースがある場合、またはそのパラメータ値が予約語である場合は、パラメータ値を一重引用符か二重引用符で囲みます。パラメータがデータベース名または所有者名で修飾されているオブジェクト名である場合は、その名前全体を一重引用符か二重引用符で囲みます。

注意 システム・プロシージャのパラメータとして区切り識別子を使用しないでください。予測できない結果を生じる可能性があります。

プロシージャにオプションのパラメータが複数ある場合、すべてのパラメータを指定する代わりに次の形式でパラメータを指定できます。

```
@parametername = value
```

構文内のパラメータ名は、プロシージャによって定義されるパラメータ名に対応します。

たとえば、`sp_addlogin` の構文は、次のようになります。

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

`sp_addlogin` を使用して “susan” のログインを作成する場合、“wonderful” というパスワードと “Susan B. Anthony” というフルネームで、サーバのデフォルトのデータベースと言語を使用するには、次のようにします。

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

これは、すべてのパラメータを指定した次のようなコマンドと同じ情報を指定したことになります。

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

また、プレースホルダとして “null” も使用できます。

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

“null” は引用符で囲まないでください。

1 行内に指定できる語数や改行位置についての SQL の規則はありません。システム・プロシージャの後に続けてコマンドを発行すると、Adaptive Server は、システム・プロシージャを実行しようとし、次にコマンドを実行しようとしています。たとえば、次のコマンドを実行した場合、Adaptive Server は `sp_help` の出力を返して `checkpoint` コマンドを実行します。

```
sp_help checkpoint
```

システム・プロシージャが想定している数を上回るパラメータを指定すると、超過したパラメータは無視されます。

メッセージ

システム・プロシージャは、情報メッセージとエラー・メッセージを返します。これらのメッセージは、このマニュアルの各プロシージャの項で説明されています。システム・プロシージャのエラー・メッセージは、エラー番号 17000 から始まります。

プロシージャに組み込まれた関数やコマンドが生成するエラー・メッセージについては、『トラブルシューティングおよびエラー・メッセージ・ガイド』で説明しています。

システム・プロシージャ・テーブル

内部システム値 (ステータス・ビットなど) を判読できるフォーマットに変換するために、`spt_values`、`spt_committab`、`spt_monitor`、`spt_limit_types` など、master データベース内の複数の「システム・プロシージャ・テーブル」がシステム・プロシージャによって使用されます。

`spt_values` が更新されることはありません。この使用方法を知るには、`sp_helptext` を実行して、このテーブルを参照するシステム・プロシージャのいずれかのテキストを調べてください。

また、テンポラリ・テーブルを作成して削除するシステム・プロシージャもあります。

sp_activeroles

説明	すべてのアクティブな役割を表示します。
構文	sp_activeroles [expand_down]
パラメータ	<p>expand_down 自分の役割が含まれたすべてのアクティブな役割の階層ツリーを表示します。</p>
例	<p>例 1 すべてのアクティブな役割を表示します。</p> <pre> sp_activeroles Role Name ----- sa_role sso_role oper_role replication_role </pre> <p>例 2 アクティブな役割とその階層ツリーを表示します。</p> <pre> sp_activeroles expand_down Role Name Parent Role Name Level ----- sa_role NULL 1 doctor_role NULL 1 oper_role NULL 1 </pre>
使用法	sp_activeroles では、自分が持つすべてのアクティブな役割と、それらの役割に含まれるすべての役割が表示されます。
パーミッション	すべてのユーザが sp_activeroles を実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。
監査	sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル 役割の作成、管理、使用の各方法については、『システム管理ガイド』を参照してください。

コマンド [alter role](#), [create role](#), [drop role](#), [grant](#), [revoke](#), [set](#)

関数 [mut_excl_roles](#), [proc_role](#), [role_contain](#), [role_name](#)

システム・プロシージャ [sp_displayroles](#)

sp_add_qpgroup

説明	抽象プラン・グループを追加します。
構文	<code>sp_add_qpgroup new_name</code>
パラメータ	<code>new_name</code> 新しい抽象プラン・グループの名前です。グループ名は、有効な識別子でなければなりません。
例	新しい抽象プラン・グループ <code>dev_plans</code> を作成します。 <code>sp_add_qpgroup dev_plans</code>
使用法	<ul style="list-style-type: none"> <code>sp_add_qpgroup</code> は、抽象プランの取得または作成に使用する抽象プラン・グループを追加するときに使用します。抽象プラン・グループを実際に設定してから、グループにプランを作成、保存、またはコピーしてください。 <code>sp_add_qpgroup</code> はトランザクション内では実行できません。
パーミッション	<code>sp_add_qpgroup</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> を持つユーザであることが必要。
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照	<p>コマンド set</p> <p>システム・プロシージャ sp_help_qpgroup</p>
----	--

sp_add_resource_limit

説明	Adaptive Server ログインまたはアプリケーションがクエリ、クエリ・バッチ、トランザクションの実行に使用できるサーバ・リソース数の制限値を作成します。
構文	<code>sp_add_resource_limit name, appname, rangename, limittype, limitvalue [, enforced [, action [, scope]]]</code>
パラメータ	<p><i>name</i> 制限が適用される Adaptive Server ログインです。 <i>name</i> か <i>appname</i> またはその両方を指定してください。特定のアプリケーションのすべてのユーザに適用される制限値を作成するには、 <i>name</i> に NULL を指定します。</p> <p><i>appname</i> 制限が適用されるアプリケーションの名前です。 <i>name</i> か <i>appname</i> またはその両方を指定してください。 Adaptive Server ログインが使用するすべてのアプリケーションに適用する制限値を作成するには、 <i>appname</i> に null を指定します。特定のアプリケーションに適用する制限値を作成するには、クライアント・プログラムがログイン・パケット内で Adaptive Server に渡すアプリケーション名を指定します。</p> <p><i>rangename</i> 制限が実施される時間範囲です。時間範囲は、制限を作成するときに、master データベースの <code>sys timeranges</code> システム・テーブルに存在していなければなりません。</p> <p><i>limittype</i> 制限を適用するリソースのタイプです。これは、次のいずれかでなければなりません。</p>

制限タイプ	説明
row_count	クエリが返せるローの数を制限する。
elapsed_time	クエリ・バッチやトランザクションが実行できる秒数をおおよその経過時間で制限する。
io_cost	クエリを処理するための実際のコストまたはオプティマイザのコスト見積もりのいずれかを制限する。
tempdb_space	tempdb データベースが単一セッション間に保持できるページの数を制限する。

limitvalue

Adaptive Server が制限を実施する前に、ログインやアプリケーションが使用できるサーバ・リソース (I/O コスト、経過時間、ローの数、または tempdb 領域) の最大リソース値です。制限値は正の数で、0 以外の 2^{31} 以下の整数とします。次の表に、制限タイプ別の制限値を示します。

制限タイプ	制限値
row_count	制限が実施される前に、クエリが返せるローの最大数。
elapsed_time	制限が実施される前に、クエリ・バッチまたはトランザクションが返せる秒数 (およそその経過時間)。
io_cost	オプティマイザのコスト計算式から計算した単位のない値。
tempdb_space	セッションごとの tempdb の使用ページ数。

enforced

クエリの実行前、または実行中に制限を実施するかどうかを指定します。次の表に、制限タイプ別の有効値を示します。

enforced コード	説明	制限タイプ
1	実行時の見積もり I/O コストが指定制限値を超えると、動作 (action) を実行する。	io_cost
2	実際のロー数、経過時間、実行時の I/O コストが指定制限値を超えると、動作 (action) を実行する。	row_count elapsed_time io_cost
3	見積もりコストまたは実際のコストのいずれかが指定制限値を超えると、動作 (action) を実行する。	io_cost

enforced 値に 3 を指定すると、1 と 2 の論理 "or" が実行されます。たとえば、*enforced* が 3 に設定されている場合、*io_cost* が見積もりコストを超過するクエリを実行すると、指定した *action* が実行されます。クエリが見積もりコスト用に指定されている制限内であっても実際のコストを超過すれば、指定した *action* が実行されます。

enforced 値を指定しないと、*row_count* と *elapsed_time* に制限 2 が実施され、*io_cost* には制限 3 が実施されます。つまり、制限タイプが *io_cost* の場合、クエリが見積もりコストまたは実際のコストのどちらを超過しても、指定した *action* が実行されます。

action

制限値を超えたときに実行される動作です。次の表に、すべての制限タイプに有効な動作 (action) コードを示します。

action コード	説明
1	警告を出す。
2	クエリ・バッチをアボートする。
3	トランザクションをアボートする。
4	セッションを強制終了する。

action の値を指定しなければ、Adaptive Server はデフォルト値の 2 を使用します (クエリ・バッチのアボート)。

scope

制限が適用されるスコープです。制限タイプに応じて、次のコードのいずれかを指定します。

scope コード	説明	制限タイプ
1	クエリ	io_cost row_count
2	クエリ・バッチ (クライアントがサーバに送信する 1 つ以上の SQL 文)	elapsed_time
4	トランザクション	elapsed_time
6	クエリ・バッチとトランザクション	elapsed_time

scope の値を指定しなければ、制限タイプのあらゆるスコープが対象になります。

例

例 1 時間範囲 `early_morning` 中に、`payroll` アプリケーションのすべてのユーザに適用するリソース制限値を作成します。クエリ・バッチの実行が 120 秒を超えると、Adaptive Server から警告が出されます。

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

例 2 “`joe_user`” で実行されるすべてのアドホック・クエリとアプリケーションに対して、時間範囲 `midday` 中に適用されるリソース制限値が作成されます。クエリが返すローの数が 5000 を超えると、Adaptive Server はトランザクションをアボートします。

```
sp_add_resource_limit joe_user, NULL, midday, row_count, 5000, 2, 3, 1
```

例 3 “`joe_user`” で実行されるすべてのアドホック・クエリとアプリケーションに対して、時間範囲 `midday` 中に適用されるリソース制限値が作成されます。I/O コストが 650 を超えるとオペティマイザが見積もった場合、Adaptive Server はトランザクションをアボートします。

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

使用法

- リソース制限を適用するには、`sp_configure "allow resource limits"` を有効にしてください。
- 時間範囲が重複しなければ、指定のユーザ、アプリケーション、制限タイプ、スコープ、実施時間に複数のリソース制限値を設定できます。
- ログインかアプリケーション名またはその両方を対象にした、現在アクティブな名前付き時間範囲と “at all times” 時間範囲の制限はすべて、ログイン時にユーザのセッションにバインドされません。したがって、ユーザが特定のアプリケーションとは関係なく Adaptive Server にログインした場合、アプリケーションを併用するユーザを制約するためのリソース制限は適用されません。ユーザに確実に制限を適用するには、そのユーザに限定したリソース制限値を、アプリケーションとは別に作成します。
- リソース制限値はユーザ・ログイン名またはアプリケーション名のどちらか、あるいは両方によって識別されるため、Adaptive Server は事前に定義されている検索優先度に従って `sysresourcelimits` テーブルをスキャンし、該当する制限値を探します。次の表では、ログイン名とアプリケーション名のペアの優先度を、一致する順序ごとに説明しています。

レベル	ログイン名	アプリケーション名
1	“joe_user”	payroll
2	NULL	payroll
3	“joe_user”	NULL

ある優先度レベルで一致するものが 1 つ以上ある場合は、それ以降のレベルは検索されません。これによって、異なるログイン／アプリケーションの組み合わせに対し、同様の制限値が適用されないようにします。

どのレベルでも一致が見つからない場合は、そのセッションでは制限は施行されません。

- リソース制限値を追加、削除、変更すると、そのセッションで次にクエリ・バッチが開始されるときに、ログインかアプリケーションまたはその両方の制限値が再バインドされます。
- 現在アクティブな時間範囲を変更すると、そのセッションの制限値が再バインドされます。この再バインドは次にクエリ・バッチが開始されたときに行われます。
- 特定のログイン、アプリケーション、またはログインとアプリケーションの組み合わせの制限値は、重複する複数の名前付き時間範囲に対応させることはできません（同じ時間範囲を共有する制限値を除く）。

たとえば、ユーザが、午前 9 時から午後 1 時の間に 50 ローを検索するように制限されている場合、同じユーザに別の制限値を作成して、午前 10 時から正午の間に 100 ローを検索するように制限することはできません。ただし、100 ローの制限を午前 10 時から正午の間でユーザに設定し、50 ローの制限を、午前 9 時から午後 1 時の間に *アプリケーション* (たとえば、*isql*) に割り当てると、リソース階層を作成できます。

注意 時間制限の制限値に達すると、*Adaptive Server* は現在のトランザクションを終了しますが、別の *SQL* コマンドまたはバッチを実行するまで 1105 エラー・メッセージは表示されません。このメッセージは接続を再び試みたときに初めて表示されます。

パーミッション

sp_add_resource_limit のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <i>manage resource limit</i> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <i>sa_role</i> が付与されたユーザであることが必要。

監査

sysaudits テーブルの *event* カラムと *extrainfo* カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<i>exec_procedure</i>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <i>set proxy</i> が有効な場合は元のログイン名

参照

マニュアル リソースの制限値の詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_configure](#), [sp_drop_resource_limit](#), [sp_help_resource_limit](#), [sp_modify_resource_limit](#)

ユーティリティ *isql*

sp_add_time_range

説明	名前付き時間範囲を Adaptive Server に追加します。
構文	<code>sp_add_time_range name, startday, endday, starttime, endtime</code>
パラメータ	<p><i>name</i> 時間範囲の名前です。時間範囲名は最長 255 文字まで指定できます。master データベースの <code>systemranges</code> システム・テーブルにすでに存在する名前は指定できません。</p> <p><i>startday</i> 時間範囲の開始曜日です。master データベースの <code>syslanguages</code> システム・テーブルに保管されているとおりに、デフォルト・サーバ言語の曜日名をフルスペルで指定してください。</p> <p><i>endday</i> 時間範囲の終了曜日です。master データベースの <code>syslanguages</code> システム・テーブルに保管されているとおりに、デフォルト・サーバ言語の曜日名をフルスペルで指定してください。<i>endday</i> には <i>startday</i> に指定した曜日より前か後の曜日、または <i>startday</i> に指定した曜日と同じ曜日を指定できます。</p> <p><i>starttime</i> 時間範囲の開始時刻です。<i>starttime</i> は 24 時間形式で指定します。値として、“00:00” (真夜中) ~ “23:59” (午後 11:59) の間の時刻を指定できます。次のフォーマットを使用します。 "HH:MM"</p> <p><i>endtime</i> 時間範囲の終了時刻です。<i>endtime</i> は 24 時間形式で指定します。指定できる値は、“00:00” (真夜中) ~ “23:59” (午後 11:59) の間の値です。次のフォーマットを使用します。 "HH:MM"</p>

注意 全日にわたる時間範囲を作成するには、開始時間と終了時間の両方に 00:00 を指定します。

endtime には、*starttime* に指定した時刻より後の時刻を指定します (*endtime* に “00:00” を指定する場合を除く)。

例 1 月曜日から金曜日の午前 9 時から午後 5 時までをアクティブとする時間範囲 `business_hours` を作成します。

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


例 2 月曜日から金曜日の営業時間外を範囲とする 2 つの時間範囲 `before_hours` と `after_hours` を作成します。時間範囲 `before_hours` には、月曜日から金曜日の真夜中の 12 時から午前 9 時までの時間を設定します。時間範囲 `after_hours` には、月曜日から金曜日の午後 6 時から真夜中の 12 時までの値を設定します。

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

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

例 3 土曜日の真夜中の 12 時から日曜日の真夜中の 12 時までを範囲とする時間範囲 `weekends` を作成します。

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

例 4 金、土、日、月の午前 9 時から午後 5 時までを範囲とする時間範囲 `Fri_thru_Mon` を作成します。

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

例 5 水曜日の午後 5 時から真夜中の 12 時までを範囲とする時間範囲 `Wednesday_night` を作成します。

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

使用法

- Adaptive Server には、“at all times” という名前付き時間範囲が 1 つ組み込まれています。この時間範囲は、週の最初の曜日から最後の曜日まで、また 00:00 から 23:59 までのすべての時間に対応します。変更や削除はできません。
- 各名前付き時間範囲に対してユニークな ID 番号を割り当て、各 ID 番号を `systemranges` システム・テーブルに挿入します。
- 時間範囲を `systemranges` システム・テーブルに保管するとき、Adaptive Server は `startday` と `endday` の値を整数に変換します。us_english のデフォルト言語を使用しているサーバでは、週の初めは Monday (第 1 曜日) であり、週の終わりは Sunday (第 7 曜日) です。
- 他の複数の時間範囲と重複する時間範囲も作成できます。
- 範囲の曜日は隣接しているため、複数の週日で、週の初めに週の終りをラップ (隣接するように移動) させることができます。つまり、日曜日 (第 7 曜日) と月曜日 (第 1 曜日) は、火曜日と水曜日と同じように隣接して処理されます。

- アクティブな時間範囲は、各クエリ・バッチの開始時に、1つのセッションにバインドされます。実時間の変更が原因でサーバのアクティブな時間範囲の値が変更されても、クエリ・バッチの処理中には、セッションへの影響はありません。つまり、特定の時間範囲中にリソース制限値がクエリ・バッチを制限したとしても、クエリ・バッチが始まってからその時間範囲がアクティブになっていた場合、すでに実行中のクエリ・バッチはリソース制限値の影響を受けません。
- システム・プロシージャによって時間範囲が追加、変更、削除されても、現在進行中のセッションに対応するアクティブな時間範囲値が影響を受けることはありません。
- リソース制限でトランザクションがそのスコープとなっている場合、トランザクション実行中にサーバのアクティブな時間範囲の変更が発生しても、現在進行中のトランザクションは新しいアクティブ時間範囲の影響を受けません。
- トランザクションがそのスコープとなっているリソース制限に変更があっても、現在進行中のトランザクションに影響はありません。
- 時間範囲の詳細については、『システム管理ガイド』を参照してください。

パーミッション

sp_add_time_range のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage resource limit 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sso_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_add_resource_limit](#), [sp_drop_time_range](#), [sp_modify_time_range](#)

sp_addalias

説明	Adaptive Server のユーザをデータベース内で別のユーザとして認識されるようにします。
構文	<code>sp_addalias loginame, name_in_db</code>
パラメータ	<p><i>loginame</i> 現在のデータベースに代替 ID を作成するユーザの <code>master.dbo.syslogins</code> 名です。</p> <p><i>name_in_db</i> <i>loginame</i> のエイリアスとして使用するデータベース・ユーザ名です。この名前は、<code>master.dbo.syslogins</code> と現在のデータベースの <code>sysusers</code> テーブルの両方に存在していなければなりません。</p>
例	データベースの <code>sysusers</code> テーブルに “albert” というユーザ名が存在し、 <code>master.dbo.syslogins</code> に “victoria” というユーザ・ログインが存在します。このコマンドにより、“victoria” は “albert” という名前で現在のデータベースを使用できるようになります。
使用法	<pre>sp_addalias victoria, albert</pre> <ul style="list-style-type: none">• <code>sp_addalias</code> を実行すると、現在のデータベース内にあるユーザが別のユーザにマップされます。このマッピングを示す <code>sysalternates</code> では、2 つのユーザの <code>suids</code> (システム・ユーザ ID) が結合されています。• ユーザがエイリアスとして指定できるのは、一度に 1 つのデータベース・ユーザだけです。• <code>sp_helpuser</code> に特定のユーザ名を引数として指定すると、そのユーザにマップされているすべてのユーザのレポートを生成できます。• ユーザがデータベースを使用しようとする時、Adaptive Server はそのユーザが <code>sysusers</code> にリストされているかどうかを確認します。ユーザがこのテーブルにリストされていない場合、Adaptive Server は次に <code>sysalternates</code> を調べます。そのユーザの <code>suid</code> が <code>sysalternates</code> にリストされていて、データベース・ユーザの <code>suid</code> にマップされていると、そのユーザがデータベースを使用する間、Adaptive Server はマッピングの最初のユーザ名を 2 番目のユーザ名として扱います。 <p><i>loginame</i> に指名されたユーザ名がデータベースの <code>sysusers</code> テーブルにある場合、Adaptive Server はそのユーザのエイリアス ID を使用しません。これは、エイリアスがリストされている <code>sysalternates</code> を調べる前に <code>sysusers</code> が調べられ、<code>loginame</code> が検出されるためです。</p>

パーミッション `sp_addalias` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any user</code> 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者、 <code>sa_role</code> を持つユーザー、または <code>sso_role</code> を持つユーザーであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 **コマンド** [use](#)
システム・プロシージャ [sp_addlogin](#), [sp_adduser](#), [sp_dropalias](#), [sp_helpuser](#)

sp_addauditrecord

説明 ユーザが、監査証跡にユーザ定義の監査レコード(コメント)を入力できるようにします。

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

パラメータ

text

現在の監査テーブルに追加するメッセージのテキストです。テキストはテーブルの **extrainfo** フィールドに挿入されます。

db_name

レコード内で参照されるデータベースの名前です。この名前は現在の監査テーブルの **dbname** フィールドに挿入されます。

obj_name

レコード内で参照されるオブジェクト名です。この名前は現在の監査テーブルの **objname** フィールドに挿入されます。

owner_name

レコード内で参照されるオブジェクトの所有者です。この名前は現在の監査テーブルの **objowner** フィールドに挿入されます。

dbid

db_name のデータベース ID 番号です。この整数値は引用符で囲まないでください。 **dbid** は現在の監査テーブルの **dbid** フィールドに挿入されます。

objid

obj_name のオブジェクト ID 番号です。この整数値は引用符で囲まないでください。 **objid** は現在の監査テーブルの **objid** フィールドに挿入されます。

例 **例 1** “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.” を現在の監査テーブルの **extrainfo** フィールドに、“corporate” を **dbname** フィールドに、“payroll” を **objname** フィールドに、“dbo” を **objowner** フィールドに、“10” を **dbid** フィールドに、“1004738270” を **objid** フィールドにそれぞれ追加します。

```
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
```

例 2 このレコードを監査証跡に追加します。この例では、@ プレフィクスを付けたパラメータ名を使用しています。この方法を使用すると、一部のフィールドを空のままにしておくことができます。

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

使用法

- Adaptive Server は、すべての監査レコードを現在の監査テーブルに書き込みます。現在の監査テーブルは、**current audit table** 設定パラメータの値 (**sp_configure** で設定) によって決まります。1 つのシステムで最大 8 つの監査テーブル (**sysaudits_01**、**sysaudits_02**、以下同様に **sysaudits_08** まで) を設定できます。

注意 実際には、レコードはまずメモリ内監査キューに格納され、次に監査プロセスによって監査キューから現在の監査テーブルに書き込まれます。そのため、監査レコードがただちに監査テーブルに格納されると想定することはできません。

- 次の場合に、**sp_addauditrecord** を使用できます。
 - **sp_addauditrecord** の実行権限を持っている — その他の特別な役割は不要。
 - 監査が有効になっている — システム・セキュリティ担当者によって、**sp_configure** で **auditing** 設定パラメータが有効に設定されている。
 - **sp_audit** の **adhoc** オプションが **on** に設定されている。

パーミッション

sp_addauditrecord のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効化されている場合は、次のようになる。 <ul style="list-style-type: none"> • プロシージャに対する実行パーミッションを持つユーザは、sp_addauditrecord を実行できる。 • デフォルトでは、sso_role には実行パーミッションがある。 • sybsystemprocs のデータベース所有者は、実行パーミッションを付与できる。
細密なパーミッションが無効	細密なパーミッションが無効化されている場合は、次のようになる。 <ul style="list-style-type: none"> • プロシージャに対する実行パーミッションを持つユーザは、sp_addauditrecord を実行できる。 • デフォルトでは、sso_role には実行パーミッションがある。 • sa_role を持つユーザは、実行パーミッションを付与できる。 • sybsystemprocs のデータベース所有者は、他のユーザに実行パーミッションを付与できる。

監査

sysaudits テーブルの **event** カラムと **extrainfo** カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
1	adhoc	ユーザ定義監査レコード	extrainfo には、sp_addauditrecord の <i>text</i> パラメータが挿入される
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_audit](#)

sp_addaudittable

説明	監査機能のインストール後に、別のシステム監査テーブルを追加します。
構文	<code>sp_addaudittable devname</code>
パラメータ	<p><i>devname</i></p> <p>監査テーブルのデバイスの名前です。デバイス名または "default" を指定します。"default" を指定すると、<code>sybsecurity</code> データベースと同じデバイス上に監査テーブルが作成されます。"default" を指定しないと、Adaptive Server によって、指定したデバイスにテーブルが作成されます。</p>
例	<p>例 1 <code>auditdev2</code> にシステム監査テーブルを作成します。このプロシージャの実行時にシステム監査テーブルが 1 つ (<code>sysaudits_01</code>) しかなければ、新しい監査テーブルは <code>sysaudits_02</code> という名前が指定され、<code>auditdev2</code> 上の専用セグメント <code>aud_seg_02</code> に配置されます。</p> <pre>sp_addaudittable auditdev2</pre> <p>例 2 <code>sybsecurity</code> データベースと同じデバイス上にシステム監査テーブルを作成します。このプロシージャの実行時にシステム監査テーブルが 2 つ (<code>sysaudits_01</code> と <code>sysaudits_02</code>) ある場合、新しい監査テーブルは <code>sysaudits_03</code> という名前が指定され、<code>sybsecurity</code> データベースと同じデバイス上の専用セグメント <code>aud_seg_03</code> に配置されます。</p> <pre>sp_addaudittable "default"</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_addaudittable</code> を実行するときに、監査機能があらかじめインストールされていなければなりません。システム監査テーブルを追加するには、次の手順に従います。 <ol style="list-style-type: none"> a <code>disk init</code> を使用して、監査テーブル用のデバイスを作成します。UNIX 用であれば、次のようなコマンドを実行します。 <pre>disk init name = "auditdev2", physname = "/dev/rxyla", size = "5K"</pre> b <code>alter database</code> コマンドを使用して、このデバイスを <code>sybsecurity</code> データベースに追加します。たとえば、<code>auditdev2</code> を <code>sybsecurity</code> データベースに追加するには、次のコマンドを実行します。 <pre>alter database sybsecurity on auditdev2</pre> c <code>sp_addaudittable</code> を実行してテーブルを作成します。

- 定義済みの監査テーブルがいくつあるかによって、Adaptive Server は新しい監査テーブルと新しいセグメントの名前を決定します。たとえば、このプロシージャの実行前に 5 つの監査テーブルが定義されている場合、新しい監査テーブルの名前は `sysaudits_06` になり、新しいセグメントの名前は `aud_seg_06` になります。“default” を指定した場合、セグメントは `sybsecurity` データベースと同じデバイスに配置されます。“default” を指定しないでデバイス名を指定すると、そのデバイスにセグメントが配置されます。
- 監査テーブルの最大許容数は 8 です。すでに 8 つの監査テーブルが定義されている場合に、`sp_addauditable` を実行して新しくテーブルを追加しようとすると、エラー・メッセージが表示されます。
- 監査機能のインストール方法については、使用しているプラットフォームのインストール・ガイドを参照してください。監査機能の使用方法については、『システム管理ガイド』を参照してください。

パーミッション

`sp_addauditable` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage auditing</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sso_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_audit](#)

sp_addengine

説明	sp_addengine は、スレッド・モードでは動作しません。
プロセス・モードの考慮事項	既存のエンジン・グループにエンジンを追加します。グループがなければ、エンジン・グループを作成してからエンジンを追加します。
構文	sp_addengine engine_number, engine_group [, instance_id]
パラメータ	<p>engine_number グループに追加するエンジンの番号です。有効な番号は、0 と最大値の間で、最大値は設定されたオンライン・エンジン数から 1 を引いた値です。</p> <p>engine_group エンジンの追加先とするエンジン・グループの名前です。engine_group に指定したエンジン・グループが存在しない場合は、その名前のエンジン・グループが作成され、そのグループにエンジンが追加されます。エンジン・グループの名前は、識別子の規則に従ってください。詳細については、『リファレンス・マニュアル：ビルディング・ブロック』の「第 4 章 式、識別子、およびワイルドカード文字」を参照してください。</p> <p>instance_id クラスタ環境の場合 — エンジンまたはエンジン・グループの追加先とするインスタンスの ID です。</p>
例	<p>例 1 次の文では、DS_GROUP という名前のグループが存在しない場合に、このグループが作成されます。DS_GROUP が存在する場合は、そのグループにエンジン番号 2 が追加されます。</p> <pre>sp_addengine 2, DS_GROUP</pre> <p>例 2 インスタンス ID 8 にエンジン番号 5 を追加します。</p> <pre>sp_addengine 5, 8</pre>
使用法	<ul style="list-style-type: none"> sp_addengine は、engine_group の値が存在しない場合、新しいエンジン・グループを作成します。 sp_cluster set system_view が cluster に設定されている場合、クラスタ内のインスタンスにエンジンまたはエンジン・グループを追加できます。system_view が instance に設定されている場合、ローカル・インスタンスにのみエンジンまたはエンジン・グループを追加できます。

- エンジン・グループ ANYENGINE と LASTONLINE は定義済みのエンジン・グループです。ANYENGINE には、既存のすべてのエンジン・グループが含まれます。LASTONLINE は、最も大きい番号を持つエンジンを表します。新しいエンジン・グループはシステム管理者が追加します。定義済みのエンジン・グループは変更できません。
- [sp_bindexclass](#) を使用して、`engine_group` に関連付けられた実行クラスにアプリケーションまたはログインをバインドすると、すぐに `engine_number` に指定したエンジン上で関連するプロセスが開始されることがあります。
- `sp_engine` は、`sp_procxmode` を使用してトランザクション・モードを `anymode` に変更した後に、連鎖トランザクションを使用するセッションで実行できます。
- エンジンとの結び付きの割り当てを行うときは、あらかじめ環境を調査し、優先順位の低いアプリケーションの数と、利用できる Adaptive Server エンジン数を確認しておきます。優先順位の低いアプリケーションの詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。

パーミッション

`sp_addengine` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any execution class</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_addexclass](#), [sp_bindexclass](#), [sp_clearpsex](#), [sp_dropengine](#), [sp_setpsex](#), [sp_showcontrolinfo](#), [sp_showexclass](#), [sp_showpsex](#), [sp_unbindexclass](#)

sp_addexeclass

説明	クライアント・アプリケーション、ログイン、ストアド・プロシージャにバインドできるユーザ定義の実行クラスを作成または更新します。
プロセス・モードの考慮事項	定義済みのエンジン・グループ パラメータ ANYENGINE と LASTONLINE はプロセス・モードでのみ有効です。
構文	<code>sp_addexeclass classname, priority, timeslice, engine_group [, instance_id]</code>
パラメータ	<p><i>classname</i> 新しい実行クラスの名前です。</p> <p><i>priority</i> この実行クラスに関連付けたクライアント・アプリケーション、ログイン、またはストアド・プロシージャを実行するときの優先順位 の値を指定します。有効な値は、HIGH、LOW、MEDIUM です。</p> <p><i>timeslice</i> このクラスに関連付けられたプロセスに割り当てる時間単位です。 Adaptive Server は現在このパラメータを無視します。</p> <p><i>engine_group</i> このクラスに関連付けられたプロセスを実行できる既存のエンジン・グループを指定します。</p> <p><i>instance_id</i> (クラスタ環境の場合) ユーザ定義の実行クラスをバインドする インスタンスの ID です。</p>
例	<p>例 1 <i>priority</i> 値が LOW で、名前が DS である新しい実行クラスを定義し、このクラスをエンジン・グループ DS_GROUP に関連付けます。</p> <pre>sp_addexeclass "DS", "LOW", 0, "DS_GROUP"</pre> <p>例 2 クラスタ環境では、次のように優先度の値が LOW である、DS という名前の新しい実行クラスを定義し、インスタンス番号 8 にある DS_GROUP というエンジン・グループに関連付けます。</p> <pre>sp_addexeclass "DS", "LOW", 0, "DS_GROUP", 8</pre>
使用法	<ul style="list-style-type: none"> sp_addexeclass は、クライアント・アプリケーション、ログイン、ストアド・プロシージャにバインドできるユーザ定義の実行クラスを作成または更新します。指定したクラスがすでにある場合は、ユーザが指定した値でクラス属性が更新されます。 sp_addexeclass をスレッド・モードで実行する場合、Adaptive Server では、スレッド・プールの名前に engine_group を使用します。

- クラスタ環境では、`sp_cluster set system_view` が `cluster` に設定されている場合、クラスタ内のインスタンスに実行クラスを追加できます。`system_view` が `instance` に設定されている場合、ローカル・インスタンスにのみ実行クラスを追加できます。
- 作成する実行オブジェクトを特定のエンジン・グループのみに関連付けない場合は、定義済みのエンジン・グループ・パラメータ `ANYENGINE` を使用します。
- エンジン・グループの定義には、`sp_addengine` を使用します。実行クラス属性と、指定した実行クラスに関連付けられたあらゆるエンジン・グループのエンジンを表示するには、`sp_showexclass` を使用します。`sp_showcontrolinfo` では、既存のエンジン・グループのリストが出力されます。

パーミッション

`sp_addexclass` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any execution class</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ `sp_addengine`, `sp_bindexclass`, `sp_clearpsex`, `sp_dropengine`, `sp_dropexclass`, `sp_setpsex`, `sp_showcontrolinfo`, `sp_showexclass`, `sp_showpsex`, `sp_unbindexclass`

sp_addextendedproc

説明	master データベース内に ESP (拡張ストアド・プロシージャ) を作成する。
構文	<code>sp_addextendedproc esp_name, dll_name</code>
パラメータ	<p><i>esp_name</i> 拡張ストアド・プロシージャの名前です。ESP を実装する手続き型言語関数の名前と同じ名前を指定します。<i>esp_name</i> は、有効な Adaptive Server の識別子にしてください。</p> <p><i>dll_name</i> <i>esp_name</i> に指定した関数が含まれる DLL (動的リンク・ライブラリ) ファイルの名前です。<i>dll_name</i> の値は、拡張子なしで指定するか、または Windows NT の <i>.dll</i> や Sun Solaris の <i>.so</i> などのようにプラットフォーム固有の拡張子付きで指定できます。拡張子を指定する場合は、<i>dll_name</i> を引用符で囲みます。</p>
例	<p><i>sqlsrv.dll</i> ファイルに含まれている関数 <i>my_esp</i> の ESP を登録します。新しい ESP データベース・オブジェクトの名前も <i>my_esp</i> になります。</p> <pre>sp_addextendedproc my_esp, "sqlsrv.dll"</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_addextendedproc</code> は master データベースから実行します。 • <code>sp_addextendedproc</code> で追加できるのは、パラメータを取らない拡張ストアド・プロシージャだけです。正式なパラメータ・リストを必要とする拡張ストアド・プロシージャを追加する場合は、create procedure コマンドを使用し、<code>as external name</code> オプションと完全なパラメータ・リストを指定する必要があります。 • <i>esp_name</i> では大文字と小文字を区別します。DLL の関数名と同じにしてください。 • <i>dll_name</i> に指定する DLL は ESP の作成先のサーバ・マシンに存在し、DLL ディレクトリが次の場所に配置されていなければなりません。 <ul style="list-style-type: none"> • Windows NT — <i>\$PATH</i> • Compaq Tru64 — <i>\$LD_LIBRARY_PATH</i> • HP — <i>\$SH_LIBRARY_PATH</i> ファイルが見つからない場合、検索メカニズムはこれらの場所だけでなく、Windows NT では <i>%SYBASE%¥dll</i> を、その他のプラットフォームでは <i>\$SYBASE ¥lib</i> を検索します。

- Windows NT では、ESP 関数で C のランタイム・シグナル・ルーチン呼び出さないでください。Open Server™ は、Windows NT 上でシグナル処理をサポートしないので、このような呼び出しを行うと XP Server が失敗する可能性があります。

パーミッション

sp_addextendedproc のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage any ESP 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザーであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> Roles – 現在のアクティブな役割 Keywords or options – NULL Previous value – NULL Current value – NULL Other information – すべての入力パラメータ Proxy information – set proxy が有効な場合は元のログイン名

参照

コマンド [create procedure](#)

システム・プロシージャ [sp_dropextendedproc](#), [sp_helpextendedproc](#)

sp_addexternlogin

説明	(コンポーネント統合サービスのみ)コンポーネント統合サービスを介してリモート・サーバとの通信に使用する代替ログイン・アカウントとパスワードを作成します。
構文	<code>sp_addexternlogin server, loginame, externname [, externpasswd] [rolename]</code>
パラメータ	<p><i>server</i> リモート・サーバの名前です。master.dbo.syssservers テーブルのエントリによって、<i>remote_server</i> がローカル・サーバに認識されるようにします。</p> <p><i>loginame</i> ローカル・サーバに認識されているアカウントです。<i>loginame</i> には、master.dbo.syslogins テーブルに対応するエントリが存在する名前を指定する必要があります。ローカル・ユーザのリモート・アクセスを修正する権限を持つのは、“sa”アカウントと“sso”アカウント、および <i>loginame</i> アカウントだけです。</p> <p><i>externname</i> <i>server</i> に認識されているアカウントです。<i>server</i> が稼動しているノードの有効なアカウントでなければなりません。このアカウントは、<i>server</i> へのログイン時に使用します。</p> <p><i>externpasswd</i> <i>externname</i> のパスワードです。</p> <p><i>rolename</i> Adaptive Server のユーザに割り当てられた役割です。<i>rolename</i> を指定した場合、<i>login_name</i> は無視されます。</p>
例	<p>例 1 ログイン名 “bobj” でログインしたユーザがリモート・サーバ OMNI1012 にアクセスしたときに、リモート名 “jordan” とリモート・パスワード “hitchpost” が使用されます。ログイン名 “bobj” のリモート・ログインを追加または修正する権限を持つのは、“bobj” アカウントと、“sa” アカウントおよび “sso” アカウントだけです。</p> <pre>sp_addexternlogin OMNI1012, bobj, jordan, hitchpost</pre> <p>例 2 多対 1 のマッピングを設定して、DB2 に接続する必要がある Adaptive Server Enterprise の全ユーザに同じユーザ名とパスワードを割り当てます。</p> <pre>sp_addexternlogin DB2, NULL, login2, password2</pre> <p>例 3 Adaptive Server Enterprise の役割にリモート・ログインを割り当てることもできます。この機能によって、一定の役割を持つユーザ全員に、特定のリモート・サーバ用のログイン名とパスワードを割り当てることができます。</p> <pre>sp_addexternlogin DB2, NULL, login3, password3, role</pre>

使用法

- `sp_addexternlogin` は、リモート・サーバとの通信に使用する代替ログイン名とパスワードを割り当てます。パスワードは暗号形式で内部に格納されます。

注意 `sp_addexternlogin` を使用できるのは、コンポーネント統合サービスが設定されている場合だけです。

- マッピングは、1対1(特定のユーザ対象)、役割対1(役割ベース)、多対1(サーバ・ベース)、または `TDS loginrec` のクライアント・ログインとパスワードに基づいて設定できます。
- 1つのマッピングに対して、多数のログインとパスワードが割り当てられます。つまり、リモート・サーバにログインする必要があるすべてのユーザに、同じ名前とパスワードを割り当てることができます。
- 複数の外部ログインが設定されているユーザがリモート・サーバに接続する際の処理は、次のとおりです。1) 1対1のマッピングを使用します。2) 1対1のマッピングが存在しない場合は、アクティブな役割を使用します。3) 1対1のマッピングもアクティブな役割も存在しない場合は、多対1のマッピングを使用します。4) いずれも存在しない場合は、`Adaptive Server Enterprise` のログインとパスワードを使用します。
- 外部ログインを `Adaptive Server` の役割に割り当てることができます。特定の役割を共通して持つユーザ全員に対して、特定のリモート・サーバ用のログイン名とパスワードを割り当てることができます。
- 複数の役割をアクティブにしているユーザが使用するリモート・サーバへの接続を確立すると、各役割が外部のログイン・マッピングを検索します。各役割は、最初に見つけたマッピングを使用してログインを確立します。この順序は、ストアド・プロシージャ `sp_activeroles` で表示される順序と同じです。
- 役割マッピングを実行し、いずれかのユーザの役割を変更する (`set role` コマンドを使用) 場合は、役割マッピングを使用して確立されたリモート・サーバへの接続をすべて切断する必要があります。トランザクションが保留中であれば、接続を切断することはできません。トランザクションがアクティブであり、役割マッピングを使用したリモート接続が現在確立されているときは、`set role` を使用できません。
- `sp_addexternlogin` を実行するには、あらかじめ `sp_addserver` によってリモート・サーバを `Adaptive Server` に追加しておく必要があります。

- `externname` に指定するユーザ名と `externpasswd` に指定するパスワードは、`server` に指定するサーバが稼動しているノードの有効なユーザとパスワードの組み合わせでなければなりません。
- パスワードが自動的に期限切れになるように設定してあるサイトでは、外部ログインのパスワードを定期的に更新する必要があります。
- 外部ログインの定義を削除するときには、`sp_dropexternlogin` を使用します。
- `sp_addexternlogin` は、トランザクション内では使用できません。
- ローカル・ユーザのリモート・アクセスを変更できるのは、“sa” アカウントと `loginame` アカウントだけです。

パーミッション

`sp_addexternlogin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any remote login</code> 権限を持つユーザであることが必要。自分のログインについては、すべてのユーザが <code>sp_addexternlogin</code> を実行できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> または <code>sso_role</code> が付与されたユーザであることが必要。自分のログインについては、すべてのユーザが <code>sp_addexternlogin</code> を実行できる。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ `sp_addserver`, `sp_dropexternlogin`, `sp_helpexternlogin`, `sp_helpserver`

sp_addgroup

説明	データベースにグループを追加します。グループを使用すると、権限の付与および取り消しをまとめて行うことができます。
構文	<code>sp_addgroup grpname</code>
パラメータ	<code>grpname</code> グループの名前です。グループ名は、識別子の規則に従っていなければなりません。
例	現在のデータベースに <code>accounting</code> という名前のグループを作成します。 <code>sp_addgroup accounting</code>
使用法	<ul style="list-style-type: none"> • <code>sp_addgroup</code> は、データベースの <code>sysusers</code> テーブルに新しいグループを追加します。各グループのユーザ ID (<code>uid</code>) は、16384 以上になります (“<code>public</code>” は例外で、常に 0)。 • グループとユーザに同じ名前を使用することはできません。 • グループを作成した後は、<code>sp_adduser</code> を使用して新しいユーザを追加します。グループに既存のユーザを追加するときには、<code>sp_changegroup</code> を使用します。 • 各データベースには、作成時に “<code>public</code>” という名前のグループが設定されています。ユーザはすべて自動的に “<code>public</code>” のメンバーになります。各ユーザは、それ以外にもう 1 つのグループのメンバーになることができます。
パーミッション	<code>sp_addgroup</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any user</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者、 <code>sso_role</code> を持つユーザ、または <code>sa_role</code> を持つユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 コマンド [grant](#), [revoke](#)
 システム・プロシージャ [sp_adduser](#), [sp_changegroup](#), [sp_dropgroup](#), [sp_helpgroup](#)

sp_addlanguage

説明	代替言語における月と曜日の名前、およびその日付フォーマットを定義します。
構文	<code>sp_addlanguage language, alias, months, shortmons, days, datefmt, datefirst</code>
パラメータ	<p><i>language</i> 言語の公式言語名です。7 ビットの ASCII 文字のみを使用して入力します。</p> <p><i>alias</i> 代替言語の公式名を置き換えるエイリアスです。任意の名前を入力するか、または “null” を入力してエイリアスを公式言語名と同じにします。使用する端末が 8 ビットの ASCII 文字に対応している場合は、エイリアスにこれらの文字 ("francais" など) を使用できます。</p> <p><i>months</i> 各月のフル・ネームのリストです。1 月から 12 月の順に並べ、カンマで区切ります (スペースは使用不可)。月の名前は最大 20 文字まで指定できます。8 ビットの ASCII 文字も使用可能です。</p> <p><i>shortmons</i> 各月の略称のリストです。1 月から 12 月の順に並べ、カンマで区切ります (スペースは使用不可)。月の略称は最大 9 文字まで指定できます。8 ビットの ASCII 文字も使用可能です。</p> <p><i>days</i> 各曜日のフル・ネームのリストです。月曜日から日曜日の順に並べ、カンマで区切ります (スペースは使用不可)。曜日名は最大 30 文字まで指定できます。8 ビットの ASCII 文字も使用可能です。</p> <p><i>datefmt</i> <code>datetime</code>、<code>smalldatetime</code>、<code>date</code>、<code>time</code> データを入力する日付部分「月／日／年」の日付要素の順序です。有効な引数は <code>mdy</code>、<code>dmy</code>、<code>ymd</code>、<code>ydm</code>、<code>myd</code>、または <code>dym</code> です。“<code>dmy</code>” と指定すると、日付の順は「日／月／年」となります。</p> <p><i>datefirst</i> 日付計算での最初の曜日の番号を設定します。たとえば、月曜日は 1、火曜日は 2 (以下、水曜日～日曜日は 3～7) に設定します。</p>

例

次のストアド・プロシージャは、サーバで使用できる言語にフランス語を追加します。エイリアスは“null”を指定して、公式言語名と同じ“french”に設定します。日付の順序は“dmy”、つまり「日／月／年」の順です。“1”は、`days` リストの最初の項目である `lundi` (月曜日) が最初の曜日であることを表します。フランス語では、文頭以外では曜日と月の名前の最初の文字を大文字にしないため、この例で各曜日はすべて小文字で入力されています。

```
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
```

使用法

- 通常、Adaptive Server のいずれかの言語モジュールからの代替言語の追加は、`langinstall` ユーティリティまたは Adaptive Server のインストール・プログラムを使用して行います。言語モジュールは、その言語の日付の名前と翻訳されたエラー・メッセージを提供します。言語モジュールがサーバに組み込まれていない場合は、`sp_addlanguage` を使用して日付の名前とフォーマットを定義する必要があります。
- ユーザのデフォルトの言語を変更するには、`sp_modifylogin` を使用します。`sp_addlanguage` で追加した言語をユーザのデフォルト言語に設定した場合、その言語のローカライゼーション・ファイルがないと、ユーザのログイン時に、クライアント・ソフトウェアがローカライゼーション・ファイルをオープンできないという内容の説明メッセージが表示されます。

システム・テーブルの変更

- `sp_addlanguage` は `master.dbo.syslanguages` にエントリを作成し、各代替言語の `langid` カラムにユニークな数値を挿入します。`langid 0` は英語 (米国) 用に予約されています。
- `language` パラメータの値は公式言語名として `master.dbo.syslanguages` の `name` カラムに保管されます。言語名は、ユニークでなければなりません。Adaptive Server で使用できる代替言語のリストを表示するには、`sp_helplanguage` を使用します。
- `sp_addlanguage` で `alias` に NULL を指定した場合、`master.dbo.syslanguages` の `alias` カラムには公式言語名の値が設定されます。ただし、`syslanguage.alias` の値は、システム管理者が `sp_setlangalias` を使用して変更できます。

- sp_addlanguage は、master.dbo.syslanguages の upgrade カラムの値を 0 に設定します。

sp_addlanguage で追加した言語の日付

- 言語モジュールで追加した代替言語の場合、Adaptive Server は、日付の値を datetime データ型としてクライアントに送信し、クライアントは、ローカライゼーション・ファイルを使用して日付をユーザの現在の言語で表示します。sp_addlanguage で追加した日付文字列に対しては、次のように、サーバで convert 関数を使用して日付を文字データに変換します。この場合、pubdate に datetime 型のデータを指定し、table には任意のテーブルを指定します。

```
select convert(char, pubdate) from table
```

- ユーザが日付の値を入力し、sp_addlanguage で作成された日付の名前を使用する必要がある場合、クライアントは、これらの値を文字データとして入力し、文字データとしてサーバに送る必要があります。

パーミッション

sp_addlanguage のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage server 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

参照

コマンド set

システム・プロシージャ [sp_droplanguage](#), [sp_helplanguage](#), [sp_modifylogin](#), [sp_setlangalias](#)

ユーティリティ langinstall

sp_addlogin

説明	Adaptive Server 15.7 以降では、このシステム・プロシージャは推奨されていません。Adapter Server にログイン・アカウントを追加するには、 <code>create login</code> コマンドを使用します。
構文	なし
パラメータ	なし
使用法	なし

sp_addmessage

説明	ストアド・プロシージャの print 呼び出しと raiserror 呼び出し、および sp_bindmsg で使用するために、 sysusermessages にユーザ定義メッセージを追加します。
構文	<pre>sp_addmessage message_num, message_text [, language [, with_log [, replace]]]</pre>
パラメータ	<p>message_num ユーザ定義メッセージのメッセージ番号は 20000 以上でなければなりません。</p> <p>message_text 追加するメッセージのテキストです。最大長は 1024 バイトです。</p> <p>language 検索されるメッセージの言語です。language には、syslanguages テーブル内の有効な言語名を指定してください。このパラメータを指定しないと、メッセージの言語が @@langid で指定されているデフォルトのセッション言語であるとみなされます。</p> <p>with_log ロギングが有効になっている場合に、Adaptive Server のエラー・ログと、Windows NT サーバの Windows NT イベント・ログにメッセージを記録するかどうかを指定します。with_log が TRUE の場合、エラーの重大度とは関係なくメッセージが記録されます。with_log が FALSE の場合、メッセージが記録されるかどうかはエラーの重大度によって決まります。with_log の指定を省略すると、デフォルトで FALSE になります。</p> <p>replace 同じ langid と同じメッセージ番号を持つ既存のメッセージを上書きするかどうかを指定します。replace を指定すると、既存のメッセージが上書きされます。replace を指定しないと、メッセージは上書きされません。replace の値を指定しない場合、このパラメータのデフォルトの動作では、既存のメッセージは上書きされません。</p>
例	<p>例 1 メッセージ番号を 20001 として、メッセージを sysusermessages に追加します。</p>

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

例 2 メッセージ番号を 20002 として、メッセージを `sysusermessages` に追加します。イベント・ログが有効な場合、このメッセージは Windows NT サーバの Windows NT イベント・ログに記録されるだけでなく、Adaptive Server のエラー・ログにも追加されます。デフォルト・セッション言語に 20002 番のメッセージがすでにある場合は、このメッセージが古いメッセージを上書きします。

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

使用法

- `sp_addmessage` では、`@replace = "replace"` と指定しないかぎり、同じ番号と同じ `langid` を持つ既存のメッセージの上書きは行われません。
- `print` と `raiserror` は出力するメッセージ・テキスト内のプレースホルダを認識します。1 つのメッセージに 20 個までのユニークなプレースホルダを任意の順序で指定することができます。このプレースホルダは、メッセージ・テキストがクライアントに送信される時に、メッセージに続く引数のフォーマットされた内容に置き換えられます。

プレースホルダには番号が付けられます。これは、Adaptive Server で、異なる文法構造の言語にメッセージを変換するときに引数の位置を変更できるようにするためです。引数のプレースホルダは、先頭にパーセント記号 (%)、次に 1 から 20 までの整数、最後に感嘆符 (!) という形式 " `%nn!` " で表されます。引数リスト内の文字列の整数は、引数の番号を表します。"`%1!`" "`%1!`" は元のバージョンでの最初の引数、"`%2!`" は 2 番目の引数となります (以降同様)。

- 元のメッセージを置き換える `replace` オプションを使用して `sp_addmessage` を実行できるのは、メッセージを作成したユーザだけです。

パーミッション

`sp_addmessage` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	<p>細密なパーミッションが有効の場合、すべてのユーザが <code>sp_addmessage</code> を実行できる。</p> <p><code>with_log</code> を指定してメッセージを追加するには、データベース所有者であるか、データベースに対して <code>own database</code> 権限を持つユーザであることが必要。</p>
細密なパーミッションが無効	<p>細密なパーミッションが無効の場合、すべてのユーザが <code>sp_addmessage</code> を実行できる。</p> <p><code>with_log</code> を指定してメッセージを追加するには、データベース所有者または <code>sa_role</code> を持つユーザであることが必要。</p> <p>元のメッセージを置き換える <code>replace</code> オプションを使用して <code>sp_addmessage</code> を実行できるのは、メッセージを作成したユーザだけです。</p>

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
15	create	sp_addmessage	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – メッセージ番号 • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [print](#), [raiserror](#)

システム・プロシージャ [sp_altermessage](#), [sp_bindmsg](#), [sp_dropmessage](#), [sp_getmessage](#)

sp_addobjectdef

説明 (コンポーネント統合サービスのみ) ローカル・テーブルと外部記憶装置の間のマッピングを指定します。

構文 sp_addobjectdef *tablename*, *objectdef* [, "*objecttype*"]

パラメータ

tablename

ローカル・テーブルに定義されているオブジェクト名です。

tablename は次のどのフォーマットでもかまいません。

dbname.owner.object

dbname..object

owner.object

object

dbname と *owner* の指定は省略可能です。 *object* は必須です。 *owner* を選択しない場合は、デフォルト (現在のユーザ名) が使用されます。 *dbname* を指定する場合は、現在のデータベース名でなければなりません。また、この場合 *owner* を指定するか、または *dbname..object* のフォーマットのプレースホルダを所有者に設定する必要があります。 *tablename* の値が複数ある場合は、引用符で囲みます。

objectdef

オブジェクトの外部記憶ロケーションを指定する文字列です。

objectdef の *objecttype* には、テーブル、ビュー、リモート・サーバにアクセスできる読み込み専用 RPC (リモート・プロシージャ・コール) 結果セットのいずれかを指定できます。テーブル、ビュー、RPC では、 *objectdef* に次のフォーマットを使います。

server_name.dbname.owner.object

server_name と *object* の指定は必須です。 *dbname* と *owner* の指定は任意ですが、指定しない場合は、 *dbname.object* のフォーマットのプレースホルダを指定する必要があります。

詳細については、『コンポーネント統合サービス・ユーザーズ・ガイド』の「サーバ・クラス」を参照してください。

objecttype

objectdef. で指定するオブジェクトのフォーマットを指定する値の 1 つです。表 1-1 に、その有効な値を示します。 *objecttype* の値は引用符で囲みます。

表 1-1: objecttype に指定できる値

値	説明
table	<i>objectdef</i> により命名されたオブジェクトは、リモート・サーバにアクセスできるテーブルであることを示す。この値は <i>objecttype</i> のデフォルトである。

値	説明
view	<i>objectdef</i> により命名されたオブジェクトは、リモート・サーバが管理し、テーブルとして処理するビューであることを示す。
rpc	<i>objectdef</i> により命名されたオブジェクトは、リモート・サーバが管理する RPC であることを示す。Adaptive Server は、RPC からの結果セットを、読み込み専用テーブルとして処理する。

表 1-2 に、各 *objecttype* の用途を示します。

表 1-2: *objecttype* の用途

<i>objecttype</i>	<i>create table</i>	<i>create existing table</i>	テーブルへの書き込み	テーブルからの読み込み
table	可能	可能	可能	可能
view	不可	可能	可能	可能
rpc	不可	可能	不可	可能

例 1 データベース *finance* にあるローカル・テーブル *accounts* を、SYBASE という名前のリモート・サーバのリモート・オブジェクト *pubs.dbo.accounts* にマップします。現在のデータベースは *finance* です。後続の *create table* で、*pubs* データベースにテーブルを作成します。*pubs.dbo.accounts* が既存のテーブルの場合、*create existing table* 文によりテーブル *finance.dbo.accounts* にリモート・テーブルに関する情報を移植します。

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

例 2 リモート・サーバ NEWYORK にある、“kelly” が所有するデータベース *wallstreet* の *stockcheck* という名前の RPC に、ローカル・テーブル *stockcheck* をマップします。RPC *stockcheck* からの結果セットは、読み込み専用テーブルとみなされます。通常、この後の操作は、オブジェクト *stockcheck* 用の *create existing table* 文になります。

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

使用法

- *sp_addobjectdef* では、ローカル・テーブルと外部記憶装置の間のマッピングを指定します。また、その外部記憶装置にあるオブジェクトのフォーマットを示します。*sp_addobjectdef* は、コンポーネント統合サービスがインストールされ設定されている場合のみ使用できます。
- *sp_addobjectdef* は *sp_addtabledef* コマンドに代わるコマンドです。*sp_addobjectdef* では、既存のスクリプトを修正しないでそのまま実行できます。内部で、*sp_addtabledef* は *sp_addobjectdef* を呼び出します。
- テーブル所有者として他のユーザの名前を提供できるのは、システム管理者だけです。

- *objecttype* が `table`、`view`、`rpc` のいずれかの場合、*objectdef* パラメータのフォーマットは次のようになります。

```
"server_name.database.owner.tablename"
```

- *server_name* — `sp_addserver` によって `sys.servers` にすでに追加されているサーバです。
- *database* — 必須ではありません。サーバ・クラスによっては `dbname` をサポートしていないものもあります。
- *owner* — あいまいさを避ける意味からも必須です。*owner* を省略すると、外部ログインがリモート・オブジェクトの所有者に対応しているかどうかによって、参照するリモート・オブジェクトが変わる場合があります。
- *tablename* — リモート・サーバ・テーブルの名前です。
- `create table` または `create existing table` コマンドを発行する前に `sp_addobjectdef` を使用します。ただし、リモート・テーブルがある場合は、`create proxy_table` を実行する前に `sp_addobjectdef` を使用する必要はありません。

`create table` は、*objecttype* 値が `table` および `file` の場合のみ有効です。`create table` または `create existing table` を使用すると、Adaptive Server は `sysattributes` を調べ、そのオブジェクトに対して何らかのテーブル・マッピングが指定されているかどうかをチェックします。`create existing table` 文に続けて、*objecttype* の `view` 値と `rpc` 値を指定してください。

- テーブルの作成が終わると、そのローカルテーブル名に対して行われる参照 (`select`、`insert`、`delete`、`update` によって) は、すべて正しい位置にマップされます。

パーミッション

`sp_addobjectdef` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、テーブル所有者か <code>manage database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、テーブル所有者、データベース所有者、または <code>sa_role</code> を持つユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

コマンド [create existing table](#), [create table](#), [drop table](#)

システム・プロシージャ [sp_addlogin](#), [sp_addserver](#), [sp_defaultloc](#), [sp_dropobjectdef](#), [sp_helpserver](#)

sp_addremotelogin

説明	master.dbo.sysremotelogins にエントリを追加して、新しいリモート・サーバ・ユーザを認可します。
構文	sp_addremotelogin remoteserver [, loginame [, remotename]]
パラメータ	<p><i>remoteserver</i></p> <p>リモート・ログインを適用するリモート・サーバの名前です。このサーバは、master.dbo.sysservers テーブルのエントリ (sp_addserver で作成) によってローカル・サーバに認識されていなければなりません。</p>

注意 このマニュアル・ページで「ローカル・サーバ」というときは、「リモート・サーバ」から実行されているリモート・プロシージャを実行中のサーバを指します。

loginame

ローカル・サーバのユーザのログイン名です。loginame は、master.dbo.syslogins テーブルに実際に存在していなければなりません。

remotename

リモート・サーバがローカル・サーバにログインするときに使用する名前です。ローカル loginame に明示的に一致しない remotenames はすべて、自動的にローカル名になります。例 1 のローカル名は、ログインに使用するリモート名です。例 2 のローカル名は “albert” です。

例 **例 1** ログインの検証用に、リモート・サーバ GATEWAY の sysremotelogins テーブルにエントリを作成します。これは、ローカル・サーバとリモート・サーバに同じユーザが存在するときに、リモート名をローカル名にマップする簡単な方法です。

```
sp_addremotelogin GATEWAY
```

この例の結果として、sysremotelogins に、suid カラム値が -1 で、remoteusername カラム値が NULL のローが 1 つ作成されます。

例 2 リモート・サーバ GATEWAY のすべてのログインをローカル・ユーザ名 “albert” にマップするエントリを作成します。Adaptive Server は、suid カラム内の Albert のサーバ・ユーザ ID と、remoteusername の null 値を使用して、ローを sysremotelogins に追加します。

```
sp_addremotelogin GATEWAY, albert
```

これらのログイン・ユーザがローカル・サーバで RPC を実行するには、ローカル・サーバにログインするときに RPC 接続のパスワードを指定します。ただし、ローカル・サーバの “trusted” ユーザであればパスワードの指定は不要です。これらのログインを “trusted” として定義するには、[sp_remoteoption](#) を使用します。

例 3 リモート・サーバ GATEWAY のリモート・ユーザ “pogo” からのリモート・ログインを、ローカル・ユーザ “ralph” にマップします。Adaptive Server は、`suid` カラム内の Ralph のサーバ・ユーザ ID と、`remoteusername` カラム内の “pogo” を使用して、ローを `sysremotelogins` に追加します。

```
sp_addremotelogin GATEWAY, ralph, pogo
```

使用法

- リモート・ログインを受信すると、ローカル・サーバは、次の 3 とおりの方法でリモート・ユーザをローカル・ユーザにマップしようとしています。
 - まず、ローカル・サーバは、リモート・サーバ名とリモート・ユーザ名に一致するエントリを `sysremotelogins` で探します。ローカル・サーバは、エントリが見つかると、そのローのローカル・サーバ・ユーザ ID を使用して、リモート・ユーザにログインします。これは、指定されたリモート・ユーザからのマッピングに適用されます。
 - 一致するローが見つからない場合、ローカル・サーバは、ローカル・サーバ・ユーザ ID が -1 のローを、`null` リモート名で探します。そのようなローが見つければ、リモート・ユーザはそのローのローカル・サーバ・ユーザ ID にマップされます。これは、リモート・サーバのリモート・ユーザからローカル名へのマッピングであればすべてに適用されます。
 - 以上の 2 つが失敗した場合、ローカル・サーバは、リモート名が `null` であり、ローカル・サーバ・ユーザ ID が -1 であるエントリを `sysremotelogins` テーブルで探します。該当するローが見つかった場合、ローカル・サーバは、リモート・サーバによって指定されたリモート名に基いて、ローカル・サーバ・ユーザ ID を `syslogins` テーブルで探します。これはリモート・サーバとローカル・サーバのログイン名が同じ場合に適用されます。
- ローカル・ユーザの名前が、リモート・サーバ上で異なる場合があります。

- `sp_addremotelogin` でリモート・サーバのすべてのユーザを同じローカル名にマップする場合は、`sp_remoteoption` を使用してそれらのユーザに “trusted” オプションを指定します。たとえば、“albert” にマップされているサーバ `GOODSRV` のすべてのユーザを “trusted” として設定するには、`sp_remoteoption` を次のように使用します。

```
sp_remoteoption GOODSRV, albert, NULL, trusted, true
```

“trusted” として設定されていないログイン・ユーザは、リモート・サーバへのログイン時にローカル・サーバのパスワードを指定しないかぎり、ローカル・サーバ上で `RPC` を実行できません。`Open Client™` の `Client-Library™` では、ユーザは、ルーチン `ct_remote_pwd` を使用してサーバ間接続のパスワードを指定できません。`isql` と `bcp` では、`RPC` 接続のパスワードは指定できません。

ユーザが「統一化ログイン」でリモート・サーバにログインする場合、これらのログインはセキュリティ・メカニズムによってすでに認証されています。このようなログインにもローカル・サーバによる “trusted” の指定が必要です。そのような指定がない場合は、リモート・サーバにログインするときにサーバのパスワードを指定しなければなりません。

- すべてのリモートのログイン・エントリには、ステータスが設定されています。`trusted` オプションのデフォルト・ステータスは、`false` (`trusted` でない) です。これは、リモート・ログインがそのエントリを使用してログインすると、パスワードが調べられることを意味します。パスワードの検査を行わないようにするには、`sp_remoteoption` で `trusted` オプションのステータスを `true` に変更します。

パーミッション

`sp_addremotelogin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any remote login</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 **マニュアル** リモート・プロシージャ・コールに対するサーバの設定と「統一化ログイン」の使用方法の詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_addlogin](#), [sp_addserver](#), [sp_droptremotelogin](#), [sp_helpremotelogin](#), [sp_helprotect](#), [sp_helpserver](#), [sp_remoteoption](#)

ユーティリティ `isql`

sp_addsegment

説明	データベースにデータベース・デバイスのセグメントを定義します。
構文	<code>sp_addsegment segname, dbname, devname</code>
パラメータ	<p>segname データベースの <code>syssegments</code> テーブルに追加する新しいセグメントの名前です。セグメント名は、各データベース内でユニークです。</p> <p>dbname セグメントを定義するデータベース名を指定します。dbname は、現在のデータベース名か、<code>sp_addsegment</code> を修飾しているデータベース名と一致していなければなりません。</p> <p>devname segname を配置するデータベース・デバイスの名前です。データベース・デバイスは、そのデバイスに対応するセグメントを複数持つことができます。</p>
例	<p>例 1 <code>dev1</code> という名前のデータベース・デバイスで、<code>pubs2</code> データベース用に <code>indexes</code> という名前のセグメントを作成します。</p> <pre>sp_addsegment indexes, pubs2, dev1</pre> <p>例 2 <code>pubs2_dev</code> という名前のデータベース・デバイスで、<code>pubs2</code> データベース用に <code>indexes</code> という名前のセグメントを作成します。</p> <pre>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, pubs2_dev</pre>
使用法	<ul style="list-style-type: none"> • すでに排他セグメントがあるデバイスにセグメントを作成することはできません。作成しようとする、次のようなエラー・メッセージが表示されます。 <p style="margin-left: 40px;">A segment with a virtually hashed table exists on device orders_dat.</p> • <code>sp_addsegment</code> では、<code>disk init</code> で作成され <code>alter database</code> または <code>create database</code> コマンドで特定のデータベースに割り当てられたデータベース・デバイスのセグメント名を定義します。

- セグメントを定義した後は、`create table` コマンドや `create index` コマンド、または `sp_placeobject` プロシージャでそのセグメントを使用して、テーブルまたはインデックスをセグメント上に配置します。

ある特定のセグメントにテーブルまたはインデックスを作成すると、次からはそのテーブルまたはインデックス用のデータはすべて同じセグメントに置かれます。

- セグメントの範囲を、同じデータベースで使用している別のデータベース・デバイスに拡張するには、システム・プロシージャ `sp_extendsegment` を使用します。
- あるデータベースが、`alter database` によって、そのデータベースで使用しているデバイス上で拡張されると、そのデバイスにマップされているセグメントも拡張されます。
- `create database` コマンドまたは `alter database` コマンドに組み込まれた各データベース・デバイスには、`system` セグメントと `default` セグメントがマップされます。また、`logsegment` は、`create database` への `log on` 拡張機能によって、または `sp_logdevice` によって別のデバイスに配置されないかぎり、各デバイスにマップされます。詳細については、『システム管理ガイド』を参照してください。
- `sp_addsegment` はデータとログの両方が同じデバイスに存在するデータベース (`log on` オプションを指定しないで作成したデータベースなど) で使用できますが、次のようにデータベースを作成した場合、Adaptive Server からエラー・メッセージが返されます。

```
create database dbname on devicename log on devicename with override
```

パーミッション

`sp_addsegment` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage database</code> パーミッションを持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [alter database](#), [create index](#), [create table](#), [disk init](#)

システム・プロシージャ [sp_dropsegment](#), [sp_extendsegment](#), [sp_helpdb](#), [sp_helpdevice](#), [sp_placeobject](#)

sp_addserver

説明 リモート・サーバまたはローカル・サーバの名前を定義します。host パラメータと port パラメータを使用する場合はリモート・プロシージャ・コール (RPC) 用のサーバを指定します。

構文 sp_addserver *lname* [, *class* [, *pname*]]
コンポーネント統合サービス (CIS) のみ

sp_addserver '*logical_server_name*', ASEnterprise, '*host:port:filter*'

パラメータ

lname

使用しているシステム上のサーバの指定に使用する名前です。sp_addserver は、*lname* に指定したサーバのエントリが **syssservers** に存在しない場合に、このサーバのローをテーブルに追加します。サーバ名はユニークで、識別子の規則に従っていなければなりません。

class

追加するサーバのカテゴリを指定します。サーバの *class* に “null” を指定すると、デフォルトで “ASEnterprise” になります。*class* パラメータに指定できる値は次のとおりです。

- **local** — Adaptive Server の起動後または再起動後に一度だけ使用されるローカル・サーバ (1 つだけの場合もある)。このローカル・サーバ名は Adaptive Server が出力するメッセージに表示されます。
- **null** - カテゴリが定義されていないリモート・サーバ。
- **ASEnterprise** — Adaptive Server Enterprise のすべてのバージョン。SQL Server 4.9 はサポートされていません。
- **ASAnywhere** — Adaptive Server Anywhere バージョン 6.0 以降。
- **ASIQ** — サーバ・クラス ASIQ のサーバ。Adaptive Server IQ 12.0 以降のすべてのバージョンです。
- **direct_connect** (コンポーネント統合サービスのみ) — インタフェース仕様 **direct_connect** に準拠した Open Server ベースのアプリケーション。
- **sds** — 『Adaptive Server Specialty Data Store Developer’s Kit』で説明されている Specialty Data Store™ のインタフェース要件に準拠するものです。

これらのパラメータ値の詳細については、『コンポーネント統合サービス・ユーザーズ・ガイド』の「第2章 コンポーネント統合サービスの概要」にある「リモート・サーバ」を参照してください。

注意 Adaptive Server はサーバ・クラス db2 をサポートしていません。db2 を使用するには、db2 サーバ・クラスを `direct_connect` サーバ・クラスにマイグレートしてください。

pname

インタフェース・ファイル内で、*lname* に指定したサーバの名前です。これにより、通信する必要がある他の Adaptive Server や Backup Server 用のローカル・エイリアスを設定できます。*pname* を指定しない場合、*lname* が使用されます。

(コンポーネント統合サービスのみ) *pname* によって、接続先のサーバのホスト名または IP アドレスとポートを指定できます。この指定により、CT-Library の使用時に、このサーバに対してディレクトリ・サービス (LDAP や `interfaces` ファイルなど) を使用する必要がなくなります。次のフォーマットで指定します。

```
"hostname:port"
```

```
"ipaddr:port"
```

注意 このオプションを使用する場合は、ホスト名とポートを一重引用符か二重引用符で囲んでください。

filter

クラスタ環境の場合 - リモート・プロシージャ・コール (RPC) 用のリモート・サーバを追加します。

```
filter = ssl [= 'CN = common_name']
```

次のフォーマットで *host:port* 番号を宣言します。

```
ip_address:port
```

例

例 1 (クラスタ環境の場合) `big_logical_server` という名前のリモート・サーバを追加します。

```
sp_addserver 'big_logical_server', ASEntprise,
'maynard:23954:ssl= "CN=ase1.big server 1.com"
```

共通名に関する規則は、動的なリスナやディレクトリ・サービス・エントリに適用されるものと同じです。

例 2 `master.dbo.syssservers` に、GATEWAY という名前のリモート・サーバのエントリを追加します。*pname* も GATEWAY です。

```
sp_addserver GATEWAY
```

例 3 master.dbo.sys.servers に、GATEWAY という名前のリモート・サーバのエントリを追加します。pname は VIOLET です。別の pname を持つ GATEWAY 用の sys.servers エントリがすでに存在していると、サーバの pname が GATEWAY から VIOLET に変更されます。

```
sp_addserver GATEWAY, null, VIOLET
```

例 4 PRODUCTION という名前のローカル・サーバのエントリを追加します。

```
sp_addserver PRODUCTION, local
```

例 5 (コンポーネント統合サービスのみ) ホスト名を myhost ポート番号を 10224 として、リモート Adaptive Server のエントリを追加します。

```
sp_addserver S1, ASEnterprise, "myhost:10224"
```

注意 pname にこの構文を使用した場合、Adaptive Server のサイト・ハンドラはこのサーバに正常に接続できません。pname でのこの構文を認識できるのは、CIS 接続だけです。

例 6 (コンポーネント統合サービスのみ) ホスト IP を 192.123.456.010、ポート番号を 11222 として、リモート Adaptive Server のエントリを追加します。

```
sp_addserver S3, direct_connect, "192.123.456.010:11222"
```

使用法

- sys.servers テーブルによって、ローカル・サーバ名とそのオプション、およびローカル・サーバが通信できるリモート・サーバが識別されます。

リモート・サーバでリモート・プロシージャ・コールを実行するには、そのリモート・サーバが sys.servers テーブルに存在しなければなりません。
- Iname に指定したサーバ名が sys.servers テーブルにすでに存在する場合、sp_addserver はそのリモート・サーバの srvnetname を、pname に指定した名前に変更します。この場合、sp_addserver は、変更されたサーバの名前と、変更前のネットワーク名および変更後のネットワーク名をレポートします。
- サーバのインストールまたはアップグレードを行うと、sys.servers に Backup Server 用のエントリが追加されます。このエントリを削除すると、使用しているデータベースをバックアップできなくなります。

- Adaptive Server では、Backup Server に SYB_BACKUP の *Iname* が必要です。この名前を Backup Server の名前として使用しない場合や、システムで複数の Backup Server が稼動している場合は、`sp_addserver` でサーバ SYB_BACKUP の *pname* を修正することで、Adaptive Server が Backup Server と通信してデータベースのダンプやロードを実行できるようにします。
- *Iname*、*pname*、*class* に、`syssservers` にすでに存在している値を指定すると、`sp_addserver` からエラー・メッセージが出力されます。この場合、`syssservers` は更新されません。
- サーバ・オプションの設定やクリアには、`sp_serveroption` を使用します。

パーミッション

`sp_addserver` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。 共有ディスク・クラスタであるサーバについて <code>sp_addserver</code> を実行するには、 <code>manage cluster</code> 権限と <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル『コンポーネント統合サービス・ユーザズ・ガイド』
システム・プロシージャ `sp_addremotelogin`, `sp_droptremotelogin`,
`sp_dropserver`, `sp_helpremotelogin`, `sp_helpserver`, `sp_serveroption`

sp_addthreshold

説明 データベース・セグメントの領域の使用状況を監視するスレッシュホールドを作成します。セグメントの空き領域が指定されたレベルを下回ると、Adaptive Server は関連するストアド・プロシージャを実行します。

構文 `sp_addthreshold dbname, segname, free_space, proc_name`

パラメータ

dbname

スレッシュホールドを追加するデータベースです。これは、現在のデータベースの名前でなければなりません。

segname

空き領域をモニタするためのセグメントです。デフォルトのセグメント (“default”) を指定するときは、引用符を使用してください。

free_space

スレッシュホールドを超過した時点での空きページの数です。セグメント内の空き領域がこのレベルより下になると、Adaptive Server は対応するストアド・プロシージャを実行します。

proc_name

segname に指定したセグメント上の空き領域が *free_space* に指定した値を下回ったときに実行するストアド・プロシージャです。このプロシージャは、現在の Adaptive Server または Open Server 上のどのデータベースでも見つけることができます。スレッシュホールドは、リモートの Adaptive Server でプロシージャを実行することはできません。

例 **例 1** segment1 のスレッシュホールドを作成します。segment1 の空き領域が 200 ページを下回ると、Adaptive Server はプロシージャ pr_warning を実行します。

```
sp_addthreshold mydb, segment1, 200, pr_warning
```

例 2 user_data セグメントのスレッシュホールドを作成します。user_data の空き領域が 100 ページを下回ると、Adaptive Server は Open Server の mail_me プロシージャに対してリモート・プロシージャ・コールを実行します。

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

例 3 pubs2 データベースの indexes セグメントのスレッシュホールドを作成します。このコマンドはどのデータベースからも発行できます。

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

使用法

スレッシュホルドの超過

- スレッシュホルドを超過すると、Adaptive Server は関連したストアド・プロシージャを実行します。Adaptive Server は次の検索パスを使用して、スレッシュホルド・プロシージャを検索します。
- プロシージャ名にデータベースを指定しない場合は、Adaptive Server がスレッシュホルドを超過したデータベース内を検索します。
- このデータベース内にプロシージャがなく、しかもプロシージャ名が “sp_” で始まっている場合、Adaptive Server は `sysystemprocs` データベース内を検索します。

プロシージャがどちらのデータベース内にもなければ、Adaptive Server はエラー・メッセージをエラー・ログに送信します。

- Adaptive Server はヒステリシス値 (グローバル変数 `@@thresh_hysteresis`) を使用して、空き領域内での変動に対するスレッシュホルドの感度を決定します。スレッシュホルドは、そのプロシージャを実行するとアクティブでなくなります。スレッシュホルドは、セグメントの空き領域量がスレッシュホルドを超えて `@@thresh_hysteresis` ページ分に到達するまで、アクティブでない状態のままです。これによって、スレッシュホルドが空き領域のわずかな変化に反応して、繰り返しそのプロシージャを実行することがなくなります。

ラストチャンス・スレッシュホルド

- デフォルトでは、Adaptive Server は、ログが存在するセグメントの空き領域を監視し、空き領域がトランザクション・ログを正常にダンプするために必要な量を下回ると `sp_thresholdaction` を実行します。この空き領域の量は「ラストチャンス・スレッシュホルド」と呼ばれ、Adaptive Server により計算されます。ユーザは変更できません。
- トランザクションのログを取る前にラストチャンス・スレッシュホルドを超過すると、Adaptive Server は、ログの領域が解放されるまでトランザクションを中断します。特定のデータベースに対してこの動作を変更するには、`sp_dboption` を使用します。`sp_dboption "abort tran on log full" オプション` を `true` に設定すると、ラストチャンス・スレッシュホルドの超過時にログに記録されていないトランザクションがすべてロールバックされます。

- **master** データベースを含むすべてのデータベースに、ラストチャンス・スレッシュホルドがあります。スレッシュホルドとは、トランザクション・ログのバックアップが必要となる空きログ・ページ数の見積もり値のことです。ログ・セグメントに割り付ける領域を増やすと、それに従ってラストチャンス・スレッシュホルドが自動的に調整されます。

スレッシュホルドの追加作成

- データベースにはそれぞれ、ラストチャンス・スレッシュホルドを含めて最大 256 個のスレッシュホルドを作成できます。
- スレッシュホルドを追加する場合は、最も近いスレッシュホルドとの間に、`@@thresh_hysteresis` 値の 2 倍以上のページ数の間隔を空けます。

スレッシュホルド・プロシージャの作成

- データベースのスレッシュホルド・プロシージャの作成は、`create procedure` 権限を持つすべてのユーザが実行できます。通常、`sybsystemprocs` データベースでの `sp_thresholdaction` の作成はシステム管理者が行い、ユーザ、データベースでのスレッシュホルド・プロシージャの作成はデータベース所有者が行います。
- `sp_addthreshold` は、指定されたプロシージャが存在するかどうかの確認は行いません。スレッシュホルドを追加してから、スレッシュホルドが実行するプロシージャを作成することができます。
- `sp_addthreshold` は、スレッシュホルド・プロシージャを追加しようとするユーザに “`sa_role`” が直接与えられているかどうかを確認します。スレッシュホルド・プロシージャの作成時にアクティブになっているシステム標準の役割はすべて、プロシージャを記述するユーザの有効な役割として `systhresholds` に保管されます。ただし、スレッシュホルドの起動時にアクティブになっているのは、直接付与されたシステム標準の役割だけです。間接的に与えられたシステム標準の役割とユーザ定義の役割はアクティブになっていません。
- **Adaptive Server** は、次の 4 つのパラメータをスレッシュホルド・プロシージャに渡します。
 - `@dbname, varchar(30)` : データベースを特定します。
 - `@segmentname, varchar(30)` : セグメントを特定します。
 - `@space_left, int` : スレッシュホルドに関連した空きページ数を特定します。
 - `@status, int` : ラストチャンス・スレッシュホルドでは 1、その他のスレッシュホルドでは 0 です。

これらのパラメータは、名前ではなく、位置に基づいて渡されます。スレッシュールド・プロシージャは、パラメータに対して別の名前を使用できますが、パラメータを宣言するときは、示されている順序で、正確なデータ型を使用してください。

- スレッシュールドごとに別のプロシージャを作成する必要はありません。メンテナンスをできるだけ少なくするためには、**Adaptive Server** 内のすべてのスレッシュールドに対して実行するスレッシュールド・プロシージャを 1 つ **sybsystemprocs** データベースに作成します。
- エラー・ログに出力を送信するには、**print** 文と **raiserror** 文をスレッシュールド・プロシージャに入れます。

スレッシュールド・プロシージャの実行

- スレッシュールドを超えたときに開始されたタスクは、バックグラウンド・タスクとして実行されます。これらのタスクには、対応する端末セッションまたはユーザ・セッションがありません。これらのタスクの動作中に **sp_who** を実行すると、**status** カラムに “background” が表示されます。
- **Adaptive Server** は、ユーザがスレッシュールドを追加した時点で持っていたパーミッションによって、スレッシュールド・プロシージャを実行します。ただし、その時点以降に取り消されたパーミッションは差し引かれます。
- スレッシュールド・プロシージャは、プロシージャの実行中、1 つのユーザ接続を使用します。

スレッシュールドの変更または削除

- 既存のスレッシュールドに関する情報を出力するには、**sp_helpthreshold** を使用します。
- 新しいスレッシュールド・プロシージャ、空き領域の値、またはセグメントにスレッシュールドを関連付けるには、**sp_modifythreshold** を使用します (ラストチャンス・スレッシュールドに関連付けられている空き領域の値やセグメント名は変更できません)。

ユーザがスレッシュールドを変更すると、ユーザはスレッシュールドの所有者になります。スレッシュールドを超えると、**Adaptive Server** はそのスレッシュールドを所有するユーザがスレッシュールドを変更した時点で持っていたパーミッションによって、スレッシュールド・プロシージャを実行します。ただし、その時点以降に取り消されたパーミッションは差し引かれます。

- セグメントからスレッシュールドを削除するには、**sp_droptreshold** を使用します。

空き領域の計算を無効にする

警告！ 空き領域の計算が無効になっている場合、システム・プロシージャは領域の割り付けについての正確な情報を提供できません。

- ログ以外のセグメントで空き領域の計算を無効にするには、[sp_dboption](#) の `no free space acctg` オプションを使用します。
- ログ・セグメントの空き領域の計算を無効にすることはできません。

パーミッション

`sp_addthreshold` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage database</code> 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sso_role</code> を持つユーザーであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル スレッシュホールドの使用法の詳細については、『システム管理ガイド』を参照してください。

コマンド [create procedure](#), [dump transaction](#)

関数 [lct_admin](#)

システム・プロシージャ [sp_dboption](#), [sp_droptreshold](#), [sp_helpthreshold](#), [sp_modifythreshold](#), [sp_thresholdaction](#)

sp_addtype

説明	ユーザ定義データ型を作成します。
構文	<pre>sp_addtype <i>typename</i>, <i>phystype</i> [(<i>length</i>) (<i>precision</i> [, <i>scale</i>])] [, "identity" <i>nulltype</i>]</pre>
パラメータ	<p><i>typename</i></p> <p>ユーザ定義データ型の名前です。データ型名は、識別子の規則に準拠し、データベースごとにユニークでなければなりません。</p> <p><i>phystype</i></p> <p>そのユーザ定義データ型の基本になる物理データ型、または Adaptive Server 提供のデータ型です。timestamp を除くすべての Adaptive Server のデータ型を指定できます。</p> <p>char、varchar、unichar、univarchar、nchar、nvarchar、binary、varbinary の各データ型では、<i>length</i> 値をカッコで囲んで指定します。長さを指定しないと、デフォルトで 1 文字に設定されます。</p> <p>float データ型では、バイナリの <i>precision</i> をカッコで囲んで指定します。precision を指定しないと、使用しているプラットフォームのデフォルトの精度が使用されます。</p> <p>numeric データ型と decimal データ型では、10 進数の <i>precision</i> 値と <i>scale</i> 値を、カンマで区切りカッコで囲んで指定します。これらを指定しないと、デフォルトの精度 18 桁と位取り 0 が使用されます。</p> <p>カッコやカンマなどの句読表記を含む物理データ型は、一重引用符または二重引用符で囲みます。</p> <p><i>identity</i></p> <p>ユーザ定義データ型が IDENTITY プロパティを持っていることを表します。identity キーワードは、一重引用符または二重引用符で囲みます。IDENTITY プロパティは、位取りが 0 の numeric データ型だけに指定できます。</p> <p>IDENTITY カラムには、Adaptive Server が生成する連続番号 (送り状番号や従業員番号など) が保管されます。IDENTITY カラムの値は、テーブル内の各ローをユニークに識別します。IDENTITY カラムは更新できません。また、null 値は許容されません。</p>

nulltype

ユーザ定義データ型での null 値エントリの処理方法を指定します。このパラメータの許容値は null、NULL、nonnull、NONULL、"not null"、"NOT NULL" です。nulltype に空白スペースを含む値を指定する場合は、値を一重引用符か二重引用符で囲みます。

IDENTITY プロパティと nulltype のどちらも指定しないと、データベースで定義されている null モードを使用してデータ型が作成されます。デフォルトでは、nulltype が指定されていないデータ型は、NOT NULL で作成されます(つまり、null 値は許容されないため、明示的なエントリが必要になります)。SQL 標準に準拠するために、sp_dboption システム・プロシージャを使用して allow nulls by default オプションを true に設定します。この設定によって、データベースの null モードが NULL に変更されます。

例 **例 1** 社会保障番号を保持するカラムに使用する、ssn という名前のユーザ定義データ型を作成します。nulltype パラメータを指定していないため、このデータ型はデータベースのデフォルトの null モードで作成されます。varchar(11) が引用符で囲まれていることに注意してください。これは、この指定に句読表記(カッコ)が含まれているためです。

```
sp_addtype ssn, "varchar(11)"
```

例 2 null 値を許容する birthday という名前のユーザ定義データ型を作成します。

```
sp_addtype birthday, "datetime", null
```

例 3 温度情報の保管に使用する、temp52 という名前のユーザ定義データ型を作成します。この温度情報は、小数点以下 2 桁の最大 5 桁の有効数字で構成されます。

```
sp_addtype temp52, "numeric(5,2)"
```

例 4 IDENTITY プロパティを持つ、row_id という名前のユーザ定義データ型を作成します。このデータ型はローのユニークな識別子として使用します。このデータ型で作成するカラムには、最大 10 桁の長さのシステム生成値を保管します。

```
sp_addtype "row_id", "numeric(10,0)", "identity"
```

例 5 sysname を基本データ型とするユーザ定義データ型を作成します。create table 文、alter table 文、create procedure 文では、sysname データ型を使用できませんが、sysname を基本データ型とするユーザ定義データ型は使用できます。

```
sp_addtype systype, sysname
```

使用法

- `sp_addtype` では、ユーザ定義データ型が作成され、`systypes` システム・テーブルに追加されます。作成したユーザ定義データ型は、`create table` 文や `alter table` 文で使用して、デフォルトとルールをそのデータ型にバインドすることができます。
- 各ユーザ定義データ型は、Adaptive Server が提供するデータ型のいずれかを基にして作成し、必要に応じて長さ、精度、位取りを指定します。`create table` 文または `alter table` 文では、長さ、精度、位取りを上書きすることはできません。
- ユーザ定義データ型の名前はデータベース内でユニークでなければなりません、名前の異なるユーザ定義データ型が同じ定義を持つことはできます。
- `phystype` に `nchar` 型または `nvarchar` 型を指定した場合、新しいデータ型で作成されるカラムの最大長は、`sp_addtype` で指定した長さに、データ型を追加した時点で `@@ncharsize` の値を乗算した長さになります。
- `phystype` に `unichar` 型または `univarchar` 型を指定した場合、新しいデータ型で作成されるカラムの最大長は、`sp_addtype` に指定した長さに、データ型を追加した時点で値 2 を乗算した長さになります。
- 各システム・データ型には「階層」があります。この階層は `systypes` システム・テーブルに保管されています。ユーザ定義データ型の階層は、その基本データ型となる物理データ型の階層と同じになります。混合モードの式では、すべてのデータ型が、最低位階層の型である共通のデータ型に変換されます。

データベースで使用されているシステム提供のデータ型とユーザ定義データ型の階層を出力するには、次のクエリを使用します。

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

IDENTITY プロパティによるデータ型

- ユーザ定義データ型に `IDENTITY` プロパティが設定されている場合、そのユーザ定義データ型を基に作成されるカラムはすべて `IDENTITY` カラムになります。`create` 文または `alter table` 文で、`IDENTITY` と `NOT NULL` のどちらかを指定できます。または、両方とも指定しなくてもかまいません。次に、`IDENTITY` プロパティを持つユーザ定義データ型から `IDENTITY` カラムを作成する 3 通りの方法を示します。

```
create table new_table (id_col Identity)
create table new_table (id_col Identity identity)
```

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

- `create table` 文または `alter table` 文でカラムを作成するときには、`sp_addtype` システム・プロシージャで指定した `null` データ型を上書きできます。
- `NOT NULL` として指定されたデータ型を使用して、`NULL` または `IDENTITY` カラムを作成できます。
- `NULL` として指定されたデータ型を使用して、`NOT NULL` カラムを作成することができますが、`IDENTITY` カラムは作成できません。

注意 `IDENTITY` 型を基に `null` カラムを作成しようとする、`create` 文と `alter table` 文はいずれもエラーになります。

パーミッション

`sp_addtype` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [create default](#), [create rule](#), [create table](#)

データ型 ユーザ定義型

システム・プロシージャ [sp_bindefault](#), [sp_bindrule](#), [sp_dboption](#), [sp_droptype](#), [sp_rename](#), [sp_unbindefault](#), [sp_unbindrule](#)

sp_addumpdevice

説明	Adaptive Server にダンプ・デバイスを追加します。
構文	<code>sp_addumpdevice {"tape" "disk"}, logicalname, physicalname [, tapesize]</code>
パラメータ	<p>"tape" テープ・ドライブ用です。tape は引用符で囲んでください。</p> <p>"disk" ディスク、またはファイル・デバイス用です。disk は引用符で囲んでください。</p> <p><i>logicalname</i> ダンプ・デバイスの「論理」名です。この名前は、有効な識別子でなければなりません。sysdevices にダンプ・デバイスを追加すると、load コマンドと dump コマンド内でダンプ・デバイスの論理名を指定できます。</p> <p><i>physicalname</i> デバイスの物理名です。絶対パス名または相対パス名を指定できます。ダンプおよびロード時に、Backup Server は Adaptive Server の現在の作業ディレクトリの位置から相対パス名を解析します。名前に英数字以外の文字が含まれている場合は、名前全体を引用符で囲みます。UNIX プラットフォームでは、ノンリワインド・テープ・デバイス名を指定します。</p> <p><i>tapesize</i> テープ・ダンプ・デバイスの容量 (MB 単位) です。全てのプラットフォームにおいて、テープ・デバイス用にこのパラメータが必要ですが、ディスク・デバイスの場合には無視されます。tapesize は、最低でも 5 データベース・ページにしてください (1 ページあたり 2048 バイトが必要)。使用するデバイスの定格容量を少し下回る容量を指定することをおすすめします。</p>
例	<p>例 1 40MB のテープ・デバイスを追加します。dump コマンドと load コマンドでは、デバイスの参照に、物理名 <code>/dev/nrmt8</code> または論理名 <code>mytapedump</code> を使用できます。</p> <pre>sp_addumpdevice "tape", mytapedump, "/dev/nrmt8", 40</pre> <p>例 2 <code>mydiskdump</code> という名前のディスク・デバイスを追加します。絶対パス名または相対パス名と、ファイル名を指定します。</p> <pre>sp_addumpdevice "disk", mydiskdump, "/dev/rxyld/dump.dat"</pre>

使用法

- `sp_addumpdevice` では、`master.dbo.sysdevices` テーブルにダンプ・デバイスが追加されます。テープ・デバイスの `cntrctype` には 3 が割り当てられ、ディスク・デバイスの `cntrctype` には 2 が割り当てられます。
- オペレーティング・システム・ファイルをダンプ・デバイスとして使用するには、`disk` 型のデバイスを指定し、`physicalname` に絶対パス名または相対パス名を指定します。`tapesize` パラメータは指定しません。相対パス名を指定した場合、`dump` コマンドまたは `load` コマンドが実行されると、ダンプ先およびロード元は、Adaptive Server の現在の作業ディレクトリになります。
- 所有権とパーミッションの問題によって、ディスクまたはファイルのダンプ・デバイスを正常に使用できないことがあります。`sp_addumpdevice` では、デバイスが `sysdevices` テーブルに追加されるだけです。ファイルをダンプ・デバイスとして作成できるかどうか、ユーザが特定のデバイスにダンプできるかどうかは保証されません。
- `sp_addumpdevice` の `tapesize` に指定したテープ・サイズは、`dump database` コマンドと `dump transaction` コマンドの `with capacity = megabytes` 句を使用して上書きできます。テープ終了マークを確実に検出できないプラットフォームでは、指定したメガバイト数のダンプが行われてから、Backup Server によってボリューム変更要求が発行されます。
- ダンプ・デバイスに障害が発生した場合は、`sp_dropdevice` を使用して、そのダンプ・デバイスを `sysdevices` から削除します。デバイスを置き換えてから、`sp_addumpdevice` を使用して論理デバイス名を新しい物理デバイスに関連付けます。これによって、ダンプ・デバイスが失敗するたびに、バックアップ・スクリプトとスレッシュホールド・プロシージャが更新されるのを防ぐことができます。
- データベース・デバイスを `sysdevices` に追加するには、`disk init` コマンドを使用します。

パーミッション

`sp_addumpdevice` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage disk</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sso_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [disk init](#), [dump database](#), [dump transaction](#), [load database](#), [load transaction](#)

システム・プロシージャ [sp_dropdevice](#), [sp_helpdevice](#)

sp_adduser

説明	新しいユーザを現在のデータベースに追加します。
構文	<code>sp_adduser loginname [, name_in_db [, grpname]]</code>
パラメータ	<p><i>loginname</i> master.dbo.syslogins にあるユーザ名です。</p> <p><i>name_in_db</i> 現在のデータベースの新しいユーザ名です。</p> <p><i>grpname</i> データベースの既存のグループにユーザを追加します。</p>
例	<p>例 1 データベースにユーザ “margaret” を追加します。このユーザのデータベース・ユーザ名は Adaptive Server のログイン名と同じで、デフォルト・グループ “public” に属します。</p> <pre>sp_adduser margaret</pre> <p>例 2 データベースにユーザ “haroldq” を追加します。“haroldq” が現在のデータベースを使用するとユーザ名は “harold” になります。このユーザは、デフォルト・グループ “public” の他に、fort_mudge グループにも属します。</p> <pre>sp_adduser haroldq, harold, fort_mudge</pre>
使用法	<ul style="list-style-type: none">データベース所有者は、sp_adduser を実行してユーザ名を現在のデータベースの sysusers テーブルに追加し、そのユーザが自分のユーザ名で現在のデータベースにアクセスできるようにします。name_in_db パラメータを指定すると、新しいユーザに、そのユーザの Adaptive Server ログイン名とは異なる、データベースでの名前が設定されます。異なる名前をユーザに割り当てるこの機能は、便宜上提供されているものです。このユーザ名は、別のユーザの ID や権限にマップされるわけではないため、sp_addalias で設定されるエイリアスとは異なります。ユーザとグループを、同じ名前にすることはできません。ユーザは、デフォルトのグループ “public” 以外に、もう 1 つだけ別のグループのメンバになることができます。すべてのユーザは、デフォルト・グループ “public” のメンバです。ユーザのグループを変更するには、sp_changegroup を使用します。

- ユーザがデータベースにアクセスするには、そのユーザのエントリが `sysusers` に作成されているか (`sp_adduser` を使用)、`sysalternates` で他のユーザにマップされているか (`sp_addalias` を使用)、または `sysusers` に “`guest`” エントリが存在しなければなりません。

パーミッション `sp_adduser` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any user</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者、 <code>sa_role</code> を持つユーザ、または <code>sso_role</code> を持つユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 コマンド `grant`, `revoke`, `use`

システム・プロシージャ `sp_addalias`, `sp_addgroup`, `sp_changegroup`, `sp_dropalias`, `sp_dropgroup`, `sp_helpuser`

sp_altermessage

説明	Adaptive Server エラー・ログ内の特定のシステム定義メッセージまたはユーザ定義メッセージのロギングを、有効にしたり無効にしたりします。
構文	<code>sp_altermessage message_id, parameter, parameter_value</code>
パラメータ	<p><i>message_id</i> 変更するメッセージのメッセージ番号です。これは、<code>sysmessages</code> または <code>sysusermessages</code> システム・テーブルの <code>error</code> カラムに記録されているメッセージ番号です。</p> <p><i>parameter</i> 変更するメッセージ・パラメータです。最大長は 30 バイトです。有効なパラメータは、<code>with_log</code> だけです。</p> <p><i>parameter_value</i> <i>parameter</i> に指定したパラメータの新しい値です。最大長は 5 バイトです。有効な値は <code>true</code> と <code>false</code> です。</p>
例	<p><code>sysmessages</code> にあるメッセージ番号 2000 のメッセージのログを、Adaptive Server のエラー・ログと Windows NT イベント・ログ (ロギングが有効になっている場合) に記録するよう指定します。</p> <pre>sp_altermessage 2000, 'with_log', 'TRUE'</pre>
使用法	<ul style="list-style-type: none"> <i>parameter_value</i> に <code>true</code> を設定した場合、指定したメッセージのログが常に記録されます。<code>false</code> の場合は、デフォルトのロギング動作が行われます。エラーの重大度やその他の要因によって、メッセージのログ記録が行われたり行われなかったりします。<i>parameter_value</i> を <code>false</code> に設定すると、<code>sp_altermessage</code> を呼び出さなかったときと同じ動作になります。 Windows NT サーバでも、<code>sp_altermessage</code> は、Windows NT イベント・ログでのロギングを有効にしたり、無効にしたりします。
パーミッション	<code>sp_altermessage</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、データベース所有者か <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> を持つユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addmessage](#), [sp_dropmessage](#)

sp_audit

説明 システム・セキュリティ担当者による監査オプションの設定を可能にします。

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

または

`sp_audit 'restart'`

パラメータ

option

設定する監査オプションの名前です。表 1-3 に、有効な監査オプションを示します。

表 1-3: sp_audit の監査オプション

オプション	説明
adhoc	ユーザが <code>sp_addauditrecord</code> を使用して、独自のユーザ定義監査レコードを監査証跡に追加できるようにする。
all	特定のユーザや特定の役割を持つユーザが実行するすべての動作を監査する。このオプションでは、システム役割のみ指定可能。 注意 すべての動作を監査しても、ユーザが特定の監査レコードを追加できるかどうかには影響しません。
alter	<code>alter table</code> または <code>alter database</code> コマンドの実行を監査する。
bcp	<code>bcp in</code> ユーティリティの実行を監査する。
bind	システム・プロシージャ <code>sp_bindefault</code> 、 <code>sp_bindmsg</code> 、 <code>sp_bindrule</code> の実行を監査する。
cluster	クラスタ・コマンドを監査する。
cmdtext	特定のユーザのすべての動作を監査する。 注意 セキュリティのために、システム・ストアド・プロシージャとコマンドのパスワード・パラメータはアスタリスクの固定長文字列で置き換えることができます。詳細については、「システム・ストアド・プロシージャとコマンドのパスワード・パラメータを隠す」(80 ページ)を参照してください。
create	データベース・オブジェクトの作成を監査する。
dbaccess	別のデータベースから現在のデータベースへのアクセスを監査する。
dbcc	任意の <code>dbcc</code> コマンドの実行を監査する。
delete	テーブルまたはビューからのローの削除を監査する。
disk	<code>disk init</code> 、 <code>disk refit</code> 、 <code>disk reinit</code> 、 <code>disk mirror</code> 、 <code>disk unmirror</code> 、 <code>disk remirror</code> の実行を監査する。
drop	データベース・オブジェクトの削除を監査する。
dump	<code>dump database</code> または <code>dump transaction</code> の実行を監査する。

オプション	説明
encryption_key	create encryption key、sp_encryption、drop encryption key、alter encryption key を監査する。
errors	エラー (致命的なエラーかどうか) を監査する。
exec_procedure	ストアド・プロシージャの実行を監査する。
exec_trigger	トリガの実行を監査する。
func_dbaccess	Transact-SQL 関数によるデータベースへのアクセスを監査する。
func_obj_access	Transact-SQL 関数によるデータベース・オブジェクトへのアクセスを監査する。
grant	grant の実行を監査する。
insert	テーブルまたはビューへのローの挿入を監査する。
install	Java クラスのインストールを監査する。
load	load database または load transaction の実行を監査する。
login	Adaptive Server へのすべてのログイン試行を監査する。
login_locked	ログイン試行の失敗回数が所定の限度を超えたためにログイン・アカウントがロックされた場合に、ホスト名とネットワーク IP アドレスを監査する。
logout	Adaptive Server からのすべてのログアウト試行を監査する。
mount	mount database コマンドを監査する。
network	特定のネットワーク関連イベント (リスナ・イベントなど) を監査する。有効な設定は次のとおり。 <ul style="list-style-type: none"> • on • off • pass • Failed
quiesce	quiesce database コマンドを監査する。
reference	テーブル間の参照を監査する。
remove	Java クラスの削除を監査する。
revoke	revoke の実行を監査する。
rpc	リモート・プロシージャ・コールの実行を監査する。

オプション	説明
security	<p>次のようなセキュリティ関連イベントを監査する。</p> <ul style="list-style-type: none"> • サーバの起動または停止。 • 役割のアクティブ化またはアクティブ化解除。 • 次のいずれかのコマンドの発行。 <ul style="list-style-type: none"> addcert connect create と drop login dropcert create、drop、alter、grant、revoke role kill online database set proxy set session authorization sp_configure • 表 1-4 (77 ページ) にリストされている関数。 • 次のいずれかの関数の使用。 <ul style="list-style-type: none"> valid_user proc_role (システム・プロシージャ内から) • SSO パスワードの再生成。
select	select の実行を監査する。
setuser	setuser の実行を監査する。
table_access	特定のユーザによる任意のテーブルへのアクセスを監査する。
transfer table	transfer table コマンドの実行を監査する。
truncate	truncate table の実行を監査する。
unbind	sp_unbindrule 、 sp_unbindmsg 、 sp_unbindefault の実行を監査する。
unmount	umount database コマンドの実行を監査する。
update	テーブルまたはビュー内のローの更新を監査する。
view_access	特定のユーザによる任意のビューへのアクセスを監査する。

表 1-4: sp_audit セキュリティが監査する関数

- | | | |
|-----------------------|--------------------|------------------------|
| • config_admin | • set_password | • valid_user |
| • attr_notify | • ha_add_companion | • ha_remove_companion |
| • ha_check_alive | • ha_getversion | • ha_failback |
| • ha_restrictionclass | • ha_getrcs | • ha_setrcs |
| • ha_hacluster_verify | • js_wakeup | • unlock_admin_account |
| • ssl_admin | • ws_admin | |

login_name

このパラメータには、**all**、システム標準の役割、または特定のログイン名を監査対象として指定できます。ただし、システム標準の役割を指定できるのは、**all** オプションを指定した場合だけです。システム役割に対して、個々のオプションを監査することはできません。

object_name

監査するオブジェクトの名前です。有効な値は、*option* に指定した値によって異なります。次に、有効な値を示します。

- オブジェクト名 (そのオブジェクトの所有者でない場合は、所有者の名前も含める)。たとえば、Joe が所有する **inventory** という名前のテーブルを監査するには、*object_name* に **joe.inventory** を指定します。
- すべてのオブジェクトを対象とする場合は **all**。
- 新しいテーブル、ビュー、プロシージャ、トリガへのアクセスを監査する場合は、**default table**、**default view**、**default procedure**、**default trigger**。
default table と **default view** は、*option* パラメータに **delete**、**insert**、**select**、または **update** を指定したときの *object_name* の有効な値です。**default procedure** は、**exec_procedure** オプションを指定したときの有効な値です。**default trigger** は **exec_trigger** オプションを指定したときの有効な値です。
- **network** – 特定のネットワーク関連イベント (リスナ・イベントなど) を監視します。

各 *option* 値によって有効になる *object_name* の値の詳細については、『システム管理ガイド』を参照してください。

setting

監査のレベルです。**setting** の値を指定しないと、オプションの現在の監査設定が表示されます。**setting** パラメータに有効な値は、次のとおりです。

- **on** — 指定したオプションの監査をアクティブ化する。**Adaptive Server** は、イベントがパーミッションの検査にパスするかどうかに関係なく、このオプションが制御するイベントの監査レコードを作成する。
- **off** — 指定したオプションの監査のアクティブ化を解除する。
- **pass** — パーミッション検査に合格したイベントの監査をアクティブ化する。
- **fail** — パーミッション検査で不合格になったイベントの監査をアクティブ化する。

オプションに **pass** を指定して、後で同じオプションに **fail** を指定しても、またはこの逆の順序で指定しても、結果は **on** を指定した場合と同じになります。**Adaptive Server** は、イベントがパーミッション検査に合格するか不合格になるかに関係なく監査レコードを生成します。

各設定は次のように適用されます。

- **on** または **off** — すべての監査オプションに適用されます。
- **pass** と **fail** — **cmdtext**、**errors**、および **adhoc** を除くすべてのオプションに適用されます。これらのオプションには、**on** または **off** だけが適用されます。すべてのオプションのデフォルトの値は **off** です。**pass** または **fail** のいずれかに **cmdtext** オプションを選択すると、**Adaptive Server** は値を **on** に置き換えます。

restart

エラーが発生したために監査プロセスが強制的に終了された場合は、次のように入力することで **sp_audit** を手動で再起動できます。

```
sp_audit restart
```

監査プロセスは、現在実行中の監査はないが、**sp_configure "auditing" 1** の入力により監査プロセスを実行するように設定されていることを条件として再起動できます。

例 **例 1** ログイン試行の失敗回数が所定の限度を超えたためにログイン・アカウントがロックされた場合に、ホスト名とネットワーク IP アドレスの監査を開始するよう **login_locked** 監査オプションを設定します。

```
sp_audit "login_locked","all","all","ON"
```


監査テーブルが満杯でイベントのログを取ることができない場合は、その情報がエラーログにメッセージとして送信されます。

監査ログを使用して **Locked Login** イベント (112) をモニタリングすると、ログイン・アカウントに対する攻撃の識別に役立ちます。

例 2 **SSL** セキュリティ関連イベントに対する監査を開始します。成功したイベントと失敗したイベントの両方が監査されます。

```
sp_audit "security", "all", "all", "on"
```

```
sample records added:
```

sybsecurity のイベントを確認するには、次のように指定します。

```
select * from sybsecurity..sysaudits_01 where event=99
```

例 3 **security** 監査オプションの設定内容を表示します。

```
sp_audit "security", "all", "all"
```

例 4 **create database** を含めて、**master** データベース内でのオブジェクトの作成に対する監査を開始します。

```
sp_audit "create", "all", master, "on"
```

例 5 **pubs2** データベース内のコマンドを監査します。

```
sp_audit "encryption_key", "all", "pubs2", "on"
```

例 6 **db1** データベース内でのすべてのオブジェクトの作成に対する監査を開始します。

```
sp_audit "create", "all", db1, "on"
```

例 7 システム管理者が失敗したすべての実行に対する監査を開始します。

```
sp_audit "all", "sa_role", "all", "fail"
```

例 8 現在のデータベース内で、今後、テーブルに対して実行される更新処理に関する監査を開始します。たとえば、現在のデータベースが **utility** である場合、**utility** 内に新規作成されるすべてのテーブルについて、更新処理に対する監査が行われます。これは既存のテーブルに対する監査には影響しません。

```
sp_audit "update", "all", "default table", "on"
```

例 9 **titles** テーブルに対して入力されたすべての **transfer table** コマンドの監査を開始します。

```
sp_audit "transfer table", "all", "titles", "on"
```

使用法

- sp_audit では、監査が有効になった場合の監査対象が設定されます。sp_configure を使用して auditing パラメータを on に設定するまでは、実際の監査は行われません。この設定を行うと、sp_audit を使用して設定したすべての監査オプションが有効になります。詳細については、sp_configure を参照してください。
- 監査対象として指定されているオブジェクトの所有者でない場合は、次のフォーマットに従い、object_name パラメータの値に所有者名を付けます。

```
"ownername.objname"
```

- tempdb データベースでは、次のオプションのデフォルトの監査機能をアクティブにすることはできません。

delete	insert	select
update	exec_procedure	exec_trigger

- 監査機能を制御する設定パラメータは次のとおりです。
 - auditing — サーバの監査を有効または無効にします。
 - audit queue size — 監査キューのサイズを設定します。
 - current audit table — 現在の監査テーブルを設定します。Adaptive Server は、すべての監査レコードを現在の監査テーブルに書き込む。
 - suspend auditing when full — 監査デバイスが満杯になったときの監査プロセスの動作を制御します。

各監査の設定パラメータはすべて動的で、すぐに有効になります。

システム・ストアド・プロシージャとコマンドのパスワード・パラメータを隠す

監査が設定されて有効になっているとき、sp_audit にオプションの 'cmdtext' が設定されていると、監査ログ内の監査レコードではシステム・ストアド・プロシージャとコマンドのパスワード・パラメータが固定長のアスタリスク文字列で置き換えられます。

たとえば、監査が有効になっていて sp_audit cmdtext が設定されている場合の出力は次のようになります。

```
sp_password 'oldpassword', 'newpassword'
```

次のような出力が返されます。

```
sp_password '*****', '*****'
```

これで監査ログにアクセスできる他のユーザにパスワードを見られる心配がなくなります。

パーミッション

sp_audit のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage auditing 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sso_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

参照

マニュアル Adaptive Server を設定して監査機能を使用する方法の詳細については、『システム管理ガイド』の sp_configure を参照してください。

システム・プロシージャ sp_addauditrecord、sp_configure、sp_addauditable
ユーティリティ・コマンド bcp

sp_autoconnect

説明	(コンポーネント統合サービスのみ) 特定のユーザが使用するリモート・サーバへの接続パススルーを定義します。これによって、このユーザはログイン時に自動的にパススルー・モードを開始できます。
構文	<code>sp_autoconnect server, {true false} [, loginame]</code>
パラメータ	<p><i>server</i> 自動パススルー接続を確立する宛先サーバ名です。 <i>server</i> は、sp_addserver によってすでに追加されているリモート・サーバ名でなければなりません。このサーバに、ローカル・サーバは指定できません。</p> <p><i>true false</i> <i>server</i> に指定したサーバに対して自動パススルー接続を有効にするか無効にするかを指定します。自動パススルー接続は、true を指定すると有効になり、false を指定すると無効になります。</p> <p><i>loginame</i> 自動接続を必要とするユーザの名前を指定します。 <i>loginame</i> を指定しなければ、自動接続ステータスが現在のユーザ用に修正されます。</p>
例	<p>例 1 現在のユーザの次回ログイン時に、このユーザをサーバ SYBASE に自動的に接続します。この接続はパススルー・モードで行います。</p> <pre>sp_autoconnect SYBASE, true</pre> <p>例 2 “steve” というユーザの自動接続機能を無効にします。</p> <pre>sp_autoconnect SYBASE, false, steve</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_autoconnect</code> では、特定のユーザが使用するリモート・サーバへの接続パススルーが定義されます。これによって、このユーザはログイン時に自動的にパススルー・モードを開始できます。 • <code>sp_autoconnect</code> は、コンポーネント統合サービスがインストールされ設定されているときのみ使用できます。 • “sa” ログイン・アカウントの自動接続ステータスは変更しないでください。 • 自動接続ステータスの変更内容は、現在接続しているユーザにはすぐに反映されません。変更内容を反映するには、ローカル・サーバを切断して、再接続する必要があります。 • パススルー・モードを終了するには、<code>disconnect</code> を使用します。

パーミッション	<code>sp_autoconnect</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any login</code> 権限を持つユーザーであることが必要。 自分自身に対しては、すべてのユーザーが <code>sp_autoformat</code> を実行できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザーであることが必要。 自分自身に対しては、すべてのユーザーが <code>sp_autoformat</code> を実行できる。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 コマンド [connect to...disconnect](#), [grant](#)
システム・プロシージャ [sp_addlogin](#), [sp_addserver](#), [sp_passthru](#), [sp_remotesql](#)

sp_autoformat

説明 sp_autoformat は、判読しやすい結果セット・データを作成するユーティリティ・ストアド・プロシージャです。可変長の文字データの幅を再フォーマットして、非ブランク文字のみを表示します。出力の後続ブランクはトランケートされます。

構文 sp_autoformat *fulltablename* [, *selectlist*, *whereclause*, *orderby*]

パラメータ

fulltablename

データを選択するテーブルの名前を指定します。オブジェクトの所有者とコマンドを実行中のユーザが異なる場合は、所有者の名前を使用します。

selectlist

結果セットに選択するカラムをカンマ区切りのリストで指定します。テーブルのカラムの名前は、*<name> = <column>* という形式に従って変更できます。例を参照してください。*selectlist* を指定しない場合、指定したテーブルのすべてのカラムがカラム ID の順序で出力されます。

whereclause

where 句として指定する検索の述部であり、選択するテーブルからローをフィルタします。

orderby

結果セットの出力順を指定する *order by* 句であり、省略可能です。

例 **例 1** `select id, colid, name from syscolumns where id = 3` に似た `select` 文から結果セットを返します。文字カラムは自動フォーマットされます。

```
1> sp_autoformat "syscolumns", "id, colid, name", "where id = 3"
2> go
```

id	colid	name
-----	-----	-----
3	1	id
3	2	number
3	3	colid
3	4	status
3	5	type
3	6	length
3	7	offset
3	8	usertype
3	9	cdefault
3	10	domain
3	11	name
3	12	printfmt

```

3      13 prec
3      14 scale
3      15 remote_type
3      16 remote_name
3      17 xstatus
3      18 xtype
3      19 xdbid
3      21 accessrule
3      22 status2

```

例 2 次の構文を使用して、出力カラムの名前を変更します。

```
[ < AS-Name label of Column> ][ ]*=[ ]*<column name>
```

<AS-Name label of Column> は省略可能です。= セパレータの前後にはスペースを使用できます。

```

1> sp_autoformat syscolumns, "'Object Id' = id, 'Column Name'=name,
   'Column ID'=colid", "where id = 3"
2> go

```

Object Id	Column Name	Column ID
3	id	1
3	number	2
3	colid	3
3	status	4
3	type	5
3	length	6
3	offset	7
3	usertype	8
3	cdefault	9
3	domain	10
3	name	11
3	printfmt	12
3	prec	13
3	scale	14
3	remote_type	15
3	remote_name	16
3	xstatus	17
3	xtype	18
3	xdbid	19
3	accessrule	21
3	status2	22

(1 row affected)

例 3 *orderby* パラメータを使用して、結果の出力の順序を指定します。

```
sp_autoformat @fulltabname = 'syscolumns',
              @selectlist = "id, name",
              @orderby = "ORDER BY name"
```

例 4 複数のテーブルから選択するときに自動フォーマットされた結果を生成する場合や、**select** のリストで式を使用した複雑な SQL **select** 文を使用する場合は、次を行う必要があります。

- 1 テンポラリ・テーブルを使用して結果セットを生成します。

次の文では、すべてのシステム・カタログでカラム ID が最も大きいカラムのリストが生成されます。

```
select o.id, o.name, c.colid, c.name
from sysobjects o, syscolumns c
where o.id < 100 and o.id = c.id
      and c.colid = (select max(c2.colid) from syscolumns c2
                    where c2.id = c.id)
order by o.name
```

次の文では、同じ結果セットを生成するときに、テンポラリ・テーブルを使用して文字データを自動フォーマットした、判読しやすい出力を作成します。さらに、細かな変更を加えて、テンポラリ・テーブルにカラム名を追加します。

```
select o.id, ObjectName = o.name, c.colid, ColumnName = c.name
into #result
from sysobjects o, syscolumns c
where o.id < 100 and o.id = c.id
      and c.colid = (select max(c2.colid) from syscolumns c2
                    where c2.id = c.id)
```

- 2 そのテンポラリ・テーブルに対して **sp_autoformat** を使用して、フォーマットされた出力を生成します。

元の **select** 文の **order by** 句は、テンポラリ・テーブルを生成するときには省略され、代わりに、出力結果を生成するときの **sp_autoformat** の呼び出しに追加されています。

```
1> exec sp_autoformat @fulltabname = #result, @orderby = "order by
      ObjectName"
2> go
```

id	ObjectName	colid	ColumnName
11	sysalternates	2	altsuid
21	sysattributes	13	comments
55	syscertificates	6	suid
45	syscharsets	8	sortfile


```

3 syscolumns          22 status2
6 syscomments        8 status
37 sysconfigures     9 value4
17 sysconstraints    7 spare2
38 syscurconfigs     15 type
30 sysdatabases      19 status4
12 sysdepends         10 readobj
35 sysdevices        7 mirrorname
43 sysengines        12 starttime

```

...

```

(1 row affected)
(return status = 0)

```

次に示すように、テンポラリ・テーブルの処理をさらに追加して、選択したテーブルの中から必要な出力についてのみレポートするようにできます。

```

1> exec sp_autoformat #result, "id, 'Object Name' = ObjectName,
   'Column Name' = ColumnName", "where id < 5"
2> go

```

```

id      Object Name Column Name
-----
1 sysobjects loginame
2 sysindexes crdate
3 syscolumns status2
4 systypes  accessrule

```

使用法

- Adaptive Server バージョン 15.0.3 以降では、`sp_autoformat` は `int` (`smallint`、`bigint`、`tinyint`、`unsigned int`)、`numeric`、`money`、`date/time`、`float`、`real`、`double` 精度のデータ型のカラムが受け入れられるようになりました。
- `sp_autoformat` は、現在のデータベース内にあるオブジェクトのみを検索します。`sp_autoformat` をテンポラリ・テーブルに対して使用するには、`tempdb` からプロシージャを実行します。
- `sp_autoformat` では、いずれかのパラメータで参照しているカラムが、`fulltablename` パラメータで指定したテーブルに実際に存在するかどうかは検証されません。存在しないカラムを参照すると、`sp_autoformat` は失敗します。
- `select` リストでは、各列のインスタンスは 1 つのみ指定します。

リターン・コード

- 0 – 正常終了。
- 1 – 内部エラー、または呼び出しの使用上のエラー。
- その他 – 生成された SQL 文の実行の間に Adaptive Server で発生した、他のすべてのエラーは、呼び出し元に返されます。

制限事項

- sp_autoformat では、内部 SQL 変数を使用して SQL 文を生成し、execute immediate を使用してその文を実行します。生成される SQL 文の長さは 2K バイトに制限されています。大量のカラム・リストや長い名前を持つカラムに対して、結果セットの自動フォーマットを行うと、生成された SQL 文を格納するバッファのサイズ不足によってエラーが生じることがあります。
- 引用符付き識別子は、テーブルとカラム名のいずれに対してもサポートされていません。引用符付き識別子を使用する結果セットがあり、それに対する自動フォーマットが必要な場合には、次のようにする必要があります。
 - a 必要なデータをテンポラリ・テーブルに生成します。テンポラリ・テーブルのカラムでは、引用符付き識別子は使用しないようにします。
 - b sp_autoformat を使用し、テンポラリ・テーブルを使用して目的の出力を生成します。
 - c 目的の出力フォーマットで、selectlist でカラムの名前を変更します。

パーミッション

sp_autoformat のパーミッション・チェックは実行されません。パーミッション・チェックは、細密なパーミッションの設定によって異なります。テーブルから選択を実行するユーザには、適切な select 権限が必要です。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

sp_bindcache

説明	データベース、テーブル、インデックス、text オブジェクト、または image オブジェクトをデータベース・キャッシュにバインドします。
構文	<pre>sp_bindcache cachename, dbname [, [ownername.]tablename [, indexname "text only"]]</pre>
パラメータ	<p>cachename アクティブなデータ・キャッシュの名前です。</p> <p>dbname キャッシュにバインドするデータベースの名前、またはキャッシュにバインドするテーブル、インデックス、text や image オブジェクトが格納されているデータベースの名前です。</p> <p>ownername テーブル所有者の名前です。“dbo” がテーブルを所有している場合は、所有者名を省略できます。</p> <p>tablename キャッシュにバインドするテーブルの名前、またはキャッシュにバインドするインデックス、text オブジェクト、image オブジェクトを保管しているテーブルの名前です。</p> <p>indexname キャッシュにバインドするインデックスの名前です。</p> <p>text only text または image オブジェクトをキャッシュにバインドします。このパラメータを使用する場合、インデックス名を同時に指定できません。</p>
例	<p>例 1 titles テーブルを pub_cache という名前のキャッシュにバインドします。</p> <pre>sp_bindcache pub_cache, pubs2, titles</pre> <p>例 2 クラスタード・インデックス titles.title_id_cix を pub_ix_cache にバインドします。</p> <pre>sp_bindcache pub_ix_cache, pubs2, titles, title_id_cix</pre> <p>例 3 pubs2 を tempdb_cache にバインドします。</p> <pre>sp_bindcache tempdb_cache, pubs2</pre> <p>例 4 pubs2 のトランザクション・ログである syslogs を、logcache という名前のキャッシュにバインドします。</p> <pre>sp_bindcache logcache, pubs2, syslogs</pre>

例 5 au_pix テーブルの image チェーンを、pub_cache という名前のキャッシュにバインドします。

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

使用法

- データベースまたはデータベース・オブジェクトは、1つのキャッシュにしかバインドできません。データベースがバインドされているキャッシュとは別のキャッシュに、そのデータベースの個々のテーブル、インデックス、text オブジェクト、または image オブジェクトをバインドできます。データベースのバインドは、他にバインドを持たないそのデータベース内のすべてのオブジェクトに対し、デフォルトのバインドとなります。テーブルまたはインデックスに対するデータ・キャッシュ階層は、次のようになります。
 - オブジェクトがキャッシュにバインドされている場合、オブジェクト・バインドが使用されます。
 - オブジェクト自体はキャッシュにバインドされていないが、そのオブジェクトのデータベースがキャッシュにバインドされている場合、データベースのバインドが使用されます。
 - オブジェクトもそのデータベースもキャッシュにバインドされていない場合、デフォルト・データ・キャッシュが使用されます。
- キャッシュとそれにバインドされるオブジェクトまたはデータベースが存在していないと、sp_bindcache は実行できません。事前に sp_cacheconfig でキャッシュを作成します。そして、オペレーションが動的でない場合は、Adaptive Server を再起動してから、キャッシュへのオブジェクトのバインドを実行します。
- キャッシュのバインドはすぐに有効になります。サーバを再起動する必要はありません。データ・キャッシュにオブジェクトをバインドすると、次のようになります。
 - 現在メモリにあるオブジェクトのページがクリアされます。
 - クエリでオブジェクトを使うと、バインドされたキャッシュにページが読み込まれます。
- インデックスを、それが参照するテーブルとは異なるキャッシュにバインドできます。クラスタード・インデックスをキャッシュにバインドする場合、そのバインドはインデックスのルートおよび中間ページだけに影響します。データ・ページ(定義では、インデックスのリーフ・ページ)には影響しません。

- データベースをバインドするには、**master** データベースを使用します。テーブル、インデックス、**text** オブジェクトまたは **image** オブジェクトをバインドするには、それらのオブジェクトが保管されているデータベースを使用します。
- データベース内の任意のシステム・テーブルをバインドするには、そのデータベースを使用し、そのデータベースをシングルユーザ・モードに設定してください。次のコマンドを使用します。

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

詳細については、[sp_dboption](#) を参照してください。

- オブジェクトまたはデータベースを異なるキャッシュへバインドするために、それらのバインドを解除する必要はありません。すでにバインドされているオブジェクトに **sp_bindcache** を発行すると、古いバインドが削除され、新しいバインドが作成されます。
- テーブルまたはそのインデックスをキャッシュにバインドする場合、バインドの実行中にページが読み込まれないよう、**sp_bindcache** で排他テーブル・ロックを取得する必要があります。ユーザがテーブルをロックしているときに、そのテーブル・オブジェクトに対して **sp_bindcache** を発行した場合、バインド処理はロックが解除されるまでスリープします。
- オブジェクトをバインドする、またはバインドを解除すると、そのオブジェクトを参照するストアド・プロシージャはすべて、次に実行されるときに再コンパイルされます。データベースのバインドを変更すると、そのバインドされたデータベースのオブジェクトを参照するストアド・プロシージャはすべて、次に実行されるときに再コンパイルされます。
- テーブル、インデックス、またはデータベースを削除する場合、対応するキャッシュ・バインドもすべて削除されます。テーブル、インデックス、またはデータベースを再作成する場合、それらをキャッシュにバインドするには **sp_bindcache** をもう一度使用します。
- データベースまたはデータベース・オブジェクトがバインドされているキャッシュを削除すると、そのキャッシュ・バインドは無効とマーク付けされますが、**sysattributes** システム・テーブルには残ります。**Adaptive Server** を再起動すると、エラー・ログに警告が出力されます。同じ名前のキャッシュを作成すると、**Adaptive Server** を再起動したときに、無効になっていたバインドが有効になります。

- 次の各プロシージャでは、それぞれ対応するオブジェクトのバインド情報が出力されます。`sp_helpdb` ではデータベースのバインド、`sp_help` ではテーブルのバインド、`sp_helpindex` ではインデックスのバインドに関する情報が出力されます。また、`sp_helpcache` では特定のキャッシュにバインドされたすべてのオブジェクトに関する情報が出力されます。
- テーブルとインデックスの現在のサイズを調べるには、`sp_spaceused` を使用します。また、増大するテーブルのサイズを見積もるには、`sp_estspace` を使用します。キャッシュのサイズとステータスに関する情報を確認するときや、キャッシュを設定または再設定するときは、`sp_cacheconfig` を使用します。
- システム `tempdb` 上で `sp_bindcache` を引き続き実行できますが、システム `tempdb` のバインドは、動的ではありません。サーバを再起動するまでは、次のようになります。
 - 変更が有効にならない。
 - `sp_helpcache` で、保留を表すステータス “P” がレポートされる。ただし、システム `tempdb` をデフォルトのデータ・キャッシュに明示的にバインドした場合は、有効を表すステータス “V” がレポートされる。これは、デフォルトでシステム `tempdb` はデフォルトのデータ・キャッシュにすでにバインドされているため。

制限事項

- `master` データベース、`master` 内のシステム・テーブル、`master` 内のシステム・テーブルのインデックスは、キャッシュにバインドできません。`master` 内のシステム・テーブル以外のテーブルやそのインデックスはキャッシュにバインドできます。
- 次の条件では、データベースやオブジェクトをキャッシュにバインドすることはできません。
 - 独立性レベル 0 の読み込みがテーブル上でアクティブな場合。
 - 現在バインドを実行中のタスクのカーソルが、テーブル上でオープンしている場合。
- キャッシュ・タイプが `log only` である場合、そのキャッシュにバインドできるのは `syslogs` テーブルだけです。キャッシュのタイプを確認するには、`sp_cacheconfig` を使用します。

パーミッション `sp_bindcache` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage data cache</code> 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザーであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_cacheconfig](#), [sp_configure](#), [sp_help](#), [sp_helpcache](#), [sp_helpdb](#), [sp_helpindex](#), [sp_poolconfig](#), [sp_unbindcache](#), [sp_unbindcache_all](#)

sp_bindefault

説明 ユーザ定義のデフォルトをカラムまたはユーザ定義データ型にバインドします。

構文 `sp_bindefault defname, objname [, futureonly]`

パラメータ

defname

`create default` 文で作成されたデフォルトの名前です。特定のカラムまたはユーザ定義データ型にバインドします。

objname

デフォルトのバインド先のテーブルとカラム、またはユーザ定義データ型の名前です。`objname` パラメータに指定した値が “`table.column`” のフォーマットになっていない場合は、ユーザ定義データ型とみなされます。オブジェクト名にブランク、句読表記または予約語が含まれている場合は、引用符で囲んでください。

`futureonly` を指定しない場合、ユーザ定義データ型の既存のカラムは、`defname` に指定したデフォルトを継承します。

計算カラムにデフォルトをバインドすることはできません。

futureonly

ユーザ定義データ型の既存のカラムが、新しいデフォルトを取得しないようにします。デフォルトをユーザ定義データ型にバインドする場合、このパラメータは省略可能です。デフォルトをカラムにバインドするためには使用しません。

例 **例 1** `create default` で `today` という名前のデフォルトが現在のデータベースに定義されている場合に、次のコマンドでは、このデフォルトが `employees` テーブルの `startdate` カラムにバインドされます。`employees` テーブルに追加される新しいローの `startdate` カラムには、別の値が入力されないかぎり、デフォルト値 `today` が設定されます。

```
sp_bindefault today, "employees.startdate"
```

例 2 `def_ssn` というデフォルトと `ssn` というユーザ定義データ型が存在する場合に、次のコマンドでは、`def_ssn` が `ssn` にバインドされます。このデフォルトは、テーブルの作成時にこのユーザ定義データ型 `ssn` に割り当てられるすべてのカラムに継承されます。`futureonly` を指定しないかぎり、またはカラムのデフォルトが変更されていないかぎり、データ型 `ssn` の既存のカラムもデフォルト `def_ssn` を継承します (`futureonly` が指定されていると、そのユーザ定義データ型の既存のカラムはデフォルトを継承しません。また、カラムのデフォルトが以前に変更されていると、変更されたデフォルトがそのまま維持されます)。

```
sp_bindefault def_ssn, ssn
```


例 3 ユーザ定義データ型 `ssn` にデフォルト `def_ssn` をバインドします。`futureonly` パラメータを指定しているため、`ssn` データ型の既存のカラムには作用しません。

```
sp_bindefault def_ssn, ssn, futureonly
```

使用法

- カラムのデフォルトを設定するには2つの方法があります。`create table` 文または `alter table` 文でデフォルトをカラムの制約として宣言する方法と、`create default` 文で作成したデフォルトを `sp_bindefault` によってカラムにバインドする方法です。`create default` を使用すると、そのデフォルトをデータベースの複数のカラムにバインドできます。
- Adaptive Server が提供するデータ型には、デフォルトをバインドできません。
- システム・テーブルには、デフォルトをバインドできません。
- `IDENTITY` プロパティを持つカラムまたはユーザ定義データ型にバインドされたデフォルトは、カラムの値には影響しません。ローをテーブルに挿入するたびに、Adaptive Server によって連続番号が順次 `IDENTITY` カラムに割り当てられます。
- カラムにデフォルトをバインドする場合は、`objname` 引数を “`table.column`” のフォーマットで指定します。これ以外のフォーマットは、ユーザ定義データ型の名前とみなされます。
- すでにカラムにデフォルトが存在する場合は、それを削除してから新しいデフォルトをバインドしてください。`sp_bindefault` で作成したデフォルトを削除するには、`sp_unbindefault` を使用します。`create table` または `alter table` で作成したデフォルトを削除するには、`alter table` を使用してデフォルトを `NULL` に置き換えます。
- ユーザ定義データ型の既存のカラムは、`futureonly` を指定しない場合にかぎり、新しいデフォルト値を継承します。ユーザ定義データ型の新しいカラムは、常にデフォルトを継承します。デフォルトをユーザ定義データ型にバインドすると、そのデータ型のカラムにバインドされたデフォルトが無効になります。カラムのバインドを復元するには、バインドを解除してから再びカラムのデフォルトをバインドします。
- デフォルトを使用する文を、その `sp_bindefault` 文と同じバッチ内に入れることはできません。

パーミッション

`sp_bindefault` を実行するには、テーブル所有者またはユーザ・データ型の所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
6	bind	sp_bindefault	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – デフォルトの名前 • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 [コマンド create default, create table, drop default](#)
[システム・プロシージャ sp_unbindefault](#)

sp_bindexclass

説明	クライアント・アプリケーション、ログイン、ストアド・プロシージャ、またはデフォルト実行クラスに実行クラスを関連付けます。
構文	<code>sp_bindexclass "object_name", "object_type", "scope", "classname"</code>
パラメータ	<p>object_name 実行クラス <code>classname</code> に関連付けるクライアント・アプリケーション、ログイン、またはストアド・プロシージャの名前です。 <code>object_type</code> が <code>df</code> の場合、NULL である必要があります。</p> <p>object_type <code>object_name</code> のタイプを指定します。次の値を使用します。</p> <ul style="list-style-type: none">• アプリケーションの場合、<code>ap</code>• ユーザ定義のデフォルトの実行クラスの場合、<code>df</code>• ログインの場合、<code>lg</code>• ストアド・プロシージャの場合、<code>pr</code>• サービス・タスクの場合 (スレッド・モードのみで有効)、<code>sv</code> <p>scope クライアント・アプリケーションまたはログインの名前です。<code>ap</code> オブジェクト、<code>df</code> オブジェクト、<code>lg</code> オブジェクト、または <code>sv</code> オブジェクトには NULL を指定できます。オブジェクトの場合、<code>scope</code> がストアド・プロシージャ所有者の名前 (ユーザ名) です。 <code>object_name</code> に指定した名前のオブジェクトがアプリケーションまたはログインと対話するときには、設定した <code>scope</code> に対して、<code>classname</code> 属性が適用されます。</p> <p>classname <code>object_name</code> に関連付けるクラスの型を指定します。値は次のとおりです。</p> <ul style="list-style-type: none">• <code>EC1</code>、<code>EC2</code>、または <code>EC3</code>• ユーザ定義実行クラスの名前• <code>ANYENGINE</code>
例	<p>例 1 次の文では、(<code>scope</code> の値が NULL であるため) <code>isql</code> を呼び出すログインまたはアプリケーション・プロセスに対して、<code>EC3</code> 属性によって Transact-SQL アプリケーションを実行することを指定しています。ただし、それらのログインまたはアプリケーションがこれより上位の実行クラスにバインドされている場合を除きます。</p>

```
sp_bindexeclass 'isql', 'ap', NULL, 'EC3'
```

例 2 次の文では、システム管理者の役割を持つログインが Transact-SQL アプリケーションを実行するとき、そのログイン・プロセスを EC1 属性で実行することを指定しています。最初の例の文がすでに実行されている場合、isql を呼び出すその他のログインまたはアプリケーションはすべて EC3 属性で実行されます。

```
sp_bindexeclass 'sa', 'lg', 'isql', 'EC1'
```

例 3 次の文では、ユーザ kundu が所有するストアード・プロシージャ my_proc に EC3 属性を割り当てます。

```
sp_bindexeclass 'my_proc', 'PR', 'kundu', 'EC3'
```

例 4 次の文では、デフォルトの実行属性で実行しているすべてのタスクに CLASS1 属性を割り当てます。

```
sp_bindexeclass NULL, 'DF', NULL, 'CLASS1'
```

例 5 ライセンスのハートビート・オペレーションをコア実行タスクにバインドします。

```
sp_bindexeclass "License Heartbeat", sv, NULL, core
```

使用法

- 実行クラスをデフォルトの実行クラスにバインドする場合は、デフォルトの実行属性で実行しているすべてのタスクが新しいクラスの属性で実行されます。
- サービス・タスクは、ユーザ・タスクを管理するために作成された既存の実行クラスにバインドできます。つまり、サービス・タスクとユーザ・タスクは、同じ実行クラスに共存することができます。
- monServiceTask モニタリング・テーブルには、名前と現在のバインドも含め、すべてのサービス・タスクが含まれます。
- sp_bindexeclass は、クライアント・アプリケーション、ログイン、またはストアード・プロシージャに実行クラスを関連付けます。また、実行クラスをデフォルトの実行クラスに関連付けます。実行クラスの作成には [sp_addexeclass](#) を使用します。
- scope が NULL の場合、object_name にスコープはありません。classname に指定した実行属性は、このオブジェクトのすべての対話に適用されます。たとえば、object_name がアプリケーション名の場合、属性はそのアプリケーションを呼び出すログイン・プロセスに適用されます。object_name がログイン名の場合、属性はそのログイン・プロセスで呼び出されるアプリケーションの特定のログイン・プロセスに適用されます。

- ストアド・プロシージャを既存のクラスにバインドするときは、**scope** パラメータに、ストアド・プロシージャ所有者の名前(ユーザ名)を使う必要があります。同じデータベースで複数の呼び出しが実行されても、これによってストアド・プロシージャの名前を限定できます。
- 優先度やスコープの設定規則に応じて、**object_name** に指定した名前を持つオブジェクトに対して、バインドされる実行クラスが有効な場合と無効な場合があります。その他のバインド仕様、優先度、スコープ設定規則に応じて、オブジェクトは自動的に別の実行クラスにバインドされます。他に適用できるバインドがない場合、オブジェクトはデフォルトの実行クラスにバインドされます。ユーザ定義のデフォルトの実行クラスが指定されていない場合、オブジェクトはシステム定義のデフォルトの実行クラス **EC2** にバインドされます。
- **sp_bindexclass** では、**rep agent** をアプリケーションとして指定することで、エラーを発生させずに **RepAgent** スレッドを実行クラスにバインドすることは可能です。しかし、**Adaptive Server** の制約によって、優先度属性は中に設定され、このバインドは有効になりません。
- オンライン・エンジンのないエンジン・グループにアクティブ・プロセスをバインドしようとしても、バインドは実行されません。
- **Adaptive Server** は、**sysattributes** テーブルに、バインド用のデータを保管したローとオブジェクト ID とユーザ ID が入っているローを作成します
- ストアド・プロシージャにバインドするには、事前にそのストアド・プロシージャが存在していなければなりません。
- ストアド・プロシージャのバインドは、ストアド・プロシージャが格納されているデータベース内で実行してください。つまり、システム・プロシージャをバインドするときは、**sybsystemprocs** データベースから **sp_bindexclass** を実行します。
- クラスをストアド・プロシージャにバインドするときは、実行クラスの優先度属性だけを使用します。
- ストアド・プロシージャを実行クラスにバインドするときは、ストアド・プロシージャの所有者名を **scope** パラメータとして指定します。これによって、データベース内に同じ名前のストアド・プロシージャが複数あっても(ただし所有者が異なる場合)、ストアド・プロシージャをユニークに識別できます。

パーミッション sp_bindexclass のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage any execution class 権限を持つユーザであることが必要。 ECO の場合、manage any execution class と sybase_ts_role を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。 ECO の場合、sa_role と sybase_ts_role を持つユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addexclass](#), [sp_showexclass](#), [sp_unbindexclass](#)
ユーティリティ isql

sp_bindmsg

説明	ユーザ・メッセージを参照整合性制約または検査制約にバインドします。
構文	<code>sp_bindmsg constrname, msgid</code>
パラメータ	<p><i>constrname</i> メッセージのバインド先の整合性制約の名前です。制約を作成してそれに名前を付けるには、<code>create table</code> コマンドの <code>constraint</code> 句か、<code>alter table</code> コマンドの <code>add constraint</code> 句を使用します。</p> <p><i>msgid</i> 整合性制約にバインドするユーザ・メッセージの番号です。メッセージは、<code>sp_bindmsg</code> を呼び出す前に、ローカル・データベース内の <code>sysusermessages</code> テーブルに存在していなければなりません。</p>
例	<pre>sp_bindmsg positive_balance, 20100</pre> <p>ユーザ・メッセージ番号 20100 を <code>positive_balance</code> 制約にバインドします。</p>
使用法	<ul style="list-style-type: none">• <code>sp_bindmsg</code> は、<code>sysconstraints</code> テーブルの制約ローにメッセージ番号を追加することによって、整合性制約にユーザ・メッセージをバインドします。• バインドできるメッセージは 1 つだけです。制約に対するメッセージを変更するには、新しいメッセージをバインドしてください。<code>sysconstraints</code> テーブル内の古いメッセージ番号が、新しいメッセージ番号に置き換えられます。• メッセージは一意性制約にはバインドできません。これは、<code>sysconstraints</code> に、一意性制約に対応する制約ローがないためです（一意性制約はユニーク・インデックスです）。• <code>sysusermessages</code> テーブルにユーザ・メッセージを挿入するには、<code>sp_addmessage</code> プロシージャを使用します。• <code>sp_getmessage</code> プロシージャでは、<code>sysusermessages</code> テーブルからメッセージ・テキストが取り出されます。• <code>sp_help tablename</code> では、<code>tablename</code> で宣言されたすべての制約名が表示されます。
パーミッション	<code>sp_bindmsg</code> を実行するには、制約の所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
7	bind	sp_bindmsg	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – メッセージ ID • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

コマンド [alter table](#), [create table](#)

システム・プロシージャ [sp_addmessage](#), [sp_getmessage](#), [sp_unbindmsg](#)

sp_bindrule

説明	カラムまたはユーザ定義データ型にルールをバインドします。
構文	<code>sp_bindrule rulename, objname [, futureonly]</code>
パラメータ	<p>rulename ルールの名前です。ルールは <code>create rule</code> 文で作成して、<code>sp_bindrule</code> で特定のカラムまたはユーザ定義データ型にバインドします。</p> <p>objname ルールのバインド先のテーブルとカラム、またはユーザ定義データ型の名前です。<code>objname</code> に指定した値が “<code>table.column</code>” のフォーマットになっていない場合は、ユーザ定義データ型とみなされます。オブジェクト名が空白や句読記を含む場合、または予約語である場合は、引用符で囲ってください。</p> <p>futureonly ユーザ定義データ型の既存のカラムが新しいルールを継承しないようにします。ルールをユーザ定義データ型にバインドする場合、このパラメータは省略可能です。ルールをカラムにバインドするときには、このパラメータは意味を持ちません。</p>
例	<p>例 1 <code>create rule</code> で <code>today</code> という名前のルールが現在のデータベースに定義されている場合、次のコマンドでは、このルールが <code>employees</code> テーブルの <code>startdate</code> カラムにバインドされます。<code>employees</code> にローが追加されると、<code>startdate</code> カラムのデータはルール <code>today</code> と照合されます。</p> <pre>sp_bindrule today, "employees.startdate"</pre> <p>例 2 <code>rule_ssn</code> という名前のルールと、<code>ssn</code> という名前のユーザ定義データ型がある場合に、次のコマンドでは、<code>rule_ssn</code> が <code>ssn</code> にバインドされます。<code>create table</code> 文では、<code>ssn</code> データ型のカラムがルール <code>rule_ssn</code> を継承します。<code>ssn</code> データ型の既存のカラムもルール <code>rule_ssn</code> を継承します。ただし、<code>ssn</code> のルールが変更されている場合は、変更されたルールがそのまま維持されます。</p> <pre>sp_bindrule rule_ssn, ssn</pre> <p>例 3 ルール <code>rule_ssn</code> をユーザ定義データ型 <code>ssn</code> にバインドします。ただし、データ型 <code>ssn</code> の既存のカラムには作用しません。<code>futureonly</code> を指定すると、データ型 <code>ssn</code> の既存のカラムはルールを継承しません。</p> <pre>sp_bindrule rule_ssn, ssn, futureonly</pre>

使用法

- ルールは **create rule** 文で作成します。その後、**sp_bindrule** を実行して、そのルールを現在のデータベースのカラムまたはユーザ定義データ型にバインドします。
- ルールは、**sp_bindrule** を実行したときではなく、**insert** を試行したときに適用されます。真数値型や概数値型のカラムに文字ルールをバインドすることは可能ですが、このような **insert** は無効です。
- **sp_bindrule** を使用して、**create table** 文でカラムの検査制約をバインドすることはできません。
- Adaptive Server が提供するデータ型と **text** 型、または **image** 型のカラムには、ルールをバインドできません。
- システム・テーブルには、ルールをバインドできません。
- 計算カラムにルールをバインドすることはできません。
- カラムにルールをバインドする場合は、**objname** 引数を “**table.column**” のフォーマットで指定します。これ以外のフォーマットは、ユーザ定義データ型の名前とみなされます。
- ルールを使用する文を、**sp_bindrule** 文と同じバッチに入れることはできません。
- 既存のルールのバインドを解除しなくても、カラムまたはユーザ定義データ型に新しいルールをバインドできます。カラムにバインドされたルールは、データ型にバインドされたルールより優先されます。カラムにルールをバインドすると、そのカラムのユーザ定義データ型にバインドされたルールに代わってそのルールが使用されますが、データ型にルールをバインドしても、そのユーザ定義データ型のカラムにバインドされているルールは置き換わりません。
- ユーザ定義データ型の既存のカラムは、そのカラムのルールが以前に変更されていないかぎり、またはオプションの3番目のパラメータ値が **futureonly** でないかぎり、新しいルールを継承します。ユーザ定義データ型の新しいカラムは、常にルールを継承します。

パーミッション

sp_bindmsg を実行するには、テーブル所有者またはユーザ・データ型の所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
8	bind	sp_bindrule	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – ルールの名前 • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [create rule](#), [drop rule](#)

システム・プロシージャ [sp_unbindrule](#)

sp_cacheconfig

説明	データ・キャッシュの作成、設定、再設定、削除を行い、データ・キャッシュについての情報を提供します。
構文	<pre>sp_cacheconfig [cachename [, "cache_size[P K M G]" [, logonly mixed inmemory_storage][, strict relaxed]] [, "cache_partition=[1 2 4 8 16 32 64]" [, instance instance_name]</pre>
パラメータ	<p><i>cachename</i> 作成または設定するデータ・キャッシュの名前です。キャッシュ名はユニークな名前にしてください。最大 30 文字まで指定できます。有効な Adaptive Server 識別子である必要はありません。スペースや他の特殊文字も指定できます。</p> <p><i>cache_size</i> 作成するデータ・キャッシュのサイズです。すでにそのキャッシュが存在する場合は、データ・キャッシュの新しいサイズを指定します。最小キャッシュ・サイズは、サーバの論理ページ・サイズの 256 倍です。サイズをページ単位で指定する場合は P を使用します。バイト数で指定する場合は、K (キロバイト)、M (メガバイト)、G (ギガバイト) を使用します。デフォルトは K です。メガバイトとギガバイトによる指定では、浮動小数点数を指定できます。キャッシュ・サイズは論理ページ・サイズの倍数で指定します。</p> <p>logonly mixed inmemory_storage キャッシュのタイプを指定します。inmemory_storage は、メモリ内データベースおよびリラックス持続性データベースのキャッシュを作成していることを示します。</p> <p>strict relaxed キャッシュの置換方式を指定します。</p> <p>cache_partition キャッシュ内に作成するパーティションの数を指定します。</p> <p>instance_name (クラスタ環境の場合) 調整するキャッシュのインスタンスの名前です。</p>
例	<p>例 1 10MB の領域を持つデータ・キャッシュ pub_cache を作成します。領域はすべてデフォルトの論理ページ・サイズのメモリ・プールに配置します。</p> <pre>sp_cacheconfig pub_cache, "10M"</pre>

例 2 `pub_cache` の現在の設定と、このキャッシュ内のメモリ・プールについてレポートします。

```
sp_cacheconfig pub_cache
```

例 3 Adaptive Server の次回の起動時に、`pub_cache` を削除します。

```
sp_cacheconfig pub_cache, "0"
```

例 4 `pub_log_cache` を作成し、キャッシュ・タイプを `logonly` に設定します。この 2 つの処理を一度に実行します。

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

例 5 最初のコマンドで、キャッシュ `pub_log_cache` を `mixed` のデフォルト・タイプで作成します。2 番目のコマンドで、このキャッシュのステータスを `logonly` に変更します。その結果、このキャッシュの設定は例 4 のキャッシュと同じになります。

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

例 6 キャッシュを作成し、このキャッシュのサイズ、タイプ、置換方式、キャッシュ・パーティションの数を設定します。

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

例 7 `pubs3_imdb` という名前メモリ内記憶域キャッシュを作成します。

```
sp_cacheconfig pubs_imdb, '500M', inmemory_storage
```

例 8 (クラスタ環境の場合) インスタンス `blade1` のキャッシュを表示します。

```
sp_cacheconfig 'instance blade1'
```

例 9 (クラスタ環境の場合) `blade1` の Sales Cache サイズを 100 メガバイトに設定します。

```
sp_cacheconfig 'Sales Cache', '100M', 'instance blade1'
```

例 10 (クラスタ環境の場合) `blade1` の Sales Cache サイズを 0 メガバイトに設定し、キャッシュを効率的に削除します。

```
sp_cacheconfig 'Sales Cache', '0M', 'instance blade1'
```

使用法

- 最小キャッシュ・サイズは論理ページ・サイズの 256 倍です。たとえば、4K のサーバの場合、最小キャッシュ・サイズは 1024K になります。

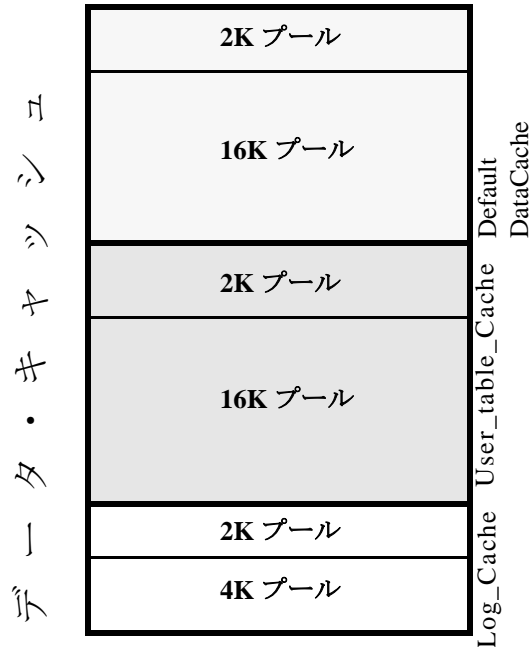
- 新しいキャッシュの作成または既存のキャッシュへのメモリの追加時に、要求したメモリのすべてを割り付けられないときは、利用可能なすべてのメモリが割り付けられます。ただし、この追加メモリの割り付けは **Adaptive Server** の次のリポート時に行われます。
- キャッシュ (デフォルトのキャッシュを含む) にオブジェクトがバインドされている場合、オブジェクトのバインドを解除するまでそのキャッシュの削除は実行できません。
- (クラスタ環境の場合) `instance_name` を指定していないと、クラスタのキャッシュが表示されます。
- `sp_cacheconfig` を使用して実行するアクションには、動的 (**Adaptive Server** のリポートは不要) なものと、静的 (リポートが必要) なものがあります。表 1-5 に、動的なアクションと静的なアクションを示します。

表 1-5: `sp_cacheconfig` の動的アクションと静的アクション

<code>sp_cacheconfig</code> の動的アクション	<code>sp_cacheconfig</code> の静的アクション
新しいキャッシュの追加	キャッシュ・パーティション数の変更
既存のキャッシュへのメモリの追加	キャッシュ・サイズの削減
キャッシュの削除	置換方式の変更
キャッシュ・タイプの変更	

- データ・キャッシュの初回作成時は、次のようになります。
 - すべての領域は論理ページ・サイズのメモリ・プールに割り付けられます。
 - デフォルトのタイプは `mixed` です。
- 図 1-1 (109 ページ) は、ユーザ定義のデータ・キャッシュが 2 つ設定された 2K サーバのデータ・キャッシュを示したものです。このデータ・キャッシュには、次のプールが存在します。
 - 2K プールと 16K プールを持つデフォルト・データ・キャッシュ
 - 2K プールと 16K プールを持つユーザ・キャッシュ
 - 2K プールと 4K プールを持つログ・キャッシュ

図 1-1: デフォルト・キャッシュとユーザ定義キャッシュを持つデータ・キャッシュ



- デフォルトのデータ・キャッシュのタイプは常に `default` です。デフォルト以外のキャッシュのタイプを `default` に設定することはできません。
- Adaptive Server のハウスキーピング・タスクは、`logonly` タイプのキャッシュまたはリラックス LRU 置換方式のキャッシュに対するバッファ・ウォッシュを行いません。
- `disk init` コマンド、一部の `dbcc` コマンド、`drop table` コマンドは、2K の I/O だけを実行します。`dbcc checkdb` コマンドと `dbcc checktable` コマンドは、テーブルに対しては大容量 I/O を実行できますが、インデックスに対しては 2K の I/O を実行します。表 1-6 に、データベースまたはオブジェクトのバインドに対応するキャッシュの使用状況を示します。

表 1-6: Transact-SQL コマンドにおけるキャッシュの使用状況

コマンド	データベースをバインド	テーブルまたはインデックスをバインド	データベースもオブジェクトもバインドしない
<code>create index</code>	バインドされたキャッシュ	なし	デフォルト・データ・キャッシュ
<code>disk init</code>	なし	なし	デフォルト・データ・キャッシュ
<code>dbcc checkdb</code>	バインドされたキャッシュ	なし	デフォルト・データ・キャッシュ
<code>dbcc checktable</code> 、 <code>indexalloc</code> 、 <code>tablealloc</code>	バインドされたキャッシュ	バインドされたキャッシュ	デフォルト・データ・キャッシュ
<code>drop table</code>	バインドされたキャッシュ	バインドされたキャッシュ	デフォルト・データ・キャッシュ

- リカバリでは、デフォルト・データ・キャッシュの論理ページ・サイズのプールだけが使用されます。ロールバックまたはロールフォワードするトランザクションのすべてのページは、プールに読み込まれて変更されます。使用するデフォルトの論理ページ・サイズを、これらのトランザクションに十分な大きさにしておきます。
- パラメータなしで `sp_cacheconfig` を使用すると、サーバ上のすべてのキャッシュに関する情報がレポートされます。キャッシュ名だけを指定すると、そのキャッシュについての情報だけがレポートされます。キャッシュ名の一部を指定すると、ワイルドカード指定 “%fragment%” と一致するすべての名前が提供されます。

すべてのレポートは、キャッシュ全体についての情報をレポートする情報ブロックと、それぞれのキャッシュについてのデータ・ブロックで構成されています。このデータ・ブロックは、キャッシュ内のプールについての情報を提供します。

2K を使用するサーバからの以下の出力例は、次の設定に基づいています。

- デフォルト・データ・キャッシュには 2 つのパーティションがあります。2K プールと 16K プールを持つユーザ・キャッシュデフォルト・データ・キャッシュには 2 つのパーティションがあります。
- 2K と 16K の 2 つのプールを持つ `pubs_cache`。

- 2K のプールと 4K のプールを持つ pubs_log。データ・キャッシュのタイプは logonly に、キャッシュ置換方式は relaxed に設定する。

```

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 M
-----
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 1-7 に、出力結果の各カラムの意味を示します。

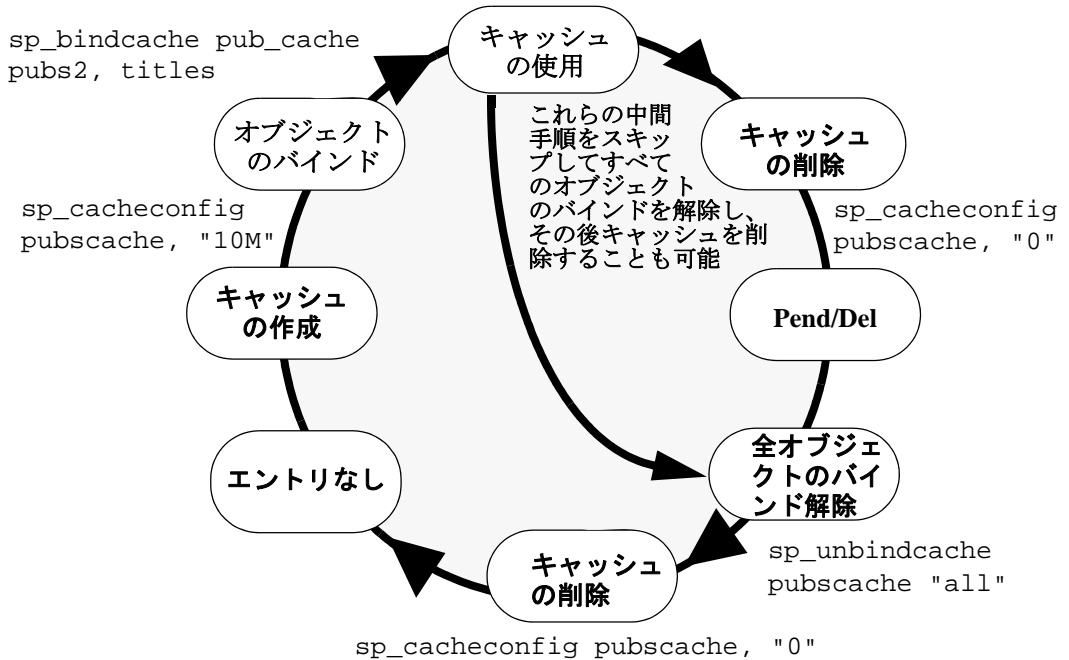
表 1-7: sp_cacheconfig の出力

カラム	意味
Cache Name	キャッシュの名前。

カラム	意味
Status	次のいずれか1つ。 <ul style="list-style-type: none"> • “Active” • “Pend/Act” • “Pend/Del” これらの詳細については下記参照。
Type	ユーザ定義のキャッシュには “Mixed” または “Log Only”、デフォルト・データ・キャッシュには “Default”。
I/O Size	メモリ・プールに対する I/O のサイズ。キャッシュの設定を示す行については、このカラムはブランク。
Wash Size	プールに対するウォッシュ・エリアのサイズ。ページがキャッシュのウォッシュ・エリアに入ると、それらはディスクに書き込まれる。キャッシュの設定を示す行については、このカラムはブランク。
Config Value または Config Size	キャッシュまたはプールのサイズ。値が 0 の場合、このサイズは明示的には設定されておらず、デフォルト値が使用される。
Run Value または Run Size	Adaptive Server で現在使用中のキャッシュまたはプールのサイズ。
Config/Run Replacement	次の再起動後、キャッシュに使うキャッシュ方式 (ストリクトまたはリラックス) と現在の置換方式。これらは、最後のレポート以降に方式が変更された場合にのみ異なる。
Config/Run Partition	次回キャッシュに使用されるキャッシュ・パーティションの数と現在のパーティション数。前回のレポート以降に、sp_cacheconfig によってパーティションの数が増えたり減ったりした場合、これらの数は異なる値になる。
APF Percent	非同期プリフェッチによってフェッチされたが使われなかったバッファを保持できるプール内のバッファの割合。
Total	すべてのキャッシュをレポートする場合は、データ・キャッシュの合計サイズ。キャッシュ名を指定した場合は、その特定のキャッシュの現在のサイズ。

ステータス "Pend" は保留 (pending) の短縮形です。このステータスは、常に、アクティブを意味する “Act” または削除を意味する “Del” のどちらかとの組み合わせで発生します。これは設定のアクションが実行されたことを示します。ただし、その変更を有効にするには、サーバを再起動する必要があります。

図 1-2: キャッシュ・ステータスでの再起動と sp_cacheconfig の影響



- キャッシュとプールの設定は、設定ファイルの編集によって行うこともできます。詳細については、『システム管理ガイド』を参照してください。

データ・キャッシュ・メモリ

- Adaptive Server の初回インストール時、データ・キャッシュ・メモリはすべて `default data cache` という名前のキャッシュの、論理ページ・サイズのプールに割り当てられます。このデフォルトのデータ・キャッシュは、`sp_bindcache` によってデータ・キャッシュに明示的にバインドされていないオブジェクト、またはキャッシュにバインドされていないデータベースを持つオブジェクトが使用します。

- データ・キャッシュの作成時には、**max memory** 値に基づいてメモリ割り付けが検証されます。キャッシュ用のメモリは、**total logical memory** 設定パラメータによって **Adaptive Server** に割り付けられたメモリの中から割り付けられます。キャッシュに使用できる領域の量を増やすには **total logical memory** 値を増やすか、メモリを使用する他の設定値を低くしてください。**total logical memory** 値と要求した追加メモリ量の合計が **max memory** 値を超えると、**Adaptive Server** からエラーが出力され、変更処理は実行されません。

デフォルトのキャッシュは、別のキャッシュにバインドされていないすべてのオブジェクト(システム・テーブルも含む)に使用されます。また、リカバリ時には、このデフォルトのキャッシュのみが使用されます。詳細については、『システム管理ガイド』を参照してください。

- データ・キャッシュには、キャッシュを管理する構造用に若干のオーバーヘッドが必要です。すべてのキャッシュ・オーバーヘッドは、空きメモリから引かれます。特定のサイズのキャッシュに必要なオーバーヘッドの量を調べるには、キャッシュのサイズを指定した **sp_helpcache** を実行します。

```
sp_helpcache "200M"
```

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

これはあくまでオーバーヘッドの予想値です。実際のオーバーヘッドは、実行時の問題が原因でこれよりも増えることがあります。

インメモリ・データベースまたはリラックス持続性データベースのキャッシュの作成

- キャッシュ名の最大長は 127 バイトです。
- メモリ内記憶域キャッシュの最小サイズは、256 論理ページです(2K の論理ページを使用したサーバの場合は 512K になります)。
- 次の処理はできません。
 - メモリ内記憶域の **strict** または **relaxed** 置換方式を含めます。デフォルトで、**sp_cacheconfig** はメモリ内記憶域の **none** 置換方式を使用します。
 - メモリ内記憶域キャッシュに対する大容量 I/O プールを作成します(インメモリ・データベースは I/O を実行しません)。**sp_poolconfig** を使用してメモリ内記憶域のバッファ・プールを作成する場合、**Adaptive Server** はエラーを発生します。

- キャッシュ・タイプを `mixed` から `logonly` (またはその逆) に変更します。

既存のキャッシュの変更

- 既存のキャッシュのサイズを変更するには、キャッシュの名前と新しいサイズを指定します。
- 既存のキャッシュのサイズを増やすと、追加された領域はすべて最小プール内に配置されます。
- 既存のキャッシュのサイズを縮小する場合、そのすべての領域を論理ページ・サイズのプール内で使用できるようにします。場合によっては、`sp_poolconfig` を使用して別のプールからこのプールへ領域を移動する必要があります。
- キャッシュにバインドされたデータベースまたはログをとらないオブジェクトがある場合、タイプを `logonly` に変更することはできません。

キャッシュ・パーティションの使用

- キャッシュ・パーティションを使用すると、独立したキャッシュを作成してデータベース・オブジェクトをこれらのキャッシュにバインドしなくても、キャッシュのスピンロック競合を減らすことができます。キャッシュのスピンロック競合を監視する方法の詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。
- `global cache partition number` 設定パラメータを使用すると、すべてのキャッシュにデフォルト数のキャッシュ・パーティションを設定できます。詳細については、『システム管理ガイド』を参照してください。

キャッシュの削除

- データ・キャッシュを削除するには、例 3 に示すようにキャッシュ・サイズを 0 に変更します。キャッシュ・サイズを 0 に設定すると、そのキャッシュは削除対象としてマーク付けされます。キャッシュはアクティブな状態を続け、そのキャッシュにバインドされたすべてのオブジェクトは引き続きそのキャッシュを使用します。

デフォルト・データ・キャッシュは削除できません。

- オブジェクトがバインドされているデータ・キャッシュを削除しようとしても、そのキャッシュはメモリ内に残ります。この場合、`Adaptive Server` から次のメッセージが発行されます。

```
Cache (nmc3) not deleted dynamically. Objects are bound to the cache. Use
```

sp_unbindcache_all to unbind all objects bound to the cache.

設定ファイル内および sysconfigures 内のこのキャッシュに対応するエントリが削除されます。キャッシュ自体は Adaptive Server の次の再起動時に削除されます。

- トランザクション内で sp_cacheconfig を実行することはできません。

パーミッション

sp_cacheconfig のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage data cache 権限を持つユーザであることが必要。 どのユーザも sp_cacheconfig を実行してキャッシュ設定を見ることができ る。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。 どのユーザも sp_cacheconfig を実行してキャッシュ設定を見ることができ る。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの 実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_bindcache](#)、[sp_helpcache](#)、[sp_poolconfig](#)、[sp_unbindcache](#)、[sp_unbindcache_all](#)

sp_cachestrategy

説明 テーブル、インデックス、text オブジェクト、または image オブジェクトに対するプリフェッチ (大容量 I/O) と MRU キャッシュの置換方式を有効または無効にします。

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

パラメータ

dbname

対象のオブジェクトが格納されているデータベースの名前です。

ownername

テーブル所有者の名前です。“dbo” がテーブルを所有している場合は、所有者名を省略できます。

tablename

テーブルの名前を指定します。

indexname

テーブルのインデックスの名前です。

text only

text オブジェクトまたは image オブジェクトのキャッシュ方式を変更します。

table only

テーブルのキャッシュ方式を変更します。

prefetch | mru

設定変更の対象を prefetch または mru のどちらかで指定します。

on | off

"on" または "off" のように引用符で囲んで設定値を指定します。

例

例 1 titles テーブルのキャッシュ方式についての情報を表示します。

```
sp_cachestrategy pubs2, titles
object name          index name          large IO  MRU
-----
dbo.titles           titleidind          ON        ON
```

例 2 titleind インデックスのキャッシュ方式についての情報を表示します。

```
sp_cachestrategy pubs2, titles, titleind
```

例 3 titles テーブルの titleind インデックスのプリフェッチを無効にします。

```
sp_cachestrategy pubs2, titles, titleind, prefetch, "off"
```

例 4 authors テーブルの MRU 置換方式を再び有効にします。

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

例 5 blurbs テーブルのテキスト・ページのプリフェッチを再び有効にします。

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

使用法

- 大容量 I/O に対するメモリ・プールが、テーブルまたはインデックスで使用されるキャッシュに設定されている場合、オプティマイザは、一度に 8 データ・ページまでの大容量 I/O を実行して、データまたはインデックス・ページをプリフェッチできます。この prefetch 方式は、テーブルのデータ・ページまたはノンクラスタード・インデックスのリーフ・レベルのページに対して使用できます。デフォルトでは、すべてのテーブル、インデックス、text オブジェクトまたは image オブジェクトに対してプリフェッチが有効になっています。prefetch オプションを“off”に設定すると、指定したオブジェクトのプリフェッチが無効になります。
- オプティマイザでは、あらゆるサイズの I/O のテーブル・スキャンとインデックス・スキャンに対して、「MRU 置換方式」を使用して、キャッシュ内のバッファを使い捨てる (fetch and discard) ことができます。デフォルトでは、この方式はすべてのオブジェクトに有効です。mru を“off”に設定すると、この方式が無効になります。あるオブジェクトで mru をオフにすると、すべてのページがキャッシュ内の MRU/LRU チェーンに読み込まれます。読み込まれたページは、追加の I/O によってフラッシュされるまでキャッシュ内に残ります。キャッシュ方式の詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。
- キャッシュ方式を変更できるのは、現在のデータベースにあるオブジェクトだけです。
- 方式と設定を指定せずに sp_cachestrategy を使用すると、そのオブジェクトに対する現在の設定内容が、例 1 に示しているようにレポートされます。
- サーバ上にあるすべてのデータ・キャッシュのサイズ、ステータス、I/O サイズを調べるには、sp_cacheconfig を使用します。
- prefetch "on" の設定は、2K の I/O のみを許容するキャッシュに読み込まれたテーブルやインデックスには作用しません。mru 方式は、使用可能な I/O サイズに関係なく、すべてのキャッシュに使用できます。

設定を無効にする

- テーブルまたはインデックスに対してプリフェッチがオンになっている場合、`set prefetch "off"` によって、セッションのプリフェッチを無効にできます。オブジェクトに対してプリフェッチがオフになっている場合、その設定は無効にできません。
- `select`、`delete`、`update` の各コマンドに `prefetch`、`lru`、`mru` オプションを指定することによって、個々の文に使用する I/O サイズとキャッシュ方式の推奨設定を表示できます。テーブルまたはインデックスに対してプリフェッチまたは MRU 方式が有効になっている場合に、`prefetch` に論理ページ・サイズの I/O を指定し、`lru` 方式を指定することによって、個々のクエリで元の設定を無効にすることができます。たとえば、次のコマンドでは、LRU 方式、論理ページ・サイズの I/O、および `titles` テーブルのテーブル・スキャンが強制的に適用されます。

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

プリフェッチのサイズを要求したときに、オブジェクトのキャッシュが要求したサイズの I/O に設定されていない場合、最適マイザは使用可能で最適な I/O サイズを選択します。

- オブジェクトに対して `sp_cachestrategy` でプリフェッチが有効である場合、`select`、`update`、または `delete` コマンドに論理ページ・サイズの `prefetch` 指定を含めると、元の `set prefetch "on"` 文が無効になります。`select`、`update`、または `delete` コマンドでより大きな I/O サイズを指定しても、`set prefetch "off"` コマンドは無効化されません。

パーミッション

`sp_cachestrategy` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、オブジェクト所有者か <code>manage data cache</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、オブジェクト所有者か <code>sa_role</code> を持つユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [delete](#)、[select](#)、[set](#)、[update](#)

ストアド・プロシージャ [sp_cacheconfig](#)、[sp_poolconfig](#)

sp_changedbowner

説明	ユーザ・データベースの所有者を変更します。
構文	<code>sp_changedbowner loginame [, true]</code>
パラメータ	<p><i>loginame</i> 現在のデータベースの新しい所有者のログイン名です。</p> <p><code>true</code> エイリアスとそのパーミッションを新しいデータベース所有者に譲渡します。有効な値は “true” と “TRUE” です。</p>
例	<p>ユーザ “albert” を現在のデータベースの所有者にします。</p> <pre>sp_changedbowner albert</pre>
使用法	<ul style="list-style-type: none"> ユーザまたはエイリアスのどちらかとしてすでに認識されている所有者を、新しい所有者として指定することはできません (つまり、新しい所有者は <code>sysusers</code> と <code>sysalternates</code> のどちらにも含まれていないユーザでなければなりません)。 <i>loginame</i> だけを指定して <code>sp_changedbowner</code> を実行すると、そのデータベースの所有者が、<i>loginame</i> に指定したユーザに変更され、以前の “dbo” のエイリアスが削除されます。 <code>sp_changedbowner</code> の実行後は、新しい所有者がそのデータベースでデータベース所有者として認識されます。 <code>sp_changedbowner</code> によってシステム・データベースの所有権を譲渡することはできません。 新しい所有者は、すでに Adaptive Server 内にログイン名を持っている必要がありますが、データベース内でデータベース・ユーザ名またはエイリアスを持つことは禁じられています。新しい所有者にするユーザのデータベース・ユーザ名またはエイリアスが対象のデータベースにすでに存在する場合は、既存のユーザ名またはエイリアスを削除してから <code>sp_changedbowner</code> を実行します。 新しい所有者にパーミッションを付与するには、システム管理者がパーミッションをデータベース所有者に与える必要があります。これは、すでにユーザはデータベース内部で認識されていないためです。
パーミッション	<code>sp_changedbowner</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>own any database</code> 権限を持つユーザであることが必要。

細密なパーミッションが無効
無効

細密なパーミッションが無効の場合、`sa_role` が付与されたユーザであることが必要。

監査 sysaudits テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [create database](#)

システム・プロシージャ [sp_addlogin](#)、[sp_dropalias](#)、[sp_dropuser](#)、[sp_helpdb](#)

sp_changegroup

説明	ユーザのグループを変更します。
構文	<code>sp_changegroup grpname, username</code>
パラメータ	<p>grpname グループの名前です。グループは、事前に現在のデータベースに存在していなければなりません。キーワード “public” を <i>grpname</i> として指定する場合は、引用符で囲みます。</p> <p>username グループに追加するユーザの名前です。ユーザは、事前に現在のデータベースに存在していなければなりません。</p>
例	<p>例 1 ユーザ “albert” をグループ “fort_mudge” のメンバにします。以前に “albert” がどのグループに属していたかは無視されます。</p> <pre>sp_changegroup fort_mudge, albert</pre> <p>例 2 “albert” を新しいグループのメンバにしないで、元のグループから削除します (ユーザは全員、常にデフォルトのグループ “public” のメンバ)。</p> <pre>sp_changegroup "public", albert</pre>
使用法	<ul style="list-style-type: none">• <code>sp_changegroup</code> を実行すると、指定したユーザが指定したグループに追加されます。この場合、ユーザは、以前属していたグループから削除されて、<i>grpname</i> に指定したグループに追加されます。• 新しいデータベース・ユーザは、<code>sp_adduser</code> によってデータベースへのアクセス権が与えられると同時に、グループに追加されます。• 各グループは、権限の付与や取り消しに対して集合名として使用されます。ユーザは全員、常に “public” のメンバであり、それ以外に所属できるグループは 1 つだけです。• ユーザを、新しいグループのメンバにしないで、元のグループから削除するには、例 2 に示すように、<code>sp_changegroup</code> でそのユーザのグループを “public” に変更します。• あるグループから別のグループに変更されたユーザは、元のグループに属していたときに持っていたすべてのパーミッションを失いますが、新しいグループに与えられているパーミッションを取得します。

パーミッション `sp_changegroup` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any user</code> 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者、 <code>sa_role</code> を持つユーザー、または <code>sso_role</code> を持つユーザーであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド `grant`、`revoke`](#)
システム・プロシージャ [sp_addgroup](#)、[sp_adduser](#)、[sp_dropgroup](#)、[sp_helpgroup](#)

sp_checknames

- 説明** 現在のデータベースで、7ビットの ASCII 文字セット以外の文字を使用している名前がないかどうかを検査します。
- 構文** `sp_checknames [help | silent]`
- パラメータ**
- help**
スキャンしたシステム・テーブルについての情報を表示します。
- silent**
現在のデータベースをサイレント・モードでチェックし、次のいずれかを返します。
- 0 – 7ビット ASCII 文字以外の文字を使用した名前がない場合
 - 1 – 少なくとも1つの名前が7ビット ASCII 文字以外の文字が使用されている場合
- 例** **例 1** 7ビットの ASCII 文字セット以外の文字が、master データベースの名前に使われていないかを検査します。

```
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
```

例 2 スキャンしたシステム・テーブルについての情報を表示します。

```
1> sp_checknames help
2> go
```

```
sp_checknames is used to search for non 7-bit ASCII characters
several important columns of system tables. The following
columns are searched:
```

```
In "master":
  sysdatabases.name
  sysdevices.name
  syslogins.name
  syslogins.dbname
  syslogins.password
  sysremotelogins.remoteusername
  syssservers.srvname
  syssservers.srvnetname
```

```
In all databases:
  syscolumns.name
  sysindexes.name
  sysobjects.name
  syssegments.name
  systypes.name
  sysusers.name
```

```
(return status = 0)
1>
```

例 3 システム・テーブル名を出力せず、リターン・ステータスのみを表示します。

```
1> sp_checknames silent
2> go

(return status = 1)
```

使用法

- `sp_checknames` は、7 ビットの ASCII 文字セット以外の文字が使われていないかどうかを調べます。現在のデータベース内にあるすべてのオブジェクト、カラム、インデックス、ユーザ名、グループ名、および他の要素が対象となります。また、無効な名前を表示して、7 ビットの ASCII 文字セットと互換性を持たせるための手順を示します。
- SQL Server のリリース 4.0.x または 4.2.x からアップグレードした後、7 ビット ASCII 以外のデフォルトの文字セットを使用した後には、サーバ上の各データベースに対して `sp_checknames` を実行してください。
- `sp_checknames` で表示された手順に従って、ASCII 以外の文字を使用した名前をすべて訂正してください。

パーミッション

`sp_checknames` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [update](#)システム・プロシージャ [sp_password](#)、[sp_rename](#)、[sp_renamedb](#)

sp_checkreswords

説明 識別子として使われている Transact-SQL の予約語を検出して表示します。サーバ名、デバイス名、データベース名、セグメント名、ユーザ定義データ型、オブジェクト名、カラム名、ユーザ名、ログイン名、リモート・ログイン名を確認します。

構文 sp_checkreswords [*user_name_param*]

パラメータ *user_name_param*

現在のデータベース内のユーザ名です。*user_name_param* を指定すると、指定したユーザが所有するオブジェクトだけが検査されます。

例 **例 1** 次の例は、master データベースで sp_checkreswords を実行したときの結果を示します。

```
1> /* executed in the master database */
2> sp_checkreswords
```

```
Reserved Words Used as Database Object Names for Database master
```

```
Upgrade renames sysobjects.schema to sysobjects.schemact.
```

```
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

例 2 次の例は、ユーザ・データベース user_db で sp_checkreswords を実行したときの結果を示します。

```
1> /* executed in the user database, user_db */
```

```
2> sp_checkreswords
```

Reserved Words Used as Database Object Names for Database user_db

Upgrade renames sysobjects schema to sysobjects.schemacnt.

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.

使用法

- `sp_checkreswords` では、既存のオブジェクトのうち、名前に予約語が使用されているものがレポートされます。Transact-SQL では、区切り識別子を使用せずにコマンド構文の構成要素となるワードを識別子として使用することはできません。予約語は SQL 構文の一部です。これがコマンドの一部として入力された場合は、特別な意味を持ちます。たとえば、リリース 10.0 より前の SQL Server では、次のクエリを使用して、`work` という名前のテーブルからデータを選択できました。

```
select * from work
```

SQL Server リリース 10.0 で、`work` は `commit work` コマンドの構成要素として新しい予約語になっていましたが、リリース 10.0 以降で同じ `select` 文を発行すると、構文エラーが発生します。

`sp_checkreswords` プロシージャを使用すると、エラーを引き起こす識別子が検出されます。

- `sp_checkreswords` では、`set quoted_identifier` オプションで作成された、識別子として使用されている予約語も検出されます。
- Adaptive Server を新しいリリースにアップグレードするときには、アップグレードの前か直後に、`sp_checkreswords` を使用します。このプロシージャをインストールし実行させてからアップグレードを行う方法の詳細については、使用しているプラットフォームのインストール・ガイドを参照してください。

`master` データベースと各ユーザ・データベースで `sp_checkreswords` を実行します。また、`model` データベースや `sybsystemprocs` データベースにユーザやオブジェクトを追加した場合は、これらのデータベースでも同様に実行します。

- リターン・ステータスは、検出された項目数を示します。
- `sp_checkreswords` にユーザ名を指定すると、テーブル、インデックス、ビュー、プロシージャ、トリガ、ルール、デフォルト、ユーザ定義データ型など、そのユーザが所有できるすべてのオブジェクトがチェックされます。予約語に一致するすべての識別子がレポートされます。

- 現在のデータベースが **master** データベースでない場合に **sp_checkreswords** に特定のユーザ名を指定しないと、上記のオブジェクトのすべてが検査され、結果がユーザ名ごとに個別のセッションに分けてレポートされます。また、**sysusers** と **syssegments** に対しても、それぞれユーザ名、セグメント名に予約語が使用されていないかチェックされます。**model** と **sysystemprocs** については、オブジェクト、ユーザ、またはユーザ定義データ型をこれらのデータベースに追加した場合のみ検査する必要があります。
- 現在のデータベースが **master** でない場合に **sp_checkreswords** にユーザ名を指定しないと、上記のオブジェクトのほかに、**sysdatabases**、**syslogins**、**syscharsets**、**sys.servers**、**sysremotelogins**、**sysdevices**、および **syslanguages** に対しても、データベース名、ローカルまたはリモート・ログイン名、ローカルまたはリモート・サーバ名、文字セット名、言語名に予約語が使用されていないかチェックされます。

レポートされた予約語のインスタンスの処理

- 予約語が識別子として使用されていることが **sp_checkreswords** でレポートされた場合、次の2つのいずれかの処置を選択できます。
 - **sp_rename** または **sp_renamedb** を使用するか、システム・テーブルを更新して問題の識別子の名前を変更します。
 - 予約語がテーブル名、ビュー名、またはカラム名として使用されている場合は、**set quoted_identifier on** を使用します。ほとんどのアプリケーションでストアド・プロシージャを使用する場合は、これらのプロシージャを一度削除してから **set quoted_identifier on** を指定して再作成することができます。この場合、すべての識別子を引用符で囲みます。これにより、ユーザ全員が、セッションに **set quoted_identifier on** を設定せずにプロシージャを実行できるようになります。**set quoted_identifier on** を設定した後、テーブルまたはカラムに代替名を割り当てるビューを作成し、代わりにそのビューを参照するようアプリケーションを変更できます。

新しい予約語 “key”、“level”、“work” に代替名を割り当てる例を次に示します。

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

set コマンドの構文は次のとおりです。

```
set quoted_identifier on
```

- 予約語として検出された識別子を変更しない場合や区切り識別子を使用しない場合、クエリでその予約語を識別子として使用すると、エラー (通常、構文エラー) がレポートされます。次に例を示します。

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

注意 区切り識別子オプションは SQL92 のオプションですが、他の Adaptive Server 機能をサポートするクライアント製品の多くでは、サポートされていないことがあります。たとえば、テーブル名に予約語を使用しているテーブルでは、bcp を使用できません。

区切り識別子オプションを選択する前に、Adaptive Server にアクセスするすべてのツールを使用して、いろいろなオブジェクトでテストを行ってください。set quoted_identifier on を使用し、テーブル名とカラム名に予約語を使用したテーブルを作成します。クライアント製品が SQL コードを生成する場合は、識別子を二重引用符 (それらが予約語の場合) で囲み、文字定数は一重引用符で囲みます。

- 名前が変更されたオブジェクトに従属するプロシージャ、トリガ、ビューは、名前の変更後も動作しますが、クエリ・プランが再コンパイルされると動作しなくなります。再コンパイルは、さまざまな理由で発生しますが、ユーザには通知されません。機能が突然停止しないようにするためには、オブジェクトの名前を変更した後、プロシージャ、トリガ、ビュー内の各オブジェクトの名前も変更してください。
- オブジェクト名を変更したり、区切り識別子を使用したりした場合は、とにかく、予約語を含んでいるすべてのストアド・プロシージャ、ビュー、トリガ、アプリケーションを変更してください。オブジェクト名を変更した場合は、識別子を変更してください。区切り識別子を使用した場合は、set quoted_identifier オプションを指定し、引用符を追加してください。
- プロシージャ、トリガ、ビュー、ルール、デフォルトのテキストがオペレーティング・システム・ファイルに保存されていない場合は、defncopy を使用して、これらの定義をサーバからファイルにコピーできます。詳細については、『ユーティリティ・ガイド』の defncopy の項を参照してください。

識別子の変更

- `sp_checkreswords` でレポートされた項目の名前を変更する場合は、その予約語を使用しているオブジェクトを参照するすべてのプロシージャ、トリガ、ビュー、アプリケーションで、問題となる名前を変更します。
- データベースをダンプしてから識別子名を変更してください。識別子名を変更した後は、`dbcc` を実行して問題がないことを確認し、データベースをもう一度ダンプします。
- アクティブな運用データベースで識別子を変更する場合は、次の作業を行ってください。
 - できるだけユーザの作業を妨害しないように、システムが混んでいないときに変更を行ってください。
 - 特定の識別子を使用するすべての **Open Client DB-Library™** プログラム、ウィンドウ・アプリケーション、ストアード・プロシージャ、トリガ、スクリプトを探して、慎重に準備してください。こうすれば、短時間でソース・コードに必要な編集を行い、識別子を変更して、プロシージャとコードを置き換えることができます。
- `sp_depends` プロシージャを使用すると、テーブルおよびビューの名前を使用しているプロシージャ、ビュー、トリガを簡単に検出できます。

`sp_rename` を使用した識別子の変更

- `sp_rename` システム・プロシージャでは、テーブル、インデックス、ビュー、プロシージャ、トリガ、ルール、デフォルト、ユーザ定義データ型、およびカラムの名前が変更されます。データベース名を変更するには、`sp_renamedb` を使用します。
- 表 1-8 に、`sp_rename` で変更できる識別子のタイプを示し、タイプごとにサーバやアプリケーション・プログラムに適用する必要があるその他の変更について示します。

表 1-8: `sp_rename` と識別子の変更

識別子	注意事項
テーブル名	<ul style="list-style-type: none"> • テーブルを参照するすべてのプロシージャ、トリガ、ビューを削除して、新しい名前で再作成する。テーブルに付属するオブジェクトを検出するには、<code>sp_depends</code> を使用する。 • テーブルを参照するすべてのアプリケーションまたは SQL ソース・スクリプトを変更して、新しい名前を使用する。 • テーブル名を使用して、テーブル・レベルの確認を実行する <code>dbcc</code> スクリプトを変更する。

識別子	注意事項
インデックス名	<ul style="list-style-type: none"> インデックスを作成、または削除するストアド・プロシージャを削除して、新しい名前で作成する。 インデックスを作成、または削除するすべてのアプリケーションや SQL ソース・スクリプトを変更する。 インデックス・レベルの確認をインデックス名によって実行する <code>dbcc</code> スクリプトを変更する。
ビュー名	<ul style="list-style-type: none"> ビューを参照するすべてのプロシージャ、トリガ、ビューを削除して、新しい名前で作成する。ビューに従属するオブジェクトを検出するには、<code>sp_depends</code> を使用する。 ビューを参照するすべてのアプリケーションまたは SQL ソース・スクリプトを変更して、新しい名前を使用する。
プロシージャ名	<ul style="list-style-type: none"> プロシージャを参照するすべてのプロシージャおよびトリガを削除して、新しいプロシージャ名で作成する。 プロシージャを実行するすべてのアプリケーションまたは SQL ソース・スクリプトを変更して、新しい名前を使用する。 プロシージャがリモートで別のサーバから呼び出される場合、リモート・サーバのアプリケーションを変更して新しい名前を使用する。
トリガ名	<ul style="list-style-type: none"> トリガを作成するすべての SQL ソース・スクリプトを変更する。
ルール名	<ul style="list-style-type: none"> ルールを作成するすべての SQL ソース・スクリプトを変更する。
デフォルト名	<ul style="list-style-type: none"> デフォルトを作成するすべての SQL ソース・スクリプトを変更する。
ユーザ定義のデータ型名	<ul style="list-style-type: none"> ユーザ定義データ型を持つテーブルを作成するすべてのプロシージャを削除して、再作成する。 ユーザ定義データ型を持つテーブルを作成するすべてのアプリケーションを変更する。
カラム名	<ul style="list-style-type: none"> カラムを参照するすべてのプロシージャ、トリガ、ビューを削除して、新しいカラム名で作成する。 <code>sp_depends</code> では、カラム名の参照は検出できない。次のクエリでは、“key” という名前のカラムを参照するプロシージャ、トリガ、およびビューの名前が表示される。 <pre> select distinct sysobjects.name from sysobjects, syscomments where sysobjects.id = syscomments.id and syscomments.text like "%key%" </pre> 名前によってカラムを参照するすべてのアプリケーションと SQL ソース・スクリプトを変更する。

次のコマンドでは、ビュー `isolation` の名前が `isolated` に変更されます。

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

次のコマンドでは、上のコマンドで名前が変更されたビュー `isolated` のカラム名が変更されます。

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

- `sp_depends` を使用すると、名前を変更するビュー、プロシージャ、またはテーブルを参照するすべてのビュー、プロシージャ、またはトリガのリストを表示させることができます。オブジェクト名を変更した後で `sp_depends` を使用する場合は、新しい名前を指定します。次に例を示します。

```
sp_depends new_name
```

`sp_renamedb` を使用したデータベース名の変更

データベース名を変更するには、`sp_renamedb` を使用します。データベースはシングルユーザ・モードでなければなりません。データベース名を明示的に参照するすべてのプロシージャ、トリガ、ビューを削除して、再作成します。詳細については、`sp_renamedb` を参照してください。

他の識別子の変更

- ユーザ名、ログイン名、デバイス名、リモート・サーバ名、リモート・サーバ・ユーザ名、セグメント名、文字セット、言語名を変更する場合は、最初に、オブジェクトやユーザを削除してから再び追加または作成できるかどうかを確認します。できない場合は、次のコマンドを使用して、システム・テーブルを直接更新できるようにします。

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

`allow updates to system tables` 設定パラメータを設定できるのは、システム・セキュリティ担当者だけです。

システム・テーブルに直接更新を行ったときのエラーは、**Adaptive Server** に重大な問題を発生させることがあります。[表 1-9](#) を参照して、オブジェクトまたはユーザを削除して再作成できるかどうかを確認してください。

[表 1-11 \(139 ページ\)](#) は、この識別子のセットに対して考えられる従属関係を示しています。オブジェクトを削除して再作成する方法、区切り識別子を使用する方法、またはシステム・テーブルを直接更新する方法のいずれかを選択してアップグレードする場合は、どの方法を選択しようとも、考えられる従属関係についてこの表を参照してください。

表 1-9: 識別子を変更する場合のシステム・テーブルの直接更新の代替法

識別子タイプ	システム・テーブルの更新を避けるための対策
ユーザ名およびログイン名	オブジェクトを所有していないユーザの名前を変更するには、まず各データベースで <code>sp_helprotect username</code> を使用してそのユーザのパーミッションを記録する。次に、すべてのデータベースからそのユーザを削除 (<code>sp_dropuser</code>) し、ログインを削除 (<code>sp_droplogin</code>) する。最後に、新しいログイン名を追加 (<code>sp_addlogin</code>) し、データベースに新しいユーザ名を追加 (<code>sp_adduser</code>) して、 <code>grant</code> によってそのユーザにパーミッションをリストアする。
デバイス名	デバイスが完全に割り付けられている場合は、 <code>create database</code> コマンドでデバイス名を使用する必要がないため、名前を変更しなくてよい。
リモート・サーバ名	リモート・サーバからログインするリモート・ログイン名の数が多くなければ、リモート・サーバを削除 (<code>sp_dropserver</code>) してから、新しい名前を追加 (<code>sp_addserver</code>) する。
リモート・サーバ・ログイン	<code>sp_dropremotelogin</code> によってリモート・ログインを削除し、 <code>sp_addremotelogin</code> を使用して新しい名前を付けて追加する。次に、 <code>grant</code> によって、プロシージャを実行するためのユーザのパーミッションをリストアする。
セグメント名	オブジェクトがセグメント上に作成された後は、ほとんど使用されない。
文字セット名と言語名	システム管理者が <code>sp_adddlanguage</code> を使用して代替言語を作成した場合には、言語および文字セットは識別子としてだけ予約語を持つことになる。 <code>sp_droplanguage</code> を使用してその言語を削除し、新しい名前を追加する。

警告! システム・テーブルを直接更新するのは、きわめて危険です。誤って更新してしまうと、**Adaptive Server** を実行できなくなったり、データベース内のオブジェクトにアクセスできなくなる場合があります。直接更新する場合は、他に問題がなく集中できる環境で、サーバでの運用アクティビティがないときに行ってください。可能なかぎり、[表 1-9](#) の代替方法を使用してください。

- 次の例では、ユーザ名を更新する場合の安全な手順を示します。すべてのデータの変更の前に `begin transaction` コマンドを付けます。システム・セキュリティ担当者が次のコマンドを実行します。

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

これで次のコマンドを実行できます。

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

この時点でクエリを実行して、変更対象のローだけにコマンドが作用したことを確認してください。複数のローに作用する識別子を変更することによってのみ、syslogins の language 名が変更されます。クエリの作用対象は次のとおりです。

- クエリが適切なローだけに作用していた場合は、**commit transaction** を使用します。
- クエリが複数のローや不適切なローに作用していた場合は、**rollback transaction** を使用して問題の原因を調べ、コマンドを正しく実行してください。

確認が終了すると、システム・セキュリティ担当者が次のコマンドを実行して、**allow updates to system tables** 設定パラメータをオフにします。

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

警告！システム・テーブルの更新は、単一のデータベースで、ユーザ定義のトランザクションごとに実行してください。複数のデータベースで **begin transaction** コマンドを発行して、テーブルを更新しないでください。そのような動作をすると、リカバリが非常に困難になります。

表 1-10 に、予約語を変更するときに更新するシステム・テーブルとカラムを示します。先頭に “master.dbo.” が付くテーブルは、master データベースだけに格納されています。他のテーブルはすべて、master データベースとユーザ・データベースに格納されています。更新を行う前に、適切なデータベースを使用していることを確認してください。次のコマンドによって現在のデータベース名を確認できます。

```
select db_name()
```

表 1-10: 識別子を変更する場合に更新するシステム・テーブルとカラム名

識別子のタイプ	更新するテーブル	カラム名
ユーザ名	sysusers	name
ログイン名	master.dbo.syslogins	name
セグメント名	syssegments	name
デバイス名	master.dbo.sysdevices	name
リモート・サーバ名	master.dbo.syssservers	srvname
リモート・サーバ・ネットワーク名	master.dbo.syssservers	srvnetname
文字セット名	master.dbo.syscharsets	name
言語名	master.dbo.syslanguages master.dbo.syslogins	name language

表 1-11 に、識別子を更新する場合に必要な注意事項を変更を示します。

表 1-11: 識別子を変更するときの注意事項

識別子	注意事項
ログイン名	このログイン名がユーザになっている各データベースで、ユーザ名を変更する。
ユーザ名	このユーザが所有するオブジェクトに対して修飾による参照 (<i>owner_name.object_name</i>) を使用するすべてのプロシージャ、トリガ、ビューは一度削除して編集してから再作成する必要がある。新しいユーザ名を使用するには、修飾されたオブジェクト名を使用するすべてのアプリケーションと SQL ソース・スクリプトを変更する。オブジェクト自体は削除する必要はない。これは、 sysusers はユーザ名ではなくユーザの ID を格納するカラムによって sysobjects にリンクされているためである。
デバイス名	新しいユーザ名を使用するには、デバイス名を参照するすべての SQL ソース・スクリプトまたはアプリケーションを変更する。
リモート・サーバ名	リモート・サーバ上の名前を変更する。 sp_checkreswords がレポートする名前がローカル・サーバ名の場合には、リモート・プロシージャ・コールを発行したり受け取る前に、サーバをリポートする必要がある。
リモート・サーバ・ネットワーク名	interfaces ファイルにあるサーバ名を変更する。
リモート・サーバ・ログイン名	リモート・サーバ上の名前を変更する。
セグメント名	セグメント名についてのテーブルまたはインデックスを作成するすべてのプロシージャを削除して再作成する。新しいユーザ名を使用するには、セグメントにオブジェクトを作成するすべてのアプリケーションを変更する。
文字セット名	なし
言語名	master.dbo.syslanguages と master.dbo.syslogins の両方を変更する。 syslogins を更新すると、多くのローが影響を受ける。また、使用しているローカライゼーション・ファイルの名前も変更する。

区切り識別子の使用

- 区切り識別子は、テーブル名、カラム名、ビュー名に使用できます。区切り識別子は、その他のオブジェクト名には使用できません。
- 区切り識別子を使用する場合は、`set quoted_identifier on` を使用し、識別子を使用するすべてのプロシージャ、トリガ、ビューを削除して、もう一度作成してください。これを行うには、予約語を二重引用符で囲み、すべての文字列を一重引用符で囲んでテキストを編集します。

次の例は、区切り識別子を使用するためのクエリへの変更を示します。この例では、`key` カラムと `level` カラムを持つ `work` という名前のテーブルを更新します。まず、文字リテラルを二重引用符で囲んでいる `10.0` 以前のリリースのクエリを示し、次に区切り識別子を使用した編集後のクエリを示します。

```
/* 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'
```

- 識別子として予約語を使用するすべてのアプリケーションには、次のような変更が必要です。
 - アプリケーションでは、区切り識別子のオプションをオンに設定してください。
 - 予約語を使用している識別子はすべて、二重引用符で囲んでください。
 - 区切り識別子オプションがオンのときにアプリケーションで使用されるすべての文字リテラルは、一重引用符で囲んでください。囲まなければ、**Adaptive Server** は文字リテラルをオブジェクト名として解釈しようとします。

たとえば、次のクエリではエラー・メッセージが表示されます。

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

次に正しいクエリを示します。

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

- 区切り識別子が有効である間に作成されたストアド・プロシージャは、オプションを設定しなくても実行できます (`allow updates to system tables` オプションの動作もこれと同じです)。つまり、区切り識別子モードをオンにして、ストアド・プロシージャを削除し、それを編集して識別子として使用される予約語を引用符で囲み、プロシージャを再作成することができます。すべてのユーザは、`set quoted_identifier` を使用しなくてもプロシージャを実行できます。

パーミッション

`sp_checkreswords` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — すべての入力パラメータ <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [set](#)

システム・プロシージャ [sp_configure](#)、[sp_depends](#)、[sp_rename](#)、[sp_renamedb](#)
ユーティリティ [defncopy](#)

sp_checksource

説明	ビュー、デフォルト、ルール、トリガ、プロシージャ、宣言デフォルト、検査制約、計算カラム、関数ベース・インデックス、述部などのコンパイル済みオブジェクトのソース・テキストが存在するかどうかを確認します。述部の名前は、ユーザ定義名でも内部名でもかまいません。
構文	<code>sp_checksource [objname [, tabname [, username]]]</code>
パラメータ	<p>objname ソース・テキストの存在を確認するコンパイル済みオブジェクトです。</p> <p>tabname すべての検査制約、デフォルト、トリガの存在を確認するテーブルまたはビューの名前です。</p> <p>username ソース・テキストの存在を確認するコンパイル済みオブジェクトを所有するユーザの名前です。</p>
例	<p>例 1 現在のデータベース内にあるすべてのコンパイル済みオブジェクトのソース・テキストの存在を確認します。</p> <pre>sp_checksource</pre> <p>例 2 <code>titleview</code> という名前のビューのソース・テキストの存在を確認します。</p> <pre>sp_checksource titleview</pre> <p>例 3 <code>Mary</code> が所有している <code>title_vu</code> という名前が付いているビューのソース・テキストの存在を確認します。</p> <pre>sp_checksource title_vu, @username = Mary</pre> <p>例 4 カスタム・ストアド・プロシージャ <code>list_phone_proc</code> のソース・テキストの存在を確認します。</p> <pre>sp_checksource list_phone_proc</pre> <p>例 5 <code>my_tab</code> という名前が付いたテーブル上で定義されたすべての検査制約、トリガ、宣言デフォルトのソース・テキストの存在を確認します。</p> <pre>sp_checksource @tabname = "my_tab"</pre> <p>例 6 <code>my_vu</code> ビューのソース・テキストの存在、および <code>my_tab</code> テーブル上で定義されたすべての検査制約、トリガ、宣言デフォルトのソース・テキストの存在を確認します。</p> <pre>sp_checksource @objname = "my_vu", @tabname = "my_tab"</pre>

例 7 Tom が所有するすべてのコンパイル済みオブジェクトのソース・テキストの存在を確認します。

```
sp_checksource @username = "Tom"
```

例 8 “pred1” 述部のソーステキストがあるかどうかを確認します。

```
sp_checksource pred1
```

使用法

- `sp_checksource` は、指定されたコンパイル済みオブジェクトのソース・テキストの存在を確認します。指定されたオブジェクトにソース・テキストが存在する場合、`sp_checksource` は 0 を返します。指定されたオブジェクトにソース・テキストが存在しなければ、`sp_checksource` は 1 を返します。
- パラメータを 1 つも指定しなければ、`sp_checksource` は、現在のデータベース内にあるすべてのコンパイル済みオブジェクトのソース・テキストの存在を確認します。
- `sp_checksource` をパラメータなしで使用するには、データベース所有者またはシステム管理者の権限が必要です。
- `sp_checksource` は、ユーザ定義関数のテキストを暗号化します。

パーミッション

`sp_checksource` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効

細密なパーミッションが有効の場合、他のユーザが所有しているコンパイル済みオブジェクトのソース・テキストの有無を確認するには、`manage database` 権限を持つユーザであることが必要。
すべてのユーザが `sp_checksource` を実行して自分自身のコンパイル済みオブジェクトのソース・テキストを確認できます。

細密なパーミッションが無効

細密なパーミッションが無効の場合、他のユーザが所有するコンパイル済みオブジェクトのソース・テキストの有無を確認するには、データベース所有者か `sa_role` を持つユーザであることが必要。
すべてのユーザが `sp_checksource` を実行して自分自身のコンパイル済みオブジェクトのソース・テキストを確認できます。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_hidetext](#)

sp_chgattribute

説明	max_rows_per_page、fillfactor、reservepagegap、exp_row_size の各値を変更して、テーブルまたはインデックスの今後の領域を割り付けます。また、テーブルに concurrency_opt_threshold を設定します。オプティミスティック・インデックス・ロック用のユーザ・インタフェースを提供します。
構文	<pre>sp_chgattribute objname, {"max_rows_per_page" "fillfactor" "reservepagegap" "exp_row_size" "concurrency_opt_threshold" "optimistic_index_lock" "identity_burn_max" "plldegree" "ptn_locking"}, value, optvalue sp_chgattribute objname, {"identity_gap", set_number "dealloc_first_txdpg" , value}</pre>
パラメータ	<p><i>objname</i> 属性を変更するテーブルまたはインデックスの名前です。</p> <p><i>max_rows_per_page</i> ローのサイズを指定します。このオプションは、可変長カラムを持つテーブルに使用します。</p> <p><i>fillfactor</i> Adaptive Server が、ロック・スキームを変更する reorg rebuild コマンドまたは alter table コマンドの結果としてインデックスを再作成またはテーブル・ページをコピーするときに、各ページを作成する方法を指定します。fillfactor の値は、インデックスを再構築するときにだけ使用します。有効な値は 0 ～ 100 です。</p> <p><i>reservepagegap</i> 埋められたページと、エクステンツ I/O 割り付けオペレーション中に残される空ページの比率を指定します。指定した各 num_pages につき、空のページ 1 つが、今後テーブルを拡張するために残されます。有効な値は 0 ～ 255 です。デフォルトは 0 です。</p> <p><i>exp_row_size</i> データオンリー・ロック・テーブルのローに指定した領域量を予約します。このオプションを使用すると、更新中に高コストになる可能性があるローの転送数を減らすことができます。有効な値は 0、1、およびテーブルのローの最小長および最大長の間の任意の値です。0 はサーバワイドな設定が適用されることを示し、1 はデータ・ページのローが完全にパックされることを示します。</p>

concurrency_opt_threshold

データオンリー・ロック・テーブルへのアクセスにおいて、同時実行性より I/O 削減のための最適化が開始されるテーブルサイズをページ数で指定します。テーブルが **concurrency_opt_threshold** によって指定された数のページ量より小さい場合は、常に、有効なインデックスを使用して同時実行性のために最適化されます。また、テーブルが **concurrency_opt_threshold** によって指定された数のページ量よりも大きい場合は、同時実行性ではなくクエリの I/O が最適化されます。有効な値は -1 ~ 32767 です。値を 0 に設定すると同時実行性の最適化が無効になります。32767 ページを超えるテーブルに同時実行性の最適化を実行するには、-1 を使用します。デフォルトは 15 ページです。

optimistic_index_lock

インデックスのルート・ページの競合をなくすパフォーマンス最適化を有効にします。インデックスの分割によるルート・ページの変更が必要な場合には、排他テーブルが取得されます。このため、**optimistic_index_lock** は、変更が比較的少ないテーブルでの使用に向いています。有効な値は、オプティミスティック・インデックス・ロックをオンにする場合は 1、オプティミスティック・インデックス・ロックをオフにする場合は 0 (デフォルト値) です。

identity_burn_max

テーブルの **identity** カラムで消去された最大値を設定できます。

identity_gap

identity ギャップを変更することを表します。

value

sp_chgattribute で指定する各種オプションの入力値を表す数値です。

optvalue

新しい値です。有効な値とデフォルト値は、指定されたパラメータによって異なります。このパラメータは、**identity_burn_max** パラメータでのみ使用されます。他のパラメータ値では、この値は NULL です。

set_number

アイデンティティ・ギャップの新しいサイズです。

dealloc_first_txdpg

text または **image** カラムを **null** に更新します。前に参照していた **text** または **image** ページを割り付け解除した後で、対応するテキスト・ポインタを **null** に設定します。この結果、**null** の **text** または **images** カラムに対する領域の割り付けが抑えられます。有効な値は、**null** 更新で **text** または **image** ページを割り付け解除しない場合は 0 (デフォルト値)、割り付け解除をオンに設定する場合は 1 です。

plldegree

クエリ・オプティマイザが使用できるスレッド数の最大数を指定します。

ptn_locking

パーティション・レベルでのロックを有効または無効にします。

例

例 1 **authors** テーブルの今後の領域の割り付けに対し、**max_rows_per_page** を 1 に設定します。

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

例 2 **titleidind** インデックスの今後の領域の割り付けに対し、**max_rows_per_page** を 4 に設定します。

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

例 3 **title_ix** のページに対して 90 % の **fillfactor** 値を指定します。

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

例 4 **authors** テーブルの今後の領域の割り付けに対し、**exp_row_size** を 120 に設定します。

```
sp_chgattribute "authors", "exp_row_size", 120
```

例 5 **titleidind** インデックスの今後の領域の割り付けに対し、**reservepagegap** を 16 に設定します。

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

例 6 **titles** テーブルの同時実行性最適化をオフにします。

```
sp_chgattribute "titles", "concurrency_opt_threshold", 0
```

例 7 **mytable** のアイデンティティ・ギャップを 20 に設定します。

```
sp_chgattribute "mytable", "identity_gap", 20
```

例 8 **identity_gap** の設定値の代わりに **"identity burning set factor"** の設定値を使用するように **mytable** を変更します。

```
sp_chgattribute "mytable", "identity_gap", 0
```

sp_chgattribute の値を 1 に設定し、オブティミスティック・インデックス・ロック機能をオンにします。

```
sp_chgattribute "mytable", "optimistic_index_lock", 1
```

sp_chgattribute の値を 0 に設定し、オブティミスティック・インデックス・ロック機能をオフにします。

```
sp_chgattribute "mytable", "optimistic_index_lock", 0
```

例 9 dealloc_first_txtpg を使用して、テキストおよびイメージの領域の割り付け解除をオンに切り替えます。

```
sp_chgattribute "mytable", "dealloc_first_txtpg", 1
```

この機能をオフに切り替えるには、次のように入力します。

```
sp_chgattribute "mytable", "dealloc_first_txtpg", 0
```

例 10 authors テーブルの identity_burn_max 値を 5 に変更します。

```
sp_chgattribute "authors", "identity_burn_max", 0, 5
```

例 11 クエリ・オブティマイザに対して、最大で 4 つのスレッドを使用するように指示します。

```
sp_chgattribute my_table, "plldegree", 4
```

十分なりソースが見つからない場合、クエリ・オブティマイザは 4 つ未満のスレッドを選択する場合があります。同様のメカニズムはインデックスにも適用できます。たとえば、次の例では、authors に存在する auth_ind という名前のインデックスを使用して、2 つのスレッドでアクセスします。

```
sp_chgattribute "authors.auth_ind", "plldegree", 4
```

sp_chgattribute は現在のデータベースから実行する必要があります。

使用法

- Adaptive Server の Cluster Edition では、sp_chgattribute を使用して *identity_gap* の値を実行時に変更することはできません。
- テーブルまたはインデックスの今後の領域割り付けやデータ変更のために、sp_chgattribute により、max_rows_per_page、fillfactor、reservepagegap、exp_row_size、または dealloc_first_txtpg の各値を変更します。既存のデータ・ページの領域の割り付けには影響しません。現在のデータベースにあるオブジェクトのこれらの値しか変更できません。
- テーブルの格納領域の管理値を確認するには、sp_help を使用します。インデックスの格納領域の管理値を確認するには、sp_helpindex を使用します。

- `max_rows_per_page` を 0 に設定すると、Adaptive Server は、ローの数を制限しないでデータ・ページまたはインデックス・ページを満たします (これは、`max_rows_per_page` が設定されていない場合の Adaptive Server のデフォルト動作です)。
- `sysobjects` に格納されている `identity_burn_max` 値と、現在の `identity` 値が、新しい値に設定されます。
- テーブルが空でない場合、`identity_burn_max` の新しい値は、`identity` カラムの現在の最大値以上であることが必要です。テーブルが空の場合は、有効な範囲の任意の正の値に設定できます。
- `max_rows_per` に小さい値を設定すると、ページ分割の原因になります。ページ分割は、あるページに新しいデータまたはインデックス・ローを追加しなければならないときに、新しいローを追加するための十分な領域がない場合に発生します。通常、既存ページのデータは、新しく割り当てられるページと既存ページの間で均等に分割されます。ノンクラスタード・インデックスの最大値を概算するには、ページ・サイズから 32 を引き、その答えをインデックス・キー・サイズで割ります。次の文は、ノンクラスタード・インデックス `titleind` の `max_rows_per_page` の最大値を計算しています。

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

288

- `max_rows_per_page`、`fillfactor`、`reservepagegap`、または `exp_row_size` に無効な値を指定すると、`sp_chgattribute` は、有効な値を指示するエラー・メッセージを返します。
- `max_rows_per_page`、`fillfactor`、`reservepagegap`、`exp_row_size`、および `concurrency_opt_threshold` の詳細については、『パフォーマンス & チューニング・シリーズ』を参照してください。
- `identity` ギャップの詳細については、『Transact-SQL ユーザーズ・ガイド』の「第 7 章 データベースおよびテーブルの作成」の「テーブルの `identity` ギャップの管理」を参照してください。
- このストアド・プロシージャは、トランザクション内からは実行できません。
- `sa_role` 権限を持つユーザだけが、このストアド・プロシージャを実行できます。

- データページまたはデータロー・ロック・スキームを持つテーブルに対しては、`OPTIMISTIC`・`INDEX`・`LOCK`・`OPTION`を設定できません。
- システム・データベースにあるテーブル (`master` や `tempdb` など) に対しては、`OPTIMISTIC`・`INDEX`・`LOCK`・`OPTION`を設定できません。設定できるのは、ユーザ定義テーブルに対してのみです。
- `text` および `image` ページは、`NULL` 更新を実行しても領域が割り付けられたままです。`dealloc_first_txtpg` を使用すると、これらの空のテキスト ページをテーブルから削除できます。
カラムが新たに更新されると、`text` または `image` ページが再割り付けされます。

パーミッション

`sp_chgattribute` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、オブジェクト所有者であることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、オブジェクトの所有者であることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – <code>NULL</code> • <i>Previous value</i> – <code>NULL</code> • <i>Current value</i> – <code>NULL</code> • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [alter table](#)、[create index](#)、[create table](#)
システム・プロシージャ [sp_helpindex](#)

sp_cleanpwdchecks

説明	sp_cleanpwdchecks はカスタム・ストアド・プロシージャであり、ユーザ定義のテーブルでログインおよびパスワード関連の属性を削除する日時と方法を定義できます。
構文	sp_cleanpwdchecks, <i>login_name</i>
パラメータ	<i>login_name</i> 実行するクリーンアップのログイン名を指定します。
使用法	sp_cleanpwdchecks はユーザ定義で、ログインを削除したときに master データベースで動的に呼び出されます。

sp_clearpsexex

説明	<code>sp_setpsexex</code> が設定した Adaptive Server セッションの実行属性をクリアします。
構文	<code>sp_clearpsexex spid, exeattr</code>
パラメータ	<p><i>spid</i> 実行属性がクリアされるセッションのプロセス ID です。</p> <p><i>exeattr</i> クリアされる実行属性を特定します。 <i>exeattr</i> の値は、“priority”と“enginegroup”です。</p>
例	<p>プロセス 12 のエンジン・グループ・エントリを削除します。</p> <pre>sp_clearpsexex 12, 'enginegroup'</pre>
使用法	<ul style="list-style-type: none"> セッションのライフタイムの間に実行属性がクリアされない場合は、セッションが終了するか、または異常終了するときに、実行属性がクリアされます。 対応するエンジン・グループにオンライン・エンジンがない場合、<code>sp_clearpsexex</code> は実行されません。 エンジン・グループ・エントリを削除すると、セッションはクラス定義またはデフォルト・クラスで決定されたエンジン・グループ上で実行されます。 プロセス ID (<i>spid</i>) をリストするには、<code>sp_who</code> を使用します。
パーミッション	<code>sp_clearpsexex</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	<p>細密なパーミッションが有効の場合、<code>manage any execution class</code> 権限を持つユーザであることが必要。</p> <p>すべてのユーザが <code>sp_clearpsexex</code> を実行して自分の所有しているタスクの優先度属性をクリアできます。</p>
細密なパーミッションが無効	<p>細密なパーミッションが無効の場合、<code>sa_role</code> が付与されたユーザであることが必要。</p> <p>すべてのユーザが <code>sp_clearpsexex</code> を実行して自分の所有しているタスクの優先度属性をクリアできます。</p>
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル『パフォーマンス&チューニング・シリーズ』

システム・プロシージャ [sp_addexclass](#)、[sp_bindexclass](#)、[sp_dropexclass](#)、[sp_showexclass](#)、[sp_unbindexclass](#)

sp_clearstats

説明 すべてのサーバ・ユーザまたは指定ユーザの新しいアカウント期間を開始します。[sp_reportstats](#) を実行して、前回の期間の統計を表示します。

構文 `sp_clearstats [loginame]`

パラメータ `loginame`
ユーザのログイン名です。

例 **例 1** ユーザ全員の新しいアカウント期間を開始します。

```
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.
```

例 2 ユーザ “kathy” の新しいアカウント期間を開始します。

```
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.
```

- 使用方法**
- `sp_clearstats` はアカウント期間を生成します。期間の終わりにだけ実行するようにしてください。
 - `sp_clearstats` は、アカウント統計をクリアします。したがって、この統計を記録してからプロシージャを実行してください。
 - `sp_clearstats` は `syslogins` の `accddate` フィールドを更新し、`syslogins` の `totcpu` フィールドと `totio` フィールドをクリアします。

パーミッション `sp_clearstats` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ `sp_reportstats`

sp_client_addr

説明 クライアント・アプリケーションが付加されている、Adaptive Server の各タスクの IP (インターネット・プロトコル) アドレスを表示します。spid とクライアント・ホスト名も表示されます。

構文 sp_client addr [spid]

パラメータ spid

IP アドレスを必要とする 1 つのタスクを指定します。

例 1 すべてのタスクの IP アドレスがリストされます。

```
sp_client_addr
-----
spid  hostname  ipaddr
-----
11    FRED      162.66.131.36
21    BARNEY    162.66.100.233
22    WILMA     162.66.100.206
23    BETTY     162.66.100.119
24    PEBBLES   162.66.100.125
25    BAMBAM    162.66.100.124
(6 rows affected)
(return status = 0)
```

例 2 spid 21 の IP アドレスを表示します。

```
sp_client_addr 21
-----
spid  hostname  ipaddr
-----
21    BARNEY    162.66.100.233
(1 row affected)
(return status = 0)
```

例 3 クライアント・アプリケーションが IP 経由で接続されなかった場合の結果を表示します。

```
sp_client_addr 11
-----
spid  hostname  ipaddr
-----
11    FRED      0.0.0.0
(1 row affected)
(return status = 0)
```

例 4 クライアントを持たないタスク (たとえば、Housekeeper) の結果を表示します。

```
sp_client_addr 9
-----
spid  hostname  ipaddr
-----
9                                NULL
(1 row affected)
(return status = 0)
```

例 5 無効な spid が指定された場合の結果を表示します。

```
sp_client_addr 99
-----
Msg 18934, Level 16, State 1:
Procedure "sp_client_addr", Line 32:
spid が見つかりません。
(return status = 1)
```

使用法

- クライアント・アプリケーションが IP によって付加されていない場合、アドレスは 0.0.0.0 と表示されます。Adaptive Server は、IP 以外のプロトコルのアドレス表示をサポートしません。
- タスクがクライアント (たとえば、Housekeeper) を持たない場合、IP アドレスは “NULL” として表示されます。パラメータを指定せずに sp_client_addr を使用すると、クライアントが付加されていないタスクはリストされません。

パーミッション

すべてのユーザが sp_client_addr を実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_who](#)

sp_cluster

説明	(クラスタ環境のみ) クラスタに関係する複数のプロシーダを実行します。
構文	<p>接続を別の論理クラスタまたはインスタンスにマイグレートする。</p> <pre>sp_cluster connection, migrate, lc_name, instance_name, "spid_list"</pre> <p>以前の接続マイグレーションが待機中かどうかを調べ、待機中ならそのマイグレーションをキャンセルする。</p> <pre>sp_cluster connection, ['migrate_status' 'migrate_cancel'], 'spid_list'</pre> <p>未処理のまま残っているアクションを変更する。アクションのキャンセルやタイミングの変更など。</p> <pre>sp_cluster logical, "action", lc_name, { cancel, action_handle modify_time, action_handle, wait_option[, timeout] release, action_handle }</pre> <p>論理クラスタに1つのリソースか1つまたは複数のルートを追加する。</p> <pre>sp_cluster logical, "add", lc_name, { route, route_type, key_list instance, instance_list failover, instance_list }</pre> <p>ルートを1つの論理クラスタから別のクラスタに移動する。</p> <pre>sp_cluster logical, "alter", lc_name, route, route_type, key_list</pre> <p>新しい論理クラスタを作成する。</p> <pre>sp_cluster logical, "create", lc_name</pre> <p>論理クラスタを、1つまたは複数のインスタンスで停止、または論理クラスタ全体を停止して、インスタンスまたは論理クラスタを非アクティブな状態にする。</p> <pre>sp_cluster logical, "deactivate", lc_name, { "cluster" "instance", instance_list } [, wait_option[, timeout[, @handle output]]]</pre> <p>論理クラスタを削除する。または論理クラスタから1つまたは複数のリソースを削除する。</p> <pre>sp_cluster logical, "drop", lc_name, {cluster instance, instance_list failover, instance_list route, route_type, key_list }</pre> <p>元のベース・インスタンスを復活させて、手動フェールオーバーをリバースする。</p> <pre>sp_cluster logical, "failback", lc_name, { cluster[, wait_option[, timeout[, @handle output]]] </pre>


```
instance, from_instance_list, to_instance_list[, wait_option[,
    timeout[, @handle output ]]] }
```

ベース・インスタンスからフェールオーバー・インスタンスへの手動フェールオーバーを開始する。

```
sp_cluster logical, "failover", lc_name, {cluster
    [, to_instance_list[, wait_option[, timeout[, @handle output ]]]
    | instance, from_instance_list, to_instance_list[, wait_option[,
    timeout[, @handle output ]]] }
```

接続グループを手動で集めて別の論理クラスタにマイグレートする。

```
sp_cluster logical, 'gather', lc_name
```

sp_cluster logical の完全な構文を表示する。

```
sp_cluster logical, "help"
```

論理クラスタを 1 つまたは複数のインスタンス上で停止する。または論理クラスタ全体を停止する。

```
sp_cluster logical, "offline", lc_name,
    {cluster | instance, instance_list}
    [, wait_option[, timeout[, @handle output ]]]
```

1 つまたは複数のインスタンス上でデフォルトの論理クラスタを開始する。

```
sp_cluster logical, "online", { lc_name[, instance_list]}
```

論理クラスタのルール (open 論理クラスタ、フェールオーバー・モード、システム・ビュー、起動モード、負荷プロファイル) を設定する。

```
sp_cluster logical, "set", lc_name, { open
    | failover, failover_mode | system_view, view_mode
    | startup, { automatic | manual } | load_profile, profile_name }
    login_distribution, { affinity | "round-robin" }
```

論理クラスタに関する情報を表示する。

```
sp_cluster logical, "show"
    [, lc_name[, {action[, state] | route[, type[, key]}]]]
```

論理クラスタの負荷プロファイルを設定し管理する。

```
sp_cluster profile, ["show" [, profile_name]
    | "create", profile_name | "drop", profile_name
    | "set", profile_name [, weight [, wt_metric [, wt_value] ]
    | threshold [, thr_metric [, thr_value ]]]]
```

論理クラスタの負荷プロファイルを設定し管理する。

```
sp_cluster profile, ["show" [, profile_name] | "create", profile_name |
    "drop", profile_name | "set", profile_name [, weight [, wt_metric [,
    wt_value] ] | threshold [, thr_metric [, thr_value ]]]]
```

パラメータ

`sp_cluster connection, migrate, lc_name, instance_name, "spid_list"`

各パラメータの意味は次のとおりです。

- `lc_name` は論理クラスタの名前。
- `instance_name` はインスタンスの名前。
- `spid_list` はマイグレートする `spid` のリスト。複数の `spid` はセミコロンで区切ります。

`sp_cluster connection, ['emigrate_status' | 'migrate_cancel'][, 'spid_list']`

各パラメータの意味は次のとおりです。

- `spid_list` は調べる `spid` のリスト。
- `migrate_cancel` は接続のマイグレーションをキャンセルすることを示します。
- `migrate_status` は接続マイグレーションのステータスを調べることを示します。

`sp_cluster logical, "action", lc_name, {cancel, action_handle | modify_time, action_handle, wait_option[, timeout] | release, action_handle}`

各パラメータの意味は次のとおりです。

- `cancel` はアクションのキャンセルを指定します。
- `action_handle` はアクションの識別子。
- `modify_time` はアクション時刻の変更を指定します。
- `wait_option` はアクション時刻の変更方法。有効な値は次のとおりです。
 - `wait` — 現存の接続をマイグレートまたは切断するまでに一定の待ち時間 (`timeout` が指定されていない場合は無期限) を与えることを示します。
 - `nowait` — 現存の接続を直ちにマイグレートまたは切断することを示します。
 - `until` — 現存の接続のマイグレートまたは切断が指定された時刻までに行われることを示します。
- `timeout` は時間 (`wait` と共に使用したとき) または時刻 (`until` と共に使用したとき)。形式は 24 時間表記の “hh:mm:ss” です。たとえば、`timeout` では 11:30 p.m. が 23:30:00 になります。
- `release` は完了したアクションが保持するすべてのリソースを解放するよう指定します。

```
sp_cluster logical, "add", lc_name, { route, route_type, key_list | instance,  
instance_list | failover, instance_list}
```

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。
- *route* は論理クラスタに 1 つまたは複数のルートを追加するよう指定します。
- *route_type* は追加するルートのタイプ。有効な値は次のとおりです。
 - *application* — アプリケーションの名前に論理クラスタへのルートを指定します。
 - *login* — ログイン名に論理クラスタへのルートを指定します。
 - *alias* — サーバ名エイリアスに論理クラスタへのルートを指定します。
- *key_list* は、ルート・タイプに応じて、アプリケーション、ログイン、またはエイリアスのリスト。キー・リスト内の要素はセミコロンで区切ります。
- *instance* は論理クラスタに 1 つまたは複数のベース・インスタンスを追加するよう指定します。
- *instance_list* は追加するインスタンスのリスト。複数のインスタンスはセミコロンで区切ります。
- *failover* は論理クラスタに 1 つまたは複数のフェールオーバー・インスタンスを追加するよう指定します。

sp_cluster logical, "alter", *lc_name*, route, *route_type*, *key_list*

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。
- route はルートを変更するよう指定します。
- *route_type* は変更するルートのタイプ。有効な値は次のとおりです。
 - application — アプリケーションの名前に論理クラスタへのルート指定します。
 - login — ログイン名に論理クラスタへのルート指定します。
 - alias — サーバ名エイリアスに論理クラスタへのルート指定します。
- *key_list* は、ルート・タイプに応じて、アプリケーション、ログイン、またはエイリアスのリスト。キー・リスト内の要素はセミコロンで区切ります。

sp_cluster logical, "create", *lc_name*

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。

```
sp_cluster logical, "deactivate", lc_name, { "cluster" ["instance", instance_list]
[, wait_option[, timeout[, @handle output ]]]
```

- *lc_name* は論理クラスタの名前。
- *cluster* はクラスタ全体を指定します。
- *instance* は論理クラスタ内の特定のインスタンスのみを非アクティブにするよう指定します。
- *instance_list* は論理クラスタ内の選択されたインスタンスのリスト。
- *wait_option* は次のオプションの 1 つ。
 - *wait* — 現存の接続をマイグレートまたは切断するまでに一定の待ち時間 (*timeout* が指定されていない場合は無期限) を与えることを示します。
 - *nowait* — 現存の接続を直ちにマイグレートまたは切断することを示します。
 - *until* — 現存の接続のマイグレートまたは切断が指定された時刻までに行われることを示します。
- *timeout* は時間 (*wait* と共に使用したとき) または時刻 (*until* と共に使用したとき)。形式は 24 時間表記の “hh:mm:ss” です。たとえば、*timeout* では 11:30 p.m. が 23:30:00 になります。
- *@handle output* はアクションのアクション・ハンドルが取得されるように指定します。

sp_cluster logical, "drop", *lc_name*, { cluster | instance, *instance_list* | failover, *instance_list* | route, *route_type*, *key_list* }

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。
- cluster はクラスタ全体を指定します。
- instance は論理クラスタ内の特定のインスタンスのみを非アクティブにするよう指定します。
- *instance_list* は論理クラスタ内の選択されたインスタンスのリスト。
- *wait_option* は次のオプションの 1 つ。
 - wait — 現存の接続をマイグレートまたは切断するまでに一定の待ち時間 (*timeout* が指定されていない場合は無期限) を与えることを示します。
 - nowait — 現存の接続を直ちにマイグレートまたは切断することを示します。
 - until — 現存の接続のマイグレートまたは切断が指定された時刻までに行われることを示します。
- *timeout* は時間 (wait と共に使用したとき) または時刻 (until と共に使用したとき)。形式は 24 時間表記の “hh:mm:ss” です。たとえば、*timeout* では 11:30 p.m. が 23:30:00 になります。
- @handle output はアクションのアクション・ハンドルが取得されるように指定します。

```
sp_cluster logical, "drop", lc_name, {cluster[, wait_option[, timeout[, @handle  
output ]]] | instance, from_instance_list, to_instance_list[, wait_option[, timeout[,  
@handle output ]]] }
```

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。
- *cluster* はクラスタ全体を指定します。
- *to_instance_list* は事前定義されたフェールオーバー・インスタンスのリスト。NULL 値は最初のフェールオーバー・グループをアクティブにします。
- *from_instance_list* オフラインにするインスタンスのリスト。
- *wait_option* は次のオプションの 1 つ。
 - *wait* — 現存の接続をマイグレートまたは切断するまでに一定の待ち時間 (*timeout* が指定されていない場合は無期限) を与えることを示します。
 - *nowait* — 現存の接続を直ちにマイグレートまたは切断することを示します。
 - *until* — 現存の接続のマイグレートまたは切断が指定された時刻までに行われることを示します。
- *timeout* は時間 (*wait* と共に使用したとき) または時刻 (*until* と共に使用したとき)。形式は 24 時間表記の “hh:mm:ss” です。たとえば、*timeout* では 11:30 p.m. が 23:30:00 になります。
- *@handle output* はアクションのアクション・ハンドルが取得されるように指定します。

sp_cluster logical, "failback", *lc_name*, {cluster[, *to_instance_list*[, *wait_option*[, *timeout*[, @handle output]]] | instance, *from_instance_list*, *to_instance_list*[, *wait_option*[, *timeout*[, @handle output]]] }

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。
- cluster は論理クラスタ全体のフェールオーバを指定します。
- *to_instance_list* は事前定義されたフェールオーバ・インスタンスのリスト。NULL 値は最初のフェールオーバ・グループをアクティブにします。
- *wait_option* はアクション時刻の記録方法。有効な値は次のとおりです。
 - wait — 現存の接続をマイグレートまたは切断するまでに一定の待ち時間 (*timeout* が指定されていない場合は無期限) を与えることを示します。
 - nowait — 現存の接続を直ちにマイグレートまたは切断することを示します。
 - until — 現存の接続のマイグレートまたは切断が指定された時刻までに行われることを示します。
- *timeout* は時間 (*wait* と共に使用したとき) または時刻 (*until* と共に使用したとき)。形式は 24 時間表記の “hh:mm:ss” です。たとえば、*timeout* では 11:30 pm が 23:30:00 になります。
- @handle output はフェールオーバのアクション・ハンドルが取得されるように指定します。
- instance は論理クラスタの選択されたインスタンスのみがフェールオーバするよう指定します。
- *from_instance_list* はオフラインにするインスタンスのリスト。

sp_cluster logical, 'gather', *lc_name*

各パラメータの意味は次のとおりです。

- gather は条件を満たす接続のセットのみを集めて別の論理クラスタにマイグレートすることを示します。
- *lc_name* は接続のマイグレート先の論理クラスタの名前。


```
sp_cluster logical, "offline", lc_name, { cluster | instance, instance_list }
[, wait_option[, timeout[, @handle output ]]]
```

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。
- *cluster* はクラスタ全体を指定します。
- *instance* は論理クラスタの選択されたインスタンスのみがオフラインになるよう指定します。
- *instance_list* は論理クラスタ内の選択されたインスタンスのリスト。
- *wait_option* はアクション時刻の指定方法。有効な値は次のとおりです。
 - *wait* — 現存の接続をマイグレートまたは切断するまでに一定の待ち時間 (*timeout* が指定されていない場合は無期限) を与えることを示します。
 - *nowait* — 現存の接続を直ちにマイグレートまたは切断することを示します。
 - *until* — 現存の接続のマイグレートまたは切断が指定された時刻までに行われることを示します。
- *timeout* は時間 (*wait* と共に使用したとき) または時刻 (*until* と共に使用したとき)。形式は 24 時間表記の “hh:mm:ss” です。たとえば、*timeout* では 11:30 pm が 23:30:00 になります。
- *@handle output* はアクションのアクション・ハンドルが取得されるように指定します。
- *from_instance_list* はオフラインにするインスタンスのリスト。

```
sp_cluster logical, "online", { lc_name[, instance_list]}
```

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。
- *instance_list* は論理クラスタ内の選択されたインスタンスのリスト。

```
sp_cluster logical, "set", lc_name, {open | failover, failover_mode | system_view,
view_mode | startup, { automatic | manual } | load_profile, profile_name |
action_release, { automatic | manual } | gather, { automatic | manual } |
login_distribution, { affinity | "round-robin" }
```

各パラメータの意味は次のとおりです。

- *lc_name* は論理クラスタの名前。
- open は open 論理クラスタを設定します。ルート指定が解除された接続は open 論理クラスタに送られます。
- failover *failover_mode* は論理クラスタのフェールオーバー・モードを設定します。 *failover_mode* の値は次のとおりです。
 - instance — 1:1 フェールオーバー方式 (ベース・インスタンスが失敗するたびに、フェールオーバー・リソースがオンラインになる) を指定します。
 - group — クラスタ内のすべてのベース・インスタンスが失敗したときに初めてフェールオーバー・リソースがオンラインになるよう指定します。
- system_view *view_mode* は論理クラスタ内で実行されているタスクのデフォルト・システム・ビューを設定します。 *view_mode* の値は次のとおりです。
 - instance — sp_who、sp_lock、モニタリング・テーブルなどの監視と情報ツールでインスタンスが記述されるように指定します。
 - cluster — sp_who、sp_lock、モニタリング・テーブルなどの監視と情報ツールでクラスタ全体が記述されるように指定します。
- startup { automatic | manual } は論理クラスタの起動モードを設定します。
 - automatic — はクラスタが開始すると論理クラスタが自動的に起動されるよう指定します。
 - manual — 論理クラスタが手動で起動されるよう指定します。
- login_distribution は論理クラスタが複数のインスタンスにまたがっているとき、クラスタ・エディションが接続をどう配分するかを指定します。
- action_release は論理クラスタのアクション (online、offline、failover、および failback) の完了またはキャンセル後の自動的な解放およびクリアを有効または無効にします。

- **automatic** — 論理クラスタのアクションが自動でクリアされるように指定します。
 - **manual** — 論理クラスタのアクションが完了またはキャンセルされた後にクリアされないように指定します。これがデフォルト値です。
 - **gather** は、事前に定義されたアクション (**online**、**add route**、**alter route**、または **drop route**) のいずれかが発生した場合に、別の論理クラスタに接続のグループを移動するかどうかを指定します。
 - **automatic** — 接続が自動で移動されるように指定します。
 - **manual** — 接続が自動では移動されないように指定します。これがデフォルト値です。
 - **@handle output** はアクションのアクション・ハンドルが取得されるように指定します。
 - **from_instance_list** はオフラインにするインスタンスのリスト。
- `sp_cluster logical, "show"[, lc_name[, {action[, state] | route[, type[, key]]]`
各パラメータの意味は次のとおりです。
- **lc_name** は論理クラスタの名前。NULL が入力されると、すべての論理クラスタの情報の要約が表示されます。
 - **action** は管理アクション (**failover**、**failback**、**online**、**offline**、**deactivate**) の情報を指定します。
 - **state** は **cancelled**、**complete**、**active** の 1 つ。
 - **route** はルートについての情報を指定します。
 - **type** は **application**、**alias**、**login** の 1 つ。
 - **key** は特定のログイン名、エイリアス名、またはアプリケーション名。

```
sp_cluster profile, [ "show" [, profile_name] | "create", profile_name | "drop",  
profile_name | "set", profile_name [, weight [, wt_metric [, wt_value] | threshold [,  
thr_metric [, thr_value]]]]
```

- `show` は設定されている負荷プロファイルとその設定値を表示します。
- `profile_name` は負荷プロファイルの名前。
- `creates` は新しい負荷プロファイルを作成します。
- `drop` は負荷プロファイルを削除します。
- `set` は負荷プロファイルの属性を指定します。属性は個別に設定します。
- `weight` は重み付け属性を指定します。
- `wt_metric` は個々の重み付け指標。有効な値は次のとおりです。
 - `user connections` — 使用できるリソースに基づいた、インスタンスが新しい接続を受け入れる能力。
 - `cpu utilization` — 使用できるリソースに基づいた、インスタンスが新しい接続を受け入れる能力。
 - `run queue` — 使用できるリソースに基づいた、インスタンスが新しい接続を受け入れる能力。
 - `io load` — 未処理のまま残っている非同期 I/O 数。
 - `engine deficit` — クラスタのインスタンス間にあるオンライン・エンジンの数の差異。

注意 `engine deficit` はクラスタ内のインスタンス間にエンジン数の違いがあるときにしか測定できません。`engine deficit` は負荷スコアに最大相対能力を反映する指標を追加します。

- `user metric` — 省略可能な、ユーザ定義の指標。
- `wt_value` は重み付けの値。有効な値は 0 ~ 255。重み付けゼロ (0) はその指標を計算から除外します。
- `threshold` はスレッシュールド属性を指定します。
- `thr_metric` は個々のスレッシュールド属性。有効な値は次のとおりです。
 - `dynamic` — 動的負荷分散のスレッシュールドを指定します。
 - `login` — ログイン・リダイレクトのスレッシュールドを指定します。

- **hysteresis** — 接続リダイレクトに要求される最低負荷スコアを指定します。
- **thr_value** は **thr_metric** の値に依存します。 **thr_metric** の値に対応して次のようになります。
 - **dynamic** または **login** では、 **thr_value** は 2 つのインスタンス間の負荷スコアのパーセント単位の差異。有効な値は 0 ~ 100。重み付けゼロ (0) はその形態の負荷分散を無効にします。
 - **hysteresis** では、 **thr_value** は動的負荷分散またはログイン・リダイレクトが行われる前に、ターゲット・インスタンスが達していなければならない最低負荷スコア。

例

例 1 spid が 73 の接続を SalesLC クラスタに移動します。

```
sp_cluster connection, migrate, SalesLC, NULL, '73'
```

例 2 現在の接続を “ase3” インスタンスに移動します。

```
sp_cluster connection, migrate, NULL, ase3
```

例 3 spid の値が 73 と 75 の接続を “ase3” インスタンスと SalesLC クラスタに移動します。

```
sp_cluster connection, migrate, SalesLC, ase3, '73;75'
```

例 4 spid 73 が接続マイグレーションの対象になっている場合、Cluster Edition はそのマイグレーションをキャンセルします。

```
sp_cluster connection, 'migrate_cancel', '73'
```

例 5 spid の値が 73 の接続の移動ステータスを調べます。

```
sp_cluster connection, 'migrate_status', '73'
```

SPID	LogicalCluster	Instance	Migration	LogicalCluster	Migration	Instance	Command
73	SystemLC	ase1	SalesLC	ase3	connection	migrate	

例 6 “SalesLC” という論理クラスタ上の時間指定付きアクションをキャンセルします。アクションのハンドルは 4390 です。

```
sp_cluster logical, "action", SalesLC, cancel, "4390"
```

例 7 既存のアクション 5364 の待機オプションを **nowait** に変更します。

```
sp_cluster logical, "action", SalesLC, modify_time, "5364", nowait
```

例 8 “SalesLC” という論理クラスタに対するアクション 3456 を解放します。

```
sp_cluster logical, "action", SalesLC, release, "3456"
```

例 9 “SalesLC” という論理クラスタに対するアクションで、完了またはキャンセルになったアクションをすべて解放します。

```
sp_cluster logical, "action", SalesLC, release, "all"
```

例 10 インスタンス “ase1” と “ase2” を “SalesLC” という論理クラスタに追加します。

```
sp_cluster logical, "add", SalesLC, instance,
"ase1;ase2"
```

例 11 “ase3” を使って “SalesLC” にフェールオーバー・グループを 1 つ作成します。

```
sp_cluster logical, "add", SalesLC, failover, ase3
```

例 12 ログイン “tom”、“dick”、および “harry” を “SalesLC” という論理クラスタに送るようにルートを指定します。

```
sp_cluster logical, "add", SalesLC, route, login,
"tom;dick;harry"
```

例 13 field_sales アプリケーションを “SalesLC” という論理クラスタに送るようにルートを指定します。

```
sp_cluster logical, "add", SalesLC, route, application,
field_sales
```

例 14 “SalesLC” というエイリアスを使って論理クラスタ “lc1” にルート・タイプがエイリアスのルートを作成します。次にそのルートに関連付けられている論理クラスタを “lc1” から “lc2” に変更します。ルートはタイプ (alias) とキー (SalesLC) で識別されます。

```
sp_cluster logical, "add", "lc1", "route", "alias", "SalesLC"
sp_cluster logical, "alter", "lc2", "route", "alias", "SalesLC"
```

例 15 “SalesLC” という名前の論理クラスタを作成します。

```
sp_cluster logical, "create", SalesLC
```

例 16 “SalesLC” 論理クラスタ内のすべてのインスタンスを直ちに停止して、“SalesLC” を非アクティブな状態にします。

```
sp_cluster logical, "deactivate", SalesLC, cluster, nowait
```

例 17 “SalesLC” 論理クラスタ内のインスタンス “ase1” と “ase2” を停止して、“SalesLC” を非アクティブな状態にします。

```
sp_cluster logical, "deactivate", SalesLC, instance, "ase1;ase2"
```

例 18 “SalesLC” 論理クラスタを削除します。

```
sp_cluster logical, "drop", SalesLC, cluster
```

例 19 “SalesLC” 論理クラスタからベース・インスタンス “ase1” と “ase2” を削除します。

```
sp_cluster logical, "drop", SalesLC, instance, "ase1;ase2"
```

例 20 “SalesLC” 論理クラスタからアプリケーション field_sales と web_sales のルート指定を削除します。

```
sp_cluster logical "drop", SalesLC, route, application,
"field_sales;web_sales"
```

例 21 “SalesLC” 論理クラスタをフェールバックさせます。

```
sp_cluster logical, "failback", SalesLC, cluster
```

例 22 “SalesLC” は “ase3” と “ase1” で稼働中です。この例では、“ase3” が “ase1” にフェールバックし、“SalesLC” は “ase2” で稼働を続けます。このアクションは 2 分で開始します。

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "failback", SalesLC, instance,
ase3, ase1, wait, "00:02:00", @handle = @out_handle
output
```

例 23 “SalesLC” 論理クラスタを事前に定義されているフェールオーバー・リソースの最初のグループにフェールオーバーさせます。フェールオーバーは接続を終了する前に 2 分間待ちます。

```
declare @out_handle varchar(15)

execute
sp_cluster logical, "failover", SalesLC, cluster, NULL, wait, "00:02:00",
@handle = @out_handle output

Action '2' has been issued for the 'failover cluster' command.

Logical Cluster Handle      Action          From      To
      State          InstancesWaiting ConnectionsRemaining WaitType
      StartTime          Deadline          CompleteTime
-----
SalesLC          2 failover cluster      2, 4      NULL
complete          0                          0 wait
Nov 15 2007  3:23PM      Nov 15 2007  3:25PM      Nov 15 2007  3:23PM
```

Remember to issue the 'sp_cluster logical, action, <logical cluster name>, release, <handle>' command for any cancelled or completed actions.

例 24 “SalesLC”は“ase1”と“ase2”で稼働中です。この例では、“ase1”が“ase3”にフェールオーバーし、“SalesLC”は“ase2”で実行を続けます。待機オプションが指定されていないので、デフォルトの無期限待機が使用されます。

```
sp_cluster logical, "failover", SalesLC, instance,
ase1, ase3
```

Action '1' has been issued for the 'failover instance' command.

Logical Cluster	Handle	Action	From	To	State	InstancesWaiting
ConnectionsRemaining	WaitType	StartTime	Deadline	CompleteTime		
SalesLC	1	failover instance	1	4	complete	0
	0	infinite	Nov 15 2007 3:06PM	NULL	Nov 15 2007 3:06PM	

Remember to issue the `sp_cluster logical, action, <logical cluster name>, release, <handle>' command for any cancelled or completed actions.

例 25 接続グループを集めて new_stores 論理クラスタへ移動します。

```
sp_cluster logical, 'gather', new_stores
```

例 26 sp_cluster logical ストアド・プロシージャの構文を表示します。

```
sp_cluster logical, "help"
```

```
Usage for sp_cluster 'logical':
sp_cluster 'logical', 'help' [, <module>]
```

To show the logical cluster configuration:

```
sp_cluster 'logical', 'show'
sp_cluster 'logical', 'show', <lcname>
sp_cluster 'logical', 'show', <lcname> | NULL, 'action' [, <state>]
sp_cluster 'logical', 'show', <lcname> | NULL, 'route' [, <type [, <key>]]
```

To create a logical cluster:

```
sp_cluster 'logical', 'create', <lcname>
```

To add resources to a logical cluster:

```
sp_cluster 'logical', 'add', <lcname>, 'failover', <instance_list> [, <group>]
sp_cluster 'logical', 'add', <lcname>, 'instance', <instance_list>
sp_cluster 'logical', 'add', <lcname>, 'route', <route_type>, <key_list>
```

To drop resources from a logical cluster:

```
sp_cluster 'logical', 'drop', <lcname>, 'cluster'
sp_cluster 'logical', 'drop', <lcname>, 'failover', <instance_list>
sp_cluster 'logical', 'drop', <lcname>, 'instance', <instance_list>
```



```
sp_cluster 'logical', 'drop', <lcname>, 'route', <route_type>, <key_list>
```

Argument details:

<lcname> is a logical cluster nam

<instance_list> is a ';' separated list of instance

<route_type> is one of {'user', 'application', 'alias

<key_list> is a ';' separated list of keys

To set attributes of a logical cluster:

```
sp_cluster 'logical', 'set', <lcname>, 'open'
```

```
sp_cluster 'logical', 'set', <lcname>, 'down_routing', 'disconnect' | 'system' |
'open'
```

```
sp_cluster 'logical', 'set', <lcname>, 'failover', 'instance' | 'group'
```

```
sp_cluster 'logical', 'set', <lcname>, 'load_profile', <profile_name>
```

```
sp_cluster 'logical', 'set', <lcname>, 'startup', 'automatic' | 'manual'
```

```
sp_cluster 'logical', 'set', <lcname>, 'system_view', 'instance' | 'cluster'
```

To start and stop a logical cluster:

```
sp_cluster 'logical', 'online', <lcname>[, <instance_list>]
```

```
sp_cluster 'logical', 'offline', <lcname>, 'cluster'[, <wait_option>[, <time>[,
@handle output]]]
```

```
sp_cluster 'logical', 'offline', <lcname>, 'instance',
```

```
<instance_list>[, <wait_option>[, <time>[, @handle output]]]
```

To failover and failback a logical cluster:

```
sp_cluster 'logical', 'failover', <lcname>, 'cluster'[, <instance_list>[,
<wait_option>[, <time>[, @handle output]]]]]
```

```
sp_cluster 'logical', 'failover', <lcname>, 'instance', <from_instance_list>,
<instance_list>[, <wait_option>[, <time>[, @handle output]]]
```

```
sp_cluster 'logical', 'failback', <lcname>, 'cluster'[, <instance_list>[,
<wait_option>[, <time>[, @handle output]]]]]
```

```
sp_cluster 'logical', 'failback', <lcname>, 'instance', <from_instance_list>,
<instance_list>[, <wait_option>[, <time>[, @handle output]]]
```

To work with action handles:

```
sp_cluster 'logical', 'action', <lcname>, 'cancel', <handle>
```

```
sp_cluster 'logical', 'action', <lcname>, 'modify_time', <handle>, <wait_option>[,
<time>]
```

```
sp_cluster 'logical', 'action', <lcname>, 'release', <handle>
```

Argument details:

<wait_option> is one of {'nowait', 'wait', 'until'}

<time> is a time in hh:mm:ss format

<handle> is an action handle

例 27 “SalesLC” 論理クラスタ内のすべてのインスタンスを直ちに停止して、“SalesLC” をオフライン状態にします。

```
sp_cluster logical, "offline", SalesLC, cluster, nowait
```

例 28 “SalesLC” 論理クラスタ内のインスタンス “ase1” と “ase2” を停止して、“SalesLC” をオフライン状態にします。

```
sp_cluster logical, "offline", SalesLC, instance, "ase1;ase2"
```

例 29 “SalesLC” 論理クラスタ内のすべてのベース・インスタンスを開始してクラスタをオンラインにします。

```
sp_cluster logical, "online", SalesLC
```

例 30 “SalesLC” 論理クラスタ内の “ase1” インスタンスを開始してクラスタをオンラインにします。

```
sp_cluster logical, "online", SalesLC, ase1
```

例 31 “SalesLC” 論理クラスタの負荷プロファイルを Sybase プロファイル `sybase_profile_oltp` に設定します。

```
sp_cluster logical, "set", SalesLC, load_profile,
sybase_profile_oltp
```

例 32 デフォルト・システム・ビューを `cluster` に設定します。

```
sp_cluster logical, "set", SalesLC, system_view, cluster
```

例 33 設定されているすべての論理クラスタに関する情報の要約を表示します。

```
sp_cluster logical, "show", NULL
```

ID	Name	State	Online	Instances	Connections
1	mycluster	online	4		1
2	SalesLC	online	2		0
3	HRLC	online	1		0
4	CatchallLC	offline	0		0

Logical cluster 'mycluster' is the system logical cluster.
 Logical cluster 'CatchallLC' is the open logical cluster.

Logical Cluster	Instance	State	Type	Connections	Load Score
HRLC	silk	online	base	0	0.01
SalesLC	cotton	offline	failover	0	0.00
SalesLC	linen	online	base	0	0.00
SalesLC	silk	offline	failover	0	0.01
SalesLC	wool	online	base	0	0.01
mycluster	cotton	online	base	0	0.00
mycluster	linen	online	base	0	0.00

```
mycluster      silk      online    base      0      0.01
mycluster      wool      online    base      1      0.01
```

例 34 未処理のまま残っているすべてのアクションのリストを表示します。

```
sp_cluster logical, "show", NULL, action
```

例 35 SalesLC 論理クラスタの情報を表示します。

```
sp_cluster logical, "show", SalesLC
ID          Name          State          Online Instances  Connections
-----
2           OrderLC       online         1                 0
```

```
Instance    State    Type    Connections Load Score Failover Gro
-----
asedemo1    online  base      0          0.78  NU
```

```
Attribute                                     Setting
-----
Down Routing Mode                             system
Failover Mode                                 instance with fail_to_any
LC Roles                                       none
Load Profile                                  sybase_profile_oltp
Login Distribution                            affinity
Startup Mode                                  automatic
System View                                    cluster
```

```
Route Type          Route Key
-----
application          order_app
```

```
Logical cluster 'OrderLC' has no associated actions.
(return status = 0)
```

例 36 負荷プロファイル “my_profile” を作成します。

```
sp_cluster profile, "create", my_profile
```

例 37 “my_profile” の測定基準の重みを指定します。sp_cluster profile — 統計指標の重みを my_profile に設定します。user connections はゼロに設定されているためプロファイから除外されます。

```
sp_cluster profile, "set", my_profile, weight, "user connections", '0'
sp_cluster profile, "set", my_profile, weight, cpu utilization, '20'
sp_cluster profile, "set", my_profile, weight, runqueue, '30'
sp_cluster profile, "set", my_profile, weight, io load, '10'
```


使用法

sp_cluster connection

現在の `spid` をマイグレートするには、`spid_list` を `sp_cluster connection, migrate` から省略します。

sp_cluster logical, action

- `monLogicalClusterAction` テーブルをクエリしてアクション・ハンドルを取得します。

```
sp_cluster logical, "show", NULL, action
```

- マイグレーションをサポートしないクライアントは、SQL バッチが完了してオープン・トランザクションがなくなったとき、または `timeout` の期間が切れたとき、どちらが先になっても、接続を失います。
- `timeout` 期間の終わりに残っていたクライアントは接続を失います。
- アクションのキャンセルはそのアクションをロールバックしません。設定を元の状態に戻すには、追加のタスクが必要になる場合があります。
- 完了したアクションのみが解放できます。アクションの解放は、完了したアクションをシステムと `monLogicalClusterAction` テーブルから削除します。

sp_cluster logical, 'add'

- ベース・インスタンスとフェールオーバー・リソースはシステム論理クラスタに追加できません。
- 複数のインスタンス、フェールオーバー・リソース、またはアプリケーションはセミコロンで区切ります。
- フェールオーバー・インスタンスをグループ化して、フェールオーバー・グループを作成できます。グループ名の指定がない場合は、新しいグループが作成され、インスタンスはそのグループに追加されます。インスタンスを入れるグループを指定できます(グループ番号は引用符で囲みます)。

次に例を示します。

```
1> sp_cluster logical, 'add', tempLC, failover, "asedemo3;asedemo2"
2> go
```

```
Added failover instance 'asedemo3' to group 1 for logical cluster 'tempLC'.
Added failover instance 'asedemo2' to group 1 for logical cluster 'tempLC'.
```

次にインスタンスをグループに追加します。

```
1> sp_cluster logical, 'add', tempLC, failover, asedemo4, "4"  
2> go
```

Added failover instance 'asedemo4' to group 4 for logical cluster 'tempLC'.

sp_cluster logical, "deactivate"

- **deactivate** コマンドはシステム論理クラスタに使用できません。
- **offline** は **deactivate** と同じですが、**deactivate** は停止したインスタンスやクラスタを非アクティブ状態にするのに対し、**offline** はオフライン状態にします。

sp_cluster logical "drop"

- インスタンスやフェールオーバー・リソースは削除する前にオフライン状態にする必要があります。
- クラスタを削除すると、そのクラスタに関連付けられているすべてのルート、リソース、設定も削除されます。

sp_cluster logical "failback"

フェールバックを開始するには、その前に論理クラスタがフェールオーバーされている必要があります。

sp_cluster logical "gather"

- 接続を手動で集めるには、論理クラスタがオンラインになっている必要があります。
- 接続を集めるには、論理クラスタにルートの定義が必要です。

sp_cluster logical, "offline"

- **offline** コマンドはシステム論理クラスタに使用できません。
- **offline** は **deactivate** と同じですが、**deactivate** は停止されたインスタンスやクラスタを非アクティブ状態にします。

sp_cluster logical "online"

online コマンドはシステム論理クラスタに使用できません。

sp_cluster logical "set"

オープン・プロパティを持つことのできる論理クラスタは1つだけです。新しい論理クラスタにオープン・プロパティを設定すると、それまでオープンだった論理クラスタからオープン・プロパティが削除されます。

sp_cluster profile

- Sybase が提供する統計指標と互換性を持たせるために、ユーザ統計指標は正規化される必要があります。反応時間を測定するユーザ統計指標を例にとると、反応時間の上限が 10 秒で、測定された値が 5 秒の場合、統計指標は 50、つまり $(5/10 \times 100 = 50)$ です。

- スレッシュホールドは負荷の不均衡がどの程度になったら接続を別のインスタンスにリダイレクトするかを設定するための指標です。**Workload Manager** はターゲット・インスタンスと負荷が最も軽いインスタンスとの間の負荷スコアの差(% 単位)がスレッシュホールド値に達すると、接続をリダイレクトします。
 - ヒステリシスの値は、負荷スコアの差がスレッシュホールド値に達しても、インスタンスの負荷スコア自体は低い(たとえば、2 と 8)のでリダイレクトが適切でない場合に、リダイレクトを防止するためのものです。

パーミッション

`sp_cluster` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage cluster</code> 権限または <code>ha_role</code> を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> または <code>ha_role</code> を持つユーザであることが必要。

sp_clusterlockusage

説明 (クラスタ環境のみ) クラスタ内のフリー・ロック、使用中のロック、保持されているロックについてレポートします。

構文 sp_clusterlockusage

例 クラスタ内で現在使用されているロックについてレポートします。

Lock Usage	count	% of total
-----	-----	-----
Total Locks	95039	n/a
Free Locks	85807	90.29 %
Used Locks	9232	9.71 %
Object Locks	4032	4.24 %
Physical Locks	233	0.25 %
Partition Locks	9	0.00 %
Table Locks	0	0.00 %
Page Locks	0	0.00 %
Row Locks	17	0.02 %
Others	501	0.53 %
Retention Used	0	0.00 %

使用法

- Retention Used はタスクに所有されていないが、ロック保持のためクラスタ・レベルで所有されているロックの数です。

パーミッション sp_clusterlockusage のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage cluster privilege を持つユーザか ha_role を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role または ha_role を持つユーザであることが必要。

sp_cmp_all_qplans

説明	2つの抽象プラン・グループにあるすべての抽象プランを比較します。
構文	<code>sp_cmp_all_qplans group1, group2 [, mode]</code>
パラメータ	<p><i>group1, group2</i> 2つの抽象プラン・グループの名前です。</p> <p><i>mode</i> 表示オプションです。各表示オプションで表示されるモードとレポート内容は次のとおりです。</p> <ul style="list-style-type: none"> • counts — デフォルト・モード。このオプションでは次の分類でプラン数が表示されます。 <ul style="list-style-type: none"> • 同一のプラン • 同じ関連付けキーを持つが、グループが異なるプラン • 一方のグループに存在し、もう一方のグループに存在しないプラン • brief — counts で取得される情報のほかに、次を表示します。 <ul style="list-style-type: none"> • プラン自体は異なるが関連付けキーが同じであるような各グループの抽象プランの ID • 片方のグループにあって別のグループにはないプランの ID • same — すべてのカウントと、クエリとプランが一致するすべての抽象プランの ID、クエリ、プラン。 • diff — すべてのカウントと、クエリとプランが異なるすべての抽象プランの ID、クエリ、プラン。 • first — 1番目のプラン・グループにあるが2番目のプラン・グループにはないすべての抽象プランの総数、ID、クエリ、およびプラン。 • second — 1番目のプラン・グループにはないが2番目のプラン・グループにはあるすべての抽象プランの総数、ID、クエリ、およびプラン。 • offending — すべてのカウントと、結合キーが異なるか、または両方のグループには属さないすべての抽象プランの ID、クエリ、プラン。これは、diff、first、second の各モードの組み合わせ。これは、diff、first、second の各モードの組み合わせ。 • full — すべてのカウントと、すべての抽象プランの ID、クエリ、プラン。これは、same と offending の2モードを合わせたもの。

例 **例 1** 2つの抽象プラン・グループについてデフォルト・レポートを生成します。

```
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
```

例 2 brief モードを使用してレポートを生成します。

```
sp_cmp_all_qplans dev_plans, prod_plans, brief
```

使用法

- `sp_cmp_all_qplans` を使用して、2つのプラン・グループ内にある抽象プランの相違点を調べます。
- `sp_cmp_all_qplans` は、各グループ内のプランが同じユーザ ID とクエリ・テキストを持っている場合に、これらのプランをペアで対応させます。プランは次の区分で分類されます。
 - 同じプラン。
 - 両方のグループで同じ結合キーを持つが、異なる抽象プランを持つプラン。結合キーはグループ ID、ユーザ ID、およびクエリ・テキストである。
 - 一方のグループには存在するが、もう一方グループには存在しないプラン。
- 個々の抽象プランを比較するには、`sp_cmp_qplans` を使用します。抽象プラン・グループの名前を確認するには、`sp_help_qpgroup` を使用します。

- システム管理者またはデータベース所有者が `sp_cmp_all_qplans` を実行すると、2つのグループ内にあるすべてのプランについてレポートされます。別のユーザが `sp_cmp_all_qplans` を実行すると、そのユーザの ID を持つプランについてのみレポートされます。

パーミッション

`sp_cmp_all_qplans` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザか <code>monitor qp performance</code> 権限を持つユーザであることが必要。 自分が所有するプランの比較は、すべてのユーザが実行できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。 自分が所有するプランの比較は、すべてのユーザが実行できる。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — すべての入力パラメータ <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_cmp_qplans](#)、[sp_help_qpgroup](#)

sp_cmp_qplans

説明 2つの抽象プランを比較します。

構文 `sp_cmp_qplans id1, id2`

パラメータ `id1, id2`
2つの抽象プランの ID です。

例 **例 1** 抽象プランの 411252620 と 1383780087 を比較します。

```
sp_cmp_qplans 411252620, 1383780087
```

同じクエリです。
同じクエリ・プランです。

例 2 抽象プランの 2091258605 と 647777465 を比較します。

```
sp_cmp_qplans 2091258605, 647777465
```

同じクエリです。
異なるクエリ・プランです。

使用法

- `sp_cmp_qplans` は、クエリ、抽象プラン、および2つの抽象プランのハッシュ・キーを比較して、クエリが同じであるかどうか、またプランが同じであるかどうかをレポートします。クエリについては、次のいずれかのメッセージを出力します。
 - 同じクエリです。
 - 異なるクエリです。
 - 異なるクエリですが、同じハッシュ・キーを持っています。抽象プランについては、次のいずれかのメッセージを出力します。
 - 同じクエリ・プランです。
 - 異なるクエリ・プランです。
- `sp_cmp_qplans` も、比較の結果を示すリターン・ステータスを出力します。ステータス値 1、2、10 は追加できます。表 1-12 は、ステータス値を示しています。

表 1-12: `sp_cmp_qplans` のリターン・ステータス値

戻り値	意味
0	クエリ・テキストと抽象プランが同じである。
+1	クエリとハッシュ・キーが異なる。
+2	クエリが異なるが、ハッシュ・キーは同じである。
+10	抽象プランが異なる。
100	プラン ID の一方、または両方が存在しない。

- プランの ID を検索するには、`sp_help_qpgroup` または `sp_find_qplan` を使用します。また、プラン ID も `create plan` によって返され、`showplan` の出力に含まれます。

パーミッション

`sp_cmp_qplans` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限または <code>monitor qp performance</code> 権限を持つユーザであることが必要。 自分が所有するプランの比較は、すべてのユーザが実行できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。 自分が所有するプランの比較は、すべてのユーザが実行できる。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ `sp_cmp_all_qplans`、`sp_help_qpgroup`

sp_commonkey

説明 2つのテーブルまたはビューの間の共通キー（頻繁にジョインされるカラム）を定義する。

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

パラメータ *tablename*
ジョインされる1つ目のテーブルまたはビューの名前です。

tabbname
ジョインされる2つ目のテーブルまたはビューの名前です。

col1a
tablename という名前のテーブルまたはビューの共通キーを構成する最初のカラム名です。少なくとも1つのペアのカラム（1つ目のテーブルまたはビューから1カラム、2つ目のテーブルまたはビューから1カラム）を指定します。

col1b
tablename という名前のテーブルまたはビューの *col1a* とジョインされる、*tabbname* というテーブルまたはビュー内の相手側カラムの名前です。

例 **例 1** `titles.titleid` と `titleauthor.titleid` 上に共通キーを定義します。

```
sp_commonkey titles, titleauthor, title_id, title_id
```

例 2 `empid` という名前のカラムを持つ、`projects` と `departments` の2つのテーブルがある場合、この文により、2つのカラムで頻繁に使用されるジョインが定義されます。

```
sp_commonkey projects, departments, empid, empid
```

- **使用法**
 - 共通キーは、データベース設計では暗黙的な論理関係を、明示的な論理関係にするために作成されます。この情報は、アプリケーションによって使用できます。ただし、`sp_commonkey` は参照整合性制約を実行しません。キーの関係を設定するには、`create table` コマンドまたは `alter table` コマンドの `primary key` 句と `foreign key` 句を使用します。
 - `sp_commonkey` を実行すると、`syskeys` システム・テーブルにキーが追加されます。定義されている共通キーのレポートを表示するには、`sp_helpkey` を使用します。
 - 共通キーを定義するには、2つのテーブルまたはビューのうち、少なくともいずれか一方の所有者でなければなりません。

- 1 つ目のテーブルまたはビューからのカラム数と、2 つ目のテーブルまたはビューからのカラム数は、同じにしてください。それぞれのテーブルまたはビューからは、最大 8 つのカラムを共通キーにすることができます。共通カラムのデータ型も一致させてください。長さを指定するカラムの場合、長さは異なってもかまいません。共通カラムの null タイプは、一致しなくてもかまいません。
- インストール・プロセスでは、システム・テーブルの該当するカラム上で `sp_commonkey` が実行されます。
- `sp_commonkey` には Java データ型を使用できません。

パーミッション

`sp_commonkey` を実行するには、テーブルの所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [alter table](#)、[create table](#)、[create trigger](#)

システム・プロシージャ [sp_dropkey](#)、[sp_foreignkey](#)、[sp_helpjoins](#)、[sp_helpkey](#)、[sp_primarykey](#)

sp_companion

説明 高可用性システムのセカンダリ・コンパニオンとして Adaptive Server を設定し、コンパニオン・サーバをフェールオーバー・モードから別のモードへ移行させるなどのクラスタ・オペレーションを実行します。sp_companion は、セカンダリ・コンパニオンから実行されます。

構文

```
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}
```

パラメータ *server_name*
クラスタ・オペレーションを実行する Adaptive Server の名前です。

configure
server_name によって指定されたサーバを、フェールオーバー設定のプライマリ・コンパニオンとして設定します。

drop
フェールオーバー構成からコンパニオンを永久的に削除します。コマンドが終了すると、サーバはシングル・サーバ・モードになります。

suspend
フェールオーバー構成からコンパニオンを一時的に削除します。コマンドが終了すると、コンパニオンはサスペンド・モードになります。

resume
suspend コマンドをリバースして、コンパニオン間で通常のコンパニオン・モードを再開します。

prepare_failback
セカンダリ・コンパニオンを準備してプライマリ・コンパニオンのリソースを解放し、フェールバックできるようにします。

do_advisory

セカンダリ・コンパニオンが、フェールオーバー・モード中にプライマリ・コンパニオンの機能を正常に実行するための互換性を持っているかどうかを確認します。

- **all** — do_advisory にすべてのパラメータを検査させます。
- **help** — do_advisory パラメータに関する情報と構文を表示します。
- **group_attribute_name** — sp_companion を使用してレポートするグループ属性の名前です。
- **base_attribute_name** — sp_companion do_advisory を使用してレポートする基本属性の名前です。

with_proxydb

このパラメータがコンパニオン・サーバの初期設定に含まれている場合に、システム・データベースを除くすべてのデータベース、および追加される後続のすべてのデータベースのセカンダリ・コンパニオンにプロキシ・データベースを作成します。デフォルトでは、with_proxydb は無効です。

srvlogin

コンパニオン・サーバにアクセスするユーザのログインです。デフォルトでは、srvlogin の値は “sa” です。

srvpassword

コンパニオン・サーバにアクセスするユーザのパスワードです。デフォルトでは、srvpassword の値は null です。

cluster_login

クラスタにログインするユーザのログインです。デフォルトでは、cluster_login は “sa” です。

cluspassword

クラスタにログインするために指定する必要がある、ユーザのパスワードです。デフォルトでは、cluspassword の値は null です。

例

例 1 Adaptive Server MONEY1 をプライマリ・コンパニオンとして設定します。

```
sp_companion "MONEY1", configure
```

例 2 Adaptive Server MONEY1 をプライマリ・コンパニオンとして設定し、セカンダリ・コンパニオン上にプロキシ・データベースを作成します。

```
sp_companion "MONEY1", configure, with_proxydb, "sa", "sapsswd"
```

例 3 Adaptive Server PERSONEL1 をフェールオーバ構成から削除します。コマンドが完了すると、プライマリ・コンパニオンとセカンダリ・コンパニオンは、いずれもシングル・サーバ・モードになります。

```
sp_companion "PERSONEL1", "drop"
```

例 4 コンパニオン・サーバ (この例では MONEY1) の通常のコンパニオン・モードを再開します。

```
sp_companion "MONEY1", "resume"
```

例 5 プライマリ・コンパニオン (この場合は PERSONEL1) を準備して通常のコンパニオン・モードを変更し、フェールオーバした Adaptive Server の制御を再開します。

```
sp_companion "PERSONEL1", "prepare_failback"
```

例 6 PERSONEL1 コンパニオンでのクラスタ・オペレーションが正常に実行されるかどうかを確認します。この例の `do_advisory` は `all` パラメータを使用するので、PERSONEL1 の `do_advisory` 属性すべてを調べて、正常なクラスタ・オペレーションを妨げるパラメータがないことを確認します。また、フェールオーバの完了後にセカンダリ・コンパニオンがプライマリ・コンパニオンのオペレーションを正常に実行できるかどうかを確認します。

```
sp_companion "PERSONEL1", do_advisory, "all"
```

例 7 コンパニオン・サーバ上の CIS (コンポーネント統合サービス) のどの属性も、ローカル・サーバとの互換性を持っていないことを確認します。

```
sp_companion "PERSONEL1", do_advisory, "CIS"
```

使用法

- `sp_companion` は、Adaptive Server を高可用性システム内のセカンダリ・コンパニオンとして Adaptive Server を設定するクラスタ・オペレーションを実行します。また、`sp_companion` は、コンパニオン・サーバをフェールオーバ・モードから別のモード (たとえば、通常のコンパニオン・モード) へ移行させます。`sp_companion` は、セカンダリ・コンパニオンから起動されます。
- `sp_companion` は、`installmaster` スクリプトではなく、`installhasvss` (Windows NT では `insthasv`) によってインストールされます。`installhasvss` は `$SYBASE_ASE` の `scripts` サブディレクトリに格納されています。
- `sp_companion` は、Sybase のミラーリングを自動的に停止します。Sybase では、サードパーティのミラーリング・ソフトウェアを使用して、ディスク障害からデータを保護することをおすすめします。

詳細については、『高可用性システムにおける Sybase フェールオーバーの使用』を参照してください。do_advisory コマンドを実行する前に、do_advisory に関する章だけでなく、このマニュアルの、設定に関する章も必ずお読みください。

パーミッション

sp_companion を実行するには、ha_role を持つユーザであることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

sp_compatmode

説明	フル互換モード <code>compatibility mode</code> を使用できるかどうか確認します。
構文	<code>sp_compatmode</code>
例	<pre>lsp_compatmode ----- 互換モードは有効です。 警告: Compatibility mode may not be used when statement cache and literalautoparam are enabled. 警告: The configuration option 'histogram tuning factor' is configured with value '20', which is not the default value in ASE 12.5.This may lead to different accuracy of statistics and different query plans. (return status = 0) 1></pre>
使用法	このクエリは <code>compatibility mode</code> が有効かどうかをレポートします。 <code>enable compatibility mode</code> の使用で競合が発生する場合は警告が出ます。
パーミッション	<code>sp_compatmode</code> は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。
参照	詳細については、『マイグレーション技術ガイド』を参照してください。

sp_config_dump

説明 ダンプ設定をリスト、追加、変更できる。

```
構文 sp_config_dump
      [@config_name = 'configuration_name'
      [, {
        [@stripe_dir = 'stripe_dir_name'],
        [@ext_api = 'external_api'],
        [@num_stripes = 'number_of_stripes'],
        [@retry = 'number_of_retries'],
        [@blocksize = 'number_of_bytes'],
        [@compression = 'compression_level'],
        [@retain_days = 'number_of_days'],
        [@init = '[noinit | init]'],
        [@verify = '[header | full]'],
        [@notify = '[client | operator_console]'],
        [@backup_srv_name = backup_server_name'],
      } | ['delete']]
```

パラメータ

@config_name = 'configuration_name'

特定のダンプ設定を追加または変更するために必要なユニークなダンプ設定名です。'configuration_name'を指定しないと、すべてのダンプ設定が表示されます。その他のパラメータを指定した場合は、それらのパラメータが新しい値に変更されます。

@stripe_dir = 'stripe_dir_name'

ダンプ・オペレーション中にファイルがアーカイブされるディレクトリです。通常、アーカイブ・ファイルには次の形式の名前が付けられます。

database_name.dump_type.date-timestamp.stripeID

@stripe_dir のデフォルトは、Backup Server の起動ディレクトリです。

@ext_api = 'external_api'

ダンプ・オペレーションに使用する外部 API (バイト・ストリーム・デバイス) の名前です。デフォルトでは、このパラメータは使用されません。external_api は、次の形式で指定してください。

external_API_name::additional_options

@num_stripes = 'number_of_stripes'

ダンプ・オペレーションで使用するストライプ・デバイスの数です。デフォルトは 1 です。

@retry = 'number_of_retries'

致命的でないエラーの場合にサーバがダンプ・オペレーションを試行する回数です。有効な値は 0 ~ 5 で、デフォルトは 0 (再試行しない) です。

@blocksize = 'number_of_bytes'

ダンプ・オペレーションのブロック・サイズです。デバイスのデフォルトのブロック・サイズを無効にします。この値は、データベース・ページ 1 ページ分 (ほとんどのシステムでは 2048 バイト) 以上で、かつデータベース・ページ・サイズの倍数でなければなりません。最適のパフォーマンスを得るには、**blocksize** を 2 の累乗 (65,536、131,072、262,144 など) で指定します。

@compression = 'compression_level'

圧縮ダンプの圧縮レベルです。デフォルトでは、圧縮は無効になっています。

@retain_days = 'number_of_days'

Backup Server がダンプの上書きを防止する日数です。期限内のボリュームに対して上書き要求があると、Backup Server はユーザに確認を求めます。デフォルト値は 0 で、これはダンプが上書き可能であること意味します。

@init = '[noinit | init]'

ボリュームを初期化するかどうかを指定します。デフォルトは **noinit** です。

@verify = '[header | full]'

Backup Server でアーカイブにコピーするデータ・ページに対して最小限のページ・ヘッダー検査または完全なロー構造検査を実行するかどうかを指定します。グローバル・アロケーション・マップ (GAM)、オブジェクト・アロケーション・マップ (OAM)、アロケーション・ページ、インデックス、テキスト、ログ・ページについては、構造検査が実行されません。デフォルトでは、アーカイブ時にデータ・ページの検査は行いません。

@notify = '[client | operator_console]'

ダンプを開始したクライアント端末、または Backup Server が実行されているオペレータ・コンソール端末に Backup Server からメッセージを送るかどうかを指定します。

```
@backup_srv_name = 'backup_server_name'
```

ダンプ・デバイスが接続されているマシン上で稼働しているリモート Backup Server のネットワーク名を指定します。デフォルトの Backup Server である SYB_BACKUP にダンプするときは、`backup_server_name` を使用しないでください。このオプションを使用して、リモート Backup Server を 32 個まで指定できます。

`interfaces` ファイルを使用するプラットフォームでは、Backup Server 名が `interfaces` ファイルに登録されている必要があります。

```
'delete'
```

削除するダンプ設定を指定します。

例

例 1 すべてのダンプ設定を表示します。

```
sp_config_dump
go

Configuration name
-----
dmp_cfg1
dmp_cfg2
dmp_cfg3
```

例 2 `dmp_cfg1` という名前のダンプ設定のパラメータ値を一覧表示します。

```
sp_config_dump 'dmp_cfg1'
go

Dump configuration:dmp_cfg1

Option name          Option value
-----
compression          5
num_stripes          3
stripe_dir           /work/dump_dir
```

例 3 `dmp_cfg2` という名前の新しいダンプ設定を作成します。ここでは、ダンプ・オペレーションが `/work1/dmp_dir` ストライプ・ディレクトリに 5 個のストライプを作成し、致命的でないエラーが発生した場合には 1 回再試行するよう指定します。

```
sp_config_dump 'dmp_cfg2',
    @stripe_dir='/work1/dmp_dir', @num_stripes='5',
    @retry='1'
```

例 4 既存のダンプ設定のストライプ・ディレクトリを変更します。

```
sp_config_dump 'dmp_cfg2',
```

```
@stripe_dir='/work2/dmp_dir'
```

例 5 ダンプ設定を削除します。

```
sp_config_dump 'dmp_cfg2', 'delete'
```

使用法

パーミッション sp_config_dump のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage dump configuration 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role または oper_role を持つユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オブション 監査対象のコマンド またはアクセス	extrainfo の情報
exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

参照 マニュアル ダンプ・オペレーションの詳細については、『システム管理ガイド』を参照してください。

コマンド dump、load、genddonly

sp_configure

説明	グループ別の設定パラメータ、その現在の値、デフォルト値、一番最後に設定された値、この設定が使用するメモリ量を表示します。表示レベルがユーザの表示レベルと同等またはそれより低いパラメータだけを表示します。
構文	<pre>sp_configure [configname [, configvalue] group_name non_unique_parameter_fragment] 'drop instance' [, instance_name] [display_nondefault_settings] sp_configure "configuration file", 0, {"write" "read" "verify" "restore"} "file_name"</pre>
パラメータ	<p>configname 現在の値、デフォルト値、一番最後に変更された値、<i>parameter</i> に一致するすべてのパラメータの設定で使用されているメモリ量を表示します。</p> <p>configvalue <i>configname</i> を <i>configvalue</i> に再設定し、現在の値、デフォルト値、設定された値、<i>configname</i> が使用するメモリ量を表示します。</p> <p><code>sp_configure configname, 0, "default"</code> は、<i>configname</i> をデフォルト値に再設定し、現在の値、デフォルト値、設定された値、<i>configname</i> で使用されているメモリ量を表示します。</p> <p>group_name <i>group_name</i> のすべての設定パラメータ、その現在の値、デフォルト値、一番最後に設定された値 (該当するものがある場合)、この特定の設定が使用するメモリ量を表示します。</p> <p>non_unique_parameter_fragment <i>non_unique_parameter_fragment</i>、その現在の値、デフォルト値、設定された値、使用されるメモリ量を表示します。</p> <p>drop instance インスタンス固有の設定値を削除できます。</p> <p>instance_name クラスタ環境内 — インスタンス固有オプションの設定対象となるインスタンスを示します。</p> <p>display_nondefault_settings デフォルト値とは異なる設定値または実行値を持つ設定オプションを表示します。</p>

write

現在の設定から *file_name* を作成します。*file_name* がすでに存在する場合は、エラー・ログにメッセージが書き込まれます。また、既存のファイルは、*file_name.001*、*file_name.002* という規則によって名前が変更されます。静的なパラメータを変更した後サーバを再起動していない場合、“write” は、そのパラメータの現在実行中の値を返します。

read

file_name 内にある値の正当性を検証し、検証をパスした値をサーバに読み込みます。*file_name* で見つからないパラメータがあると、それらのパラメータに、現在実行されている値が使用されます。

verify

file_name 内にある値の正当性を検証します。

restore

file_name を、*sysconfigures* の値を使って作成します。このコマンドは、設定ファイルのコピーがすべて消失してしまった場合に新しいコピーを生成するときに便利です。

file_name

sp_configure を使用するファイルの名前。

例 **例 1** グループ別にすべての設定パラメータ、その現在の値、デフォルト値、一番最後に設定された値(該当するものがある場合)、およびこの設定値が使用するメモリ量を表示します。

```
sp_configure
```

例 2 “identity” という単語を含む設定パラメータをすべて表示します。

```
sp_configure "identity"
```

```
Configuration option is not unique.
```

Parameter Name	Default	Memory Used	Config Value	Run Value	Unit	Type
identity burning set	1	0	1	1	id	static
identity grab size	0	0	0	0	id	dyna
size of auto identit	10	0	10	10	bytes	dyna
...						

例 3 システムの *recovery interval in minutes* を 3 分に設定します。

```
sp_configure "recovery interval in minutes", 3
```

Parameter Name	Default	Memory Used	Config Value	Run Value	Unit	Type
recovery interval	5	0	3	3	min	dyn

設定オプションが変更されました。このオプションは動的であるため、SQL Server をリブートする必要はありません。

例 4 `number of devices` の値を Adaptive Server のデフォルトに再設定します。

```
sp_configure "number of device", 0, "default"
```

例 5 4 つのデータベースが同時にリカバリされるように設定するには、次のように入力します。

```
sp_configure "max concurrently recovered db", 4
```

例 6 4 つのチェックポイント・タスクを開始するには、次のように入力します。

```
sp_configure "number of checkpoint tasks", 4
```

例 7 クエリ処理測定基準 (qp 測定基準) をサーバ・レベルで取得します。

```
sp_configure "enable metrics capture", 1
```

例 8 `srv.config` ファイル内の値の検証を実行し、検証をパスしたパラメータをサーバに読み込みます。検証をパスしない値に対しては現在の実行値が使用されます。

```
sp_configure "configuration file", 0, "read",  
"srv.config"
```

例 9 `restore.config` ファイル内の値の検証を実行します。

```
sp_configure "configuration file", 0, "restore",  
"generic.config"
```

例 10 `my_server.config` ファイルを作成し、サーバが使用している現在の設定値をこのファイルに書き込みます。

```
sp_configure "configuration file", 0, "write",  
"my_server.config"
```

例 11 `$SYBASE/backup_config.cfg` 内の値の検証を実行します。

```
sp_configure "configuration file", 0, "verify",  
"backup_config.cfg"
```

使用法

- どのユーザも `sp_configure` を実行して、パラメータとその現在の値の情報を表示できます。ただし、パラメータの修正はできません。システム管理者は、`sp_configure` を実行すると、ほとんどの設定パラメータの値を変更できます。特定のパラメータを実行できるのは、システム・セキュリティ担当者だけです。このようなパーミッションについては、この項の後述の「パーミッション」で説明しています。

- `sp_configure` を使用すると、設定パラメータの値を単位指定子で指定できます。単位指定子には、ページ数を表す `p` または `P`、メガバイト数を表す `m` または `M`、ギガバイト数を表す `g` または `G`、テラバイト数を表す `t` または `T` があります。メモリの量を指定するパラメータを設定するときに単位を指定しなかった場合は、論理ページ・サイズが基本単位として使用されます。
- `sp_configure` で作成されたファイルは、制限されたパーミッションを持ちます。
- `sp_configure` を実行して動的パラメータを変更すると、次のようになります。
 - 設定値と実行値が更新されます。
 - 設定ファイルが更新されます。
 - 変更がすぐに有効になります。
- `sp_configure` を実行して静的パラメータを修正すると、次のようになります。
 - 設定値が更新されます。
 - 設定ファイルが更新されます。
 - 変更内容は、`Adaptive Server` を再起動しないと有効になりません。
- `sp_configure` は、パラメータなしで発行すると、グループ別のすべての設定パラメータ、その現在の値、デフォルト値、一番最後に設定された値 (該当するものがある場合)、その設定値が使用するメモリ量を表示します。
 - レポート内の `default` カラムには、`Adaptive Server` の出荷時の値が表示されます。パラメータを明示的に再設定しなければ、そのパラメータにはデフォルト値が使用されます。
 - `memory used` カラムにはそのパラメータが現在の値で使用しているメモリの量がキロバイト単位で表示されます。関連するパラメータどうしが、同じメモリ・プールのメモリを使用することがあります。たとえば、`stack size` と `stack guard size` が使用するメモリは、`number of user connections` が使用するメモリの一部として含まれています。これらの各パラメータが使用するメモリを別に加算すると、実際に使用するメモリ量よりも多くなります。`memory used` カラムでは、他のパラメータとメモリを共有しているパラメータにシャープ記号 (#) が付きます。

- `config_value` カラムには、`sp_configure` によって設定された設定パラメータの最新の値が表示されます。
- `run_value` カラムには、Adaptive Server が使用している値が表示されます。パラメータの値を `sp_configure` で修正すると、この値は変更されます。静的パラメータの場合は、Adaptive Server を再起動した後にこの値が変更されます。この値は、`syscurconfigs.value` に保管されています。

注意 サーバが大文字と小文字を区別しないソート順を使用している場合、パラメータの付かない `sp_configure` は、アルファベット順にすべての設定パラメータとグループのリストを返しますが、グループ化表示はされません。

- それぞれの設定パラメータは、関連する表示レベルを持っています。表示レベルは、次の 3 とおりです。
 - **basic** (基本) レベル – 最も基本的なパラメータだけを表示します。このレベルは、一般的なサーバ・チューニングに適しています。
 - **intermediate** (中間) レベル – **basic** レベルのパラメータの他に、より複雑な設定パラメータも表示します。このレベルは、あまり複雑でないサーバ・チューニングに適しています。
 - **comprehensive** (包括) レベル – デフォルトの表示レベルです。最も複雑なものまでを含むすべてのパラメータが表示されます。このレベルは非常に細かいサーバ・チューニングに適しています。

他の表示レベルを設定すると、`sp_configure` で表示される情報量を少なくすることができ、設定パラメータのサブセットを使って処理ができるようになります。

現在の表示レベルを表示する構文は次のとおりです。

```
sp_displaylevel
```

- `sp_configure` は、オープン・トランザクションが存在しない場合に連鎖トランザクション・モードを使用するセッションで実行できます。
- 個々の設定パラメータについては、『システム管理ガイド』を参照してください。

クラスタに対する設定パラメータの設定

- 設定オプションもインスタンス名も指定されていない場合、表示される情報は `system_view` の設定に依存します。
- 設定オプションが指定されずインスタンス名が指定されていると、**Adaptive Server** は指定されたインスタンスのインスタンス固有設定すべてを表示します。
- 設定オプションは指定されているが、設定値とインスタンス名が指定されていない場合、**Adaptive Server** はその設定オプションの現在の設定値を、そのクラスタのビュー内にあるすべてのインスタンスについて表示します。インスタンス名が指定されると、**Adaptive Server** はそのインスタンスの設定情報を表示します。
- 設定オプションと値が指定され、インスタンスの指定がない場合、**Adaptive Server** はその設定オプションの値をクラスタ全体に対して設定します。インスタンス名が指定されると、そのインスタンスのみにその設定値が設定されます。構文は次のとおりです。

```
sp_configure configuration_name, config_value, NULL,  
instance_name
```

- ローカル・テンポラリ・データベース内から設定オプションを設定することはできません。
- インスタンス固有の設定値が既に設定されている設定パラメータを、クラスタ全体の設定に設定し直すことができます。
- ユーザが設定し直すことができるのは自分が接続されているインスタンスのみです。

max concurrently recovered db

このパラメータは、データベースのリカバリ中の並列度を決定します。

- **Adaptive Server** がリカバリ中でない場合、この設定パラメータは静的に有効になります。**Adaptive Server** がリカバリ中の場合は、システム管理者が強制的に動的な逐次リカバリを実行できます。
- `max concurrently recovered db` の有効性は、データベースのレイアウトと、基本となる I/O サブシステムのパフォーマンスに依存します。

number of checkpoint tasks

このパラメータは、並列チェックポイントを設定します。

- 並列チェックポイントは、データベースのレイアウトと、基本となる I/O サブシステムのパフォーマンスに依存します。このパラメータのチューニングは、アクティブなデータベースの数と、I/O サブシステムの書き込み処理能力に基づいて行います。
- この設定パラメータは動的です。このパラメータの値が減るとチェックポイント・タスクが空になり、この値が増えると追加タスクが発生します。

パーミッション

sp_configure のパーミッション・チェックは、細密なパーミッションの設定によって異なります。すべてのユーザがパラメータの情報および現在の値を表示できます。

<p>細密なパーミッションが有効</p>	<p>細密なパーミッションが有効化されている場合は、次のようになる。</p> <ul style="list-style-type: none"> • sp_configure を実行して table <table number> 内のパラメータの値を変更できるのは、manage security configuration 権限を持つユーザだけである。 • sp_configure を実行して他の設定パラメータの値を変更するには、manage server configuration 権限が必要である。 		
<p>細密なパーミッションが無効</p>	<p>細密なパーミッションが無効化されている場合は、次のようになる。</p> <ul style="list-style-type: none"> • sp_configure を実行して table <table number> 内のパラメータの値を変更できるのは、sso_role を持つユーザだけである。 • sp_configure を実行して他の設定パラメータの値を変更するには、sa_role が必要である。 <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top;"> <ul style="list-style-type: none"> • allow procedure grouping • allow remote access • allow sendmsg • allow updates to system tables • audit queue size • auditing • automatic master key access • check password for digit • curread change w/ open cursors • current audit table • enable encrypted columns • enable granular permissions • enable ldap user auth • enable logins during recovery • enable pam user auth • enable predicated privileges • enable ssl </td> <td style="vertical-align: top;"> <ul style="list-style-type: none"> • FIPS login password encryption • log audit logon failure • log audit logon success • maximum failed logins • minimum password length • msg confidentiality reqd • msg integrity reqd • net password encryption reqd • restricted decrypt permission • secure default login • select on syscomments.text • SQL Perfmon Integration • suspend auditing when device full • syb_sendmsg port number • systemwide password expiration • unified login required • use security services </td> </tr> </table>	<ul style="list-style-type: none"> • allow procedure grouping • allow remote access • allow sendmsg • allow updates to system tables • audit queue size • auditing • automatic master key access • check password for digit • curread change w/ open cursors • current audit table • enable encrypted columns • enable granular permissions • enable ldap user auth • enable logins during recovery • enable pam user auth • enable predicated privileges • enable ssl 	<ul style="list-style-type: none"> • FIPS login password encryption • log audit logon failure • log audit logon success • maximum failed logins • minimum password length • msg confidentiality reqd • msg integrity reqd • net password encryption reqd • restricted decrypt permission • secure default login • select on syscomments.text • SQL Perfmon Integration • suspend auditing when device full • syb_sendmsg port number • systemwide password expiration • unified login required • use security services
<ul style="list-style-type: none"> • allow procedure grouping • allow remote access • allow sendmsg • allow updates to system tables • audit queue size • auditing • automatic master key access • check password for digit • curread change w/ open cursors • current audit table • enable encrypted columns • enable granular permissions • enable ldap user auth • enable logins during recovery • enable pam user auth • enable predicated privileges • enable ssl 	<ul style="list-style-type: none"> • FIPS login password encryption • log audit logon failure • log audit logon success • maximum failed logins • minimum password length • msg confidentiality reqd • msg integrity reqd • net password encryption reqd • restricted decrypt permission • secure default login • select on syscomments.text • SQL Perfmon Integration • suspend auditing when device full • syb_sendmsg port number • systemwide password expiration • unified login required • use security services 		

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名
73	オプションによって制御されるのではなく、自動的に監査されるイベント。	sp_configure を使用した auditing パラメータの設定	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – NULL • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名
74	オプションによって制御されるのではなく、自動的に監査されるイベント。	sp_configure を使用して auditing パラメータの設定をオフにする	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – NULL • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名
82	security	sp_configure	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – 設定パラメータの名前 • <i>Previous value</i> – コマンドが新しい値を設定している場合は、古いパラメータ値 • <i>Current value</i> – コマンドが新しい値を設定している場合は、新しいパラメータ値 • <i>Other information</i> – パラメータを設定する場合は、設定パラメータの番号。パラメータの設定に設定ファイルが使用されている場合は、設定ファイル名 • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

max concurrently recovered db と number of checkpoint tasks の詳細については、『システム管理ガイド』の「第 27 章 ユーザ・データベースのバックアップとリストア」を参照してください。

コマンド `set`

システム・プロシージャ `sp_dboption`、`sp_displaylevel`、`sp_helpconfig`、`sp_monitorconfig`、`set`

sp_copy_all_qplans

説明	ある抽象プラン・グループのすべてのプランを別のグループにコピーします。
構文	<code>sp_copy_all_qplans src_group, dest_group</code>
パラメータ	src_group ソース抽象プラン・グループの名前です。 dest_group プランのコピー先である抽象プラン・グループの名前を示します。
例	<code>dev_plans</code> グループ内の抽象プランすべてを <code>ap_stdin</code> グループにコピーします。 <pre>sp_copy_all_qplans dev_plans, ap_stdin</pre>
使用法	<ul style="list-style-type: none">コピー先グループを実際に設定してからプランをコピーしてください。コピー先グループにプランを保管できます。<code>sp_copy_all_qplans</code> は、コピー元グループのプランごとに <code>sp_copy_qplan</code> を呼び出します。各プランは、独立したトランザクションとしてコピーされるため、<code>sp_copy_all_qplans</code> の終了を妨げる問題が、すでにコピーされたプランに影響することはありません。<code>sp_copy_qplan</code> では、特定の抽象プランをコピーできないと、メッセージが出力されます。これらのメッセージは、<code>sp_copy_all_qplans</code> を実行する場合も表示されます。コピー先グループ内のプランのクエリ・テキストがコピー元グループ内のクエリ・テキストと正確に一致し、ユーザ ID が同じである場合、プランはコピーされません。この場合、プラン ID を示すメッセージがユーザに送られますが、コピー処理は、コピー元グループ内の次のプランへ継続されます。多数の抽象プランのコピーには長時間かかる場合があります。またデータベース内の <code>system</code> セグメントの領域と、データベースへの変更内容をログに記録する領域も必要になります。<code>sp_spaceused</code> を使用して、<code>sysqueryplans</code> のサイズを確認します。また、<code>system</code> と <code>logsegment</code> については <code>sp_helpsegment</code> を使用して、使用可能な領域を確認します。

パーミッション	sp_copy_all_qplans のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、manage abstract plans 権限を持つユーザであることが必要。 どのユーザも sp_copy_all_qplans を実行して各自の抽象プランをコピーできる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か sa_role が付与されたユーザであることが必要。 どのユーザも sp_copy_all_qplans を実行して各自の抽象プランをコピーできる。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_copy_qplan](#)、[sp_help_qpgroup](#)

sp_copy_qplan

説明 抽象プランを抽象プラン・グループにコピーします。

構文 `sp_copy_qplan src_id, dest_group`

パラメータ `src_id`
コピーする抽象プランの ID を示します。

`dest_group`
コピー先の抽象プラン・グループの名前です。

例 `sp_copy_qplan 2140534659, ap_stdin`

- 使用法**
- コピー先グループを実際に設定してから、抽象プランをコピーしてください。プランはプラン ID によってユニークに識別されます。したがって、コピー元グループを指定する必要はありません。
 - 新しいプラン ID は、プランのコピー時に生成されます。システム管理者またはデータベース所有者がプランをコピーする場合でも、プランの作成者であるユーザの ID は保存されます。別のユーザ ID を割り当てるために、システム管理者またはデータベース所有者は、`sp_export_qpgroup` と `sp_import_qpgroup` を使用できます。
 - コピー先グループ内のプランのクエリ・テキストが、コピー元グループ内のクエリ・テキストおよびユーザ ID と正確に一致する場合、そのプランはコピーされません。また、プラン ID を示すメッセージがユーザに送られます。
 - 抽象プラン・グループにあるプランをすべてコピーするには、`sp_copy_all_qplans` を使用します。

パーミッション `sp_copy_qplans` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザであることが必要。 どのユーザも <code>sp_copy_qplan</code> を実行して各自のプランをコピーできる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。 どのユーザも <code>sp_copy_qplan</code> を実行して各自のプランをコピーできる。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_copy_all_qplans](#)、[sp_help_qpgroup](#)、[sp_help_qplan](#)、[sp_import_qpgroup](#)

sp_countmetadata

説明	Adaptive Server にあるインデックス、オブジェクト、またはデータベースの数を表示します。
構文	<code>sp_countmetadata "configname" [, dbname]</code>
パラメータ	<p><i>configname</i></p> <p>“number of open databases”、“number of open objects”、“number of open indexes”、“number of open partitions” のいずれかです。</p> <p><i>dbname</i></p> <p><code>sp_countmetadata</code> はこのデータベースに対して実行されます。データベース名が指定されていない場合は、<code>sp_countmetadata</code> はすべてのデータベースでの合計数を提供します。</p>
例	<p>例 1 Adaptive Server 内のユーザ・オブジェクト数をレポートします。この値を使用して、データベース内に許容オブジェクト数を設定し、さらに追加オブジェクトとテンポラリ・テーブルの領域を設定します。</p> <pre>sp_configure "number of open objects", 310 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.</pre> <p>例 2 Adaptive Server 内のインデックス数をレポートします。</p> <pre>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.</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_countmetadata</code> は、Adaptive Server 内にあるインデックス、オブジェクト、データベース、またはパーティションの数を表示します。<code>model</code> や <code>tempdb</code> などのシステム・データベースの数も含まれます。 • Adaptive Server のピーク時に <code>sp_countmetadata</code> を実行しないでください。ピーク時に実行すると、<code>sysindexes</code>、<code>sysobjects</code>、<code>sysdatabases</code>、<code>syspartitions</code> システム・テーブルで競合が発生することがあります。

- 特定のデータベースの情報が必要な場合は、指定したデータベース上で `sp_countmetadata` を実行できます。ただし、インデックス、オブジェクト、データベース、またはパーティション用のキャッシュを設定するときは、`database_name` オプションを使用しないで `sp_countmetadata` を実行します。
- `sp_countmetadata` が返すメモリの情報は、プラットフォームによって異なります。たとえば、Windows NT 版の Adaptive Server 上のデータベースは、Sun Solaris 上の同データベースとは異なる `sp_countmetadata` 結果を得ることになります。ただし、ユーザ・インデックス数、オブジェクト数、データベース数、パーティション数の情報には一貫性があります。
- `sp_countmetadata` は、テンポラリ・テーブルを計算しません。テンポラリ・テーブルを確保するために、`open objects` 値に 5%、`open indexes`、`open partitions` 値に 10% を追加します。
- “open indexes”、“open objects”、“open databases”、または “open partitions” の間で一意にならない文字列を `configname` に指定すると、`sp_countmetadata` は、一致する設定パラメータ名を、設定されている値と現在の値とともに返します。次に例を示します。

```
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
```

パーミッション

`sp_countmetadata` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_configure](#)、[sp_helpconfig](#)、[sp_monitorconfig](#)

sp_cursorinfo

説明 特定のカーソルの情報、または使用しているセッションでアクティブなすべての実行カーソルの情報をレポートします。

構文 sp_cursorinfo [{*cursor_level* | null}] [, *cursor_name*]

パラメータ *cursor_level* | null

Adaptive Server がカーソルの情報を返すレベルです。 *cursor_level* には、次のレベルを指定できます。

レベル	カーソルのタイプ
N	特定のプロシージャ・ネスト・レベルにおいて、ストアド・プロシージャ内部で宣言された任意のカーソル。このレベルには、任意の正の数を指定できる。
0	ストアド・プロシージャの外部で宣言されたカーソル。
-1	上記のいずれかのカーソル。このレベルには、任意の負の数を指定できる。

カーソル・レベルに関係なく、特定の *cursor_name* のカーソルについての情報を表示する場合は、このパラメータに null を指定してください。

cursor_name

カーソルの固有名です。 Adaptive Server は、指定した *cursor_level* で、この名前を使用するすべてのアクティブなカーソルの情報をレポートします。このパラメータを省略すると、Adaptive Server は、そのレベルのすべてのカーソル情報をレポートします。

例 **例 1** レベル 0 の c という名前のカーソルに関する情報を表示します。

```
1> declare c cursor
2> for select au_id, au_lname, au_fname from authors
3> go
1> sp_cursorinfo
2> go
```

```
Cursor name 'c' is declared at nesting level '0'.
The cursor is declared as NON-SCROLLABLE cursor.
The cursor id is 917505.
The cursor has been successfully opened 0 times.
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 updatable.
This cursor is using 5389 bytes of memory.
```

```
(return status = 0)
```

例 2 セミ・センシティブなスクロール可能カーソル `css` について、カーソルがスクロール可能かどうかと、センシティビティについての情報を表示します。

```
sp_cursorinfo 0, cursor_css
```

```
-----
```

```
Cursor name 'css' is declared at nesting level '0'.
The cursor is declared as SEMI_SENSITIVE SCROLLABLE cursor.
The cursor id is 786434.
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 on a row.
There have been 1 rows read, 0 rows updated and 0 rows deleted through this
cursor.
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.
This cursor is using 19892 bytes of memory.
There are 2 columns returned by this cursor.
The result columns are:
Name = 'c1', Table = 't1', Type = INT, Length = 4 (not updatable)
Name = 'c2', Table = 't1', Type = INT, Length = 4 (not updatable)
```

使用法

- `cursor_level` も `cursor_name` も指定しなければ、Adaptive Server はすべてのアクティブ・カーソルの情報を表示します。アクティブなカーソルとは、ユーザが宣言し、Adaptive Server が割り付けたカーソルのことです。
- Adaptive Server は、個々のカーソルについて、次の情報をレポートします。
 - カーソル名、そのネスト・レベル、そのカーソル ID、そのプロシージャ名 (ストアド・プロシージャで宣言されている場合)。
 - カーソルがオープンされた回数。
 - コンパイルされた独立性レベル (0、1、または 3) と、(オープンされている場合は) その時点でスキャンしている独立性レベル (0、1、または 3)。

- カーソルがオープンまたはクローズされているかどうか。カーソルがオープンされている場合は、現在のカーソル位置とフェッチされたローの数を示す。
- カーソルの現在の位置が削除されたときに、オープンしているカーソルがクローズされるかどうか。
- カーソルの現在のトランザクションがコミットされるか、またはロールバックされたときに、カーソルがオープン状態を続けるかまたはクローズされるかどうか。
- そのカーソルのフェッチごとに返されたローの数。
- カーソルが更新可能または読み込み専用かどうか。
- カーソルによって返されたカラムの数。各カラムに対しては、カラム名、テーブル名、式の結果を表示。また、カラムが更新可能かどうかを表示。

sp_cursorinfo からの出力は、カーソルのステータスによって異なります。上記の情報の他に、sp_cursorinfo によって、カーソルの showplan 出力が表示できます。showplan の詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。

パーミッション

すべてのユーザが sp_cursorinfo を実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照

コマンド `declare cursor`、`set`

sp_dbextend

説明

以下のことができます。

- データベース / セグメントのペアおよびデバイスに、自動データベース拡張プロシージャをインストールする。
- 個々のセグメントやデバイスにサイト固有のポリシーを定義する。
- 大量のロードを処理する前にそのオペレーションを確認するため、データベース拡張構成の実行をシミュレートする。

これらのポリシーは、master データベースの `sysattributes` テーブルに格納されます。

引数はすべて、次の文字列引数です。

構文

```
sp_dbextend 'help' [, command]

sp_dbextend [ ['set', ['threshold', dbname, segmentname, freespace |
               'database', dbname, segmentname { [ [, growby] [, maxsize] } ] |
               'device', devicename { [ [, growby] [, maxsize] } ] } |
               'clear', 'threshold', dbname, segmentname

sp_dbextend 'clear', 'database' [, dbname [, segmentname ] ]
sp_dbextend 'clear', 'device' [, devicename ]
sp_dbextend 'modify', 'database', dbname, segmentname,
             { 'growby' | 'maxsize' }, newvalue
sp_dbextend 'modify', 'device', devicename, { 'growby' | 'maxsize' },
             newvalue
sp_dbextend { 'list' | 'listfull' } [, 'database' [, dbname [, segmentname
             [, order_by_clause] ] ] ]
sp_dbextend { 'list' | 'listfull' } [, 'device' [, devicename [, order_by_clause] ] ] ]
sp_dbextend { 'list' | 'listfull' } [, 'threshold' [, @dbname
             [, @segmentname] ] ]
sp_dbextend 'check', 'database' [, dbname [, segmentname ] ]
sp_dbextend { 'simulate' | 'execute' }, dbname, segmentname [, iterations ]
sp_dbextend 'trace', { 'on' | 'off' }
sp_dbextend 'reload [defaults]'
sp_dbextend { 'enable' | 'disable' }, 'database' [, dbname [, segmentname ] ]
sp_dbextend 'who' [, 'spid' | 'block' | 'all' ]
```

パラメータ

set

データベース、セグメント、またはデバイスが起動されるスレッシュホールドを設定します。引数は次のとおりです。

- **threshold** — 指定されたデータベースとセグメントにスレッシュホールドをインストールする場合の空き領域レベルを指定します。

freespace は、メガバイトなどのサイズ単位指定子で必ず指定してください。サイズ単位を指定しない場合、*freespace* の値は、セグメント内のキロバイト数として扱われます。

- **database** — データベースとセグメントのペアの名前、データベースを変更するサイズ、および拡張プロセスを停止させる、データベースの最大サイズを指定します。

growby — データベースが拡張試行のたびに拡大する速さを単位指定子またはパーセンテージの値で表したものです。*maxsize* は、それ以上拡張が発生しなくなったときのセグメントの最大サイズです。どちらのパラメータもオプションです。

- **device** — デバイスが拡大可能であるときの、デバイスの拡大する速さと最大サイズを単位指定子またはパーセンテージの値で指定します。デバイスの *maxsize* は、OS のディスク制限の影響を受けます。

clear

指定されたデータベースとセグメント、または指定されたデバイスに対して以前に設定された拡張ルールをすべてクリアします。

modify

データベースとセグメントに対して以前に設定されたサイト固有のポリシー (*growby* や *maxsize* など) を修正します。

newvalue を使用して、自動拡張について設定する新しい値を指定します。

list

指定されたデータベース、セグメント、デバイス、または指定されたセグメントのスレッシュホールドについて、既存ルール of 簡単なリストを表示し、`master.dbo.sysattributes` からのデータを、人間が判読できるフォーマットで表示します。これによって、データベースごと、あるいはデバイスごとにルールを表示できます。

現在、有効なルールを表示します。

`order_by_clause` を使用すると、デフォルトの順序 (名前、タイプ) と異なる順序のリストを生成できます。

`threshold` を使用すると、指定のデータベース (`@dbname` を使用) とセグメント (`@segment` 名を使用) に現在インストールされているすべてのスレッシュホールドが表示されます。

listfull

サイト固有ポリシー・ルールを完全にリストし、`sysattributes` テーブルに `comment` カラムを含めます。このカラムには、ルールの設定時と最終修正時の `datetime` スタンプが表示されます。

check

現在のポリシーを検査し、各セグメントの現在の領域レイアウトと整合性があることを確認します。冗長、無効、あるいは不正確なポリシー設定が見つかった場合は、警告メッセージが表示されます。

simulate

`set` コマンドで実装されたポリシーのセットに従い、実際に実行される、データベースまたはデバイスの拡張動作をシミュレートします。

`iterations` は、拡張をシミュレートする回数を指定します。

execute

現在のポリシーのセットを使用して、データベースとセグメント、またはデバイスの実際の拡張を実行します。

reload defaults

システムから提供される `growby` と `maxsize` のデフォルトを使用して、すべてのデータベース、セグメント、デバイスで `sysattributes` を再初期化し、データベースまたはデバイスの動作を元のデフォルトに戻します。

help

`set` または `list` など、すべてのコマンド・パラメータのヘルプ情報を提供するか、すべての単独コマンドのヘルプ情報を提供します。

trace

すべての拡張プロセスのスレッシュホールド・プロシージャ実行ロジックをトレースします。

enable と disable

指定されたデータベース・セグメントまたはデバイスに対する自動拡張プロシージャを有効または無効にします。

who

現在実行されているアクティブな拡張プロセスをすべて表示します。'<spid>' を指定すると、特定の **spid** だけを出力できます。次の値を使用します。

- *block* — 現在、拡張プロセスのブロッキングの原因になっているタスクを表示します。
- *all* — 現在アクティブなタスクをすべて表示します。

freespace

指定されたセグメントにスレッシュホールド・プロシージャがインストールされる場合の空き領域の値を指定します。*freespace* を指定する場合は、必ず、メガバイトなどのサイズ単位指定子を使用してください。

dbname

スレッシュホールドがインストールされるデータベースの名前です。

segmentname

データベース *dbname* に含まれるセグメントです。

devicename

影響を受けるデバイスの論理名です。

newvalue

データベースとセグメントのペアまたはデバイスのポリシーを修正するときに自動拡張用に設定する新しい値を指定します。

order_by_clause

list コマンドで、デフォルトの順序と異なる順序のリストを生成します。デフォルトの順序は、名前、タイプです。

iterations

拡張がシミュレートまたは実行される回数を指定します。

growby

スレッシュホールド・プロシージャが試行されるたびに、指定されたデータベース・セグメントまたはデバイスが拡張されるサイズを単位指定子またはパーセンテージの値で指定します。

maxsize

セグメントとデータベースのペアまたはデバイスの最大サイズです。このサイズで自動拡張が停止します。

例 **例 1 set thresholds** — データベース **pubs2** のログ・セグメントに領域拡張スレッシュホールドを空き領域 **100MB** でインストールします。

```
sp_dbextend 'set', 'thresh', pubs2, logsegment, '100m'
```

例 2 set database — **logsegment** セグメントに対するポリシーを、1 回の拡張試行での拡張サイズを **100MB** としてインストールします。

```
sp_dbextend 'set', 'database', pubs2, logsegment, '100m'
```

例 3 set device — OS のディスク領域制限またはデバイス・サイズ **32GB** に達するまで、このデバイスを拡張します。

```
sp_dbextend 'set', 'device', pubs2-datadev1, '100m'
```

例 4 clear — **pubs2**、**logsegment** に以前にインストールされた領域拡張スレッシュホールドをすべてクリアする方法を示します。

```
sp_dbextend 'clear', 'thresh', pubs2, logsegment
```

空き領域 **200MB** でインストールされた、**pubs2** のセグメント **dataseg1** に対する領域拡張スレッシュホールドをクリアすることもできます。

```
sp_dbextend 'clear', 'thresh', pubs2, dataseg1, '200m'
```

例 5 modify — 1 回の拡張試行での増加率を現在の値の **5%** にします。

```
sp_dbextend 'modify', 'da', pubs2, logsegment, 'growby', '5%'
```

maxsize があらかじめ指定されていないと、コマンドが失敗するおそれがあります。

```
sp_dbextend 'modify', 'device', pubs2_log_dev, 'maxsize', '2.3g'
```

例 6 list — すべてのデータベースとデバイスのルールを簡単にリストします。

```
sp_dbextend 'list'
```

次のコマンドは、**pubs%** 不要です。と類似する名前のすべてのデータベースのルールをリストします。

```
sp_dbextend 'list', 'database', 'pubs%'
```

例 7 listfull — すべてのデータベースとデバイスのルールをリストします。これには、**datetime** スタンプを表示する **comment** カラムが含まれます。

```
sp_dbextend 'listfull'
```


例 8 *list threshold* — pubs2 データベースからこれを発行すると、pubs2 データベースにあるすべてのセグメントのスレッシュホールド設定をリストします。

```
sp_dbextend 'list', 'threshold'
```

特定のセグメントのスレッシュホールドを確認する場合は、次のように使用します。

```
sp_dbextend 'list', 'threshold', pubs2, 'logsegment'
```

例 9 *simulate* — スレッシュホールドを超えない拡張を 2 回シミュレートします。

```
sp_dbextend 'simulate', pubs2, logsegment, '2'
```

例 10 *execute* — 自動拡張プロシージャを実行します。

```
sp_dbextend 'execute', pubs2, logsegment
```

例 11 *help* — 特定コマンドのヘルプを取得します。

```
sp_dbextend help, 'set'
```

使用法

- 自動拡張スレッシュホールドは、任意のデータベースとセグメントのペアに対して 1 つだけ設定できます。スレッシュホールド・プロシージャのインスタンスをもう 1 つインストールしようとする、異なる空き領域値であっても、エラーが発生します。
- システムから提供されるデフォルトは設定できません。修正は可能です。修正したシステム・デフォルトをリセットするには、*installdbextend* スクリプトを再実行するか、*reload defaults* コマンドを使用します。
- 特定セグメントの自動拡大を禁止するには、*growby* または *maxsize* に 0 を指定するか、スレッシュホールド・プロシージャをまったくインストールしないようにします。このパラメータに NULL を指定すると、システムから指定されるデフォルトの *growby* の増加率がデフォルトとして使用されます。
- デフォルトでは、デバイスのサイズが 40MB を超えると、データベースのサイズが 10 パーセント増加します。40MB より小さいデータベースのサイズは 4MB ずつ増加します。サイトのニーズに応じて、データベースのサイズ変更の限界を指定できます。
- *maxsize* は、自動拡張プロセスが停止するときのセグメントの最大サイズであり、データベースの最大サイズではありません。

- **default** データベースの最大サイズは、システムからは指定されません。*maxsize* の値が指定されない場合、データベースのサイズは、データベース・デバイスの物理的制限によってのみ制限されます。
- 特定のデバイスに対して自動拡大機能をオフにするには、*growby* または *maxsize* に 0 を指定します。*growby* の値を指定しない場合は、デフォルトの増加率が使用されます。
- *maxsize* には、デバイス上で使用可能なディスク領域の総量より大きな値を設定することができますが、実際の拡張は、拡張が実行される時点で使用可能なディスク領域に制限されます。
- このストアド・プロシージャを使用してスレッシュホールドをクリアする場合は、*dbname* 引数と *segmentname* 引数が必要です。
- データベースをクリアするためにこのストアド・プロシージャを使用し、*dbname* と *segmentname* を指定しない場合は、現在のデータベースとそこにあるすべてのセグメントに対するすべてのポリシー・ルール (*master.dbo.sysattributes* 内で関連するすべてのロー) が削除されます。これは、すべての設定をデフォルトに戻して再起動するのに良い方法です。
- このストアド・プロシージャを使用してデバイスをクリアする場合、*devicename* の値を指定しないと、ポリシー・ルールがクリアされません。単一デバイスのポリシー・ルールをクリアするには、*devicename* を指定し、すべてのデバイスのポリシーをクリアするには、“%” を使用します。
- パターンを使用して *dbname*、*devicename*、*segmentname* を指定すると、指定したパターンと一致するパターンを持つ名前が、*clear*、*enable*、*disable*、*list* などのオペレーションの対象になります。
- 値やプロパティを修正するためには、まずそれらを *set* (設定) します。値があらかじめ設定されていないと、*modify* は失敗します。*growby* と *maxsize* は、*newvalue* で指定された新しい値に修正されます。
- *newvalue* で指定された新しい値は、データベースまたはデバイスを拡張する以降の実行でも引き続き有効です。*newvalue* がデータベース、セグメント、またはデバイスの現在のサイズより小さい場合、そのオブジェクトは縮小されません。*newvalue* は、その後の拡張だけに有効で、現在のサイズには影響しません。
- データベースに対して *list* を使用し、*dbname* または *segmentname* を指定しない場合は、現在のデータベースにあるすべてのセグメントのすべてのポリシー・ルール (*master.dbo.sysattributes* のロー) がリストされます。

- 個々のデータベースとそこにあるセグメントのポリシー・ルールを取得するには、*dbname* と *segmentname* を指定します。
- デバイス名に対して *list* を使用し、*devicename* を指定しない場合は、すべてのデバイスのデフォルトのポリシー・ルールがリストされます。これをフィルタして単一デバイスのポリシー・ルールをリストするには、*devicename* を指定するか、*devicename* のパターン指定子を使用します。
- 拡張を一度にシミュレートできるデータベースとセグメントのペアは 1 つだけです。*dbname* 引数と *segmentname* 引数の両方が必要です。*execute* コマンドや *simulate* コマンドの *dbname* または *segmentname* にはワイルドカード・パターンを使用できません。
- デバイスの最大サイズは 4 TB です。
- *modify* や *simulate* を使用した後にデータベースやデバイスを再初期化するには、*reload* を使用します。*reload* は、システム・デフォルト動作を表す、*master.dbo.sysattributes* の既存ローをすべて削除し、新しいローをロードします。
- *reload* は、ユーザが指定したポリシーを削除しません。
- *trace* は、サーバ全体でのトレース機能をオンまたはオフにします。*trace* がオンの場合は、スレッシュホールド動作が起動されたときにサーバ・エラー・ログにメッセージが追加されます。*trace* は、トラブルシューティング目的でのみ使用します。

パーミッション

sa_role 権限のないデータベース所有者が自動拡張プロシージャをセグメントにインストールした場合、そのデバイスは拡張されません。これは、ユーザが *disk resize* コマンドを実行できないためです。スレッシュホールド・プロシージャをインストールするための *set threshold* コマンドは、*sa_role* 権限を持つユーザが実行することをおすすめします。

sp_dbextend の次のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効

細密なパーミッションが有効化されている場合は、次のようになります。

- sp_dbextend clear database — own any database 権限を持つユーザであることが必要。または、指定したデータベースのデータベース所有者か、そのデータベースに対して own database 権限を持つユーザであることが必要。
- sp_dbextend clear device — manage disk 権限を持つユーザであることが必要。
- sp_dbextend clear threshold — データベース所有者であるか、データベースに対して own database 権限を持つユーザであることが必要。
- sp_dbextend execute — データベース所有者か、指定したデータベースに対して own database 権限を持つユーザであること、および manage disk 権限が必要。
- sp_dbextend simulate — データベース所有者か own database 権限を持つユーザであることが必要。
- sp_dbextend enable/disable — own any database 権限を持つユーザまたはデータベース所有者であるか、指定したデータベースに対する own database 権限が必要。
- sp_dbextend list database — % パターンを指定する場合は、own any database 権限を持つユーザであることが必要。
- sp_dbextend list @ verbose=2 — own any database 権限を持つユーザであることが必要。
- sp_dbextend modify database — データベース所有者か、指定したデータベースに対して own database 権限を持つユーザであることが必要。または、sp_dbextend 'modify', 'database', 'default' の場合は own any database 権限を持つユーザであることが必要。
- sp_dbextend modify device — データベース所有者か manage disk 権限を持つユーザであることが必要。
- sp_dbextend reload defaults — own any database 権限を持つユーザであることが必要。
- sp_dbextend set database — データベース所有者か、指定したデータベースに対して own database 権限を持つユーザであることが必要。
- sp_dbextend set device — manage disk 権限を持つユーザであることが必要。
- sp_dbextend set threshold — データベース所有者か、指定したデータベースに対して own database 権限を持つユーザであること、および manage disk 権限が必要。
- sp_dbextend trace — set switch 権限を持つユーザであることが必要。

細密なパーミッションが無効

細密なパーミッションが無効化されている場合は、次のようになる。

- `sp_dbextend clear database` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend clear device` — `sa_role` を持つユーザーであることが必要。
- `sp_dbextend clear threshold` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend execute` — `sa_role` を持つユーザーであることが必要。
- `sp_dbextend simulate` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend enable/disable` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend list database` — % パターンを指定する場合は、`sa_role` パーミッションを持つユーザーであることが必要。
- `sp_dbextend list @ verbose=2` — `sa_role` を持つユーザーであることが必要。
- `sp_dbextend modify database` — データベース所有者か、`dbname` が `default` の場合は `sa_role` を持つユーザーであることが必要。
- `sp_dbextend modify device` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend reload defaults` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend set database` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend set device` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend set threshold` — データベース所有者か `sa_role` を持つユーザーであることが必要。
- `sp_dbextend trace` — `sa_role` を持つユーザーであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照

ストアド・プロシージャ [sp_droptreshold](#)、[sp_modifythreshold](#)

コマンド [alter database](#)、[create database](#)、[disk init](#)、[disk resize](#)

sp_dboption

説明	データベース・オプションを表示または変更し、非同期ログ・サービス機能を有効にします。
構文	<code>sp_dboption [dbname, optname, optvalue [, dockpt]]</code>
パラメータ	<p>dbname オプションが設定されるデータベースの名前です。パラメータを指定して <code>sp_dboption</code> を実行する (つまりデータベース・オプションを変更する) には、master を使用してください。ただし、master データベースのオプション設定は変更できません。</p> <p>optname 設定されるオプションの名前です。Adaptive Server は、オプション名に含まれるあらゆるユニークな文字列を認識します。オプション名がキーワードである場合や、埋め込みブランクまたは句読表記が含まれている場合は、オプションを引用符で囲んでください。</p> <p>複数のデータベース・オプションを同時にオンにできますが、ユーザ定義のトランザクション内でデータベース・オプションを変更することはできません。</p> <p>optvalue 設定値です。true はオプションをオンに、false はオフにします。</p> <p>dockpt <code>sp_dboption</code> が <code>checkpoint</code> コマンドを dbname に対して実行するかどうかを指定します。デフォルト値は 1 です。<code>checkpoint</code> を自動的に実行する設定です。<code>checkpoint</code> を手動で実行することによって dbname に対して <code>checkpoint</code> を実行できます。</p>
例	<p>例 1 データベース・オプションのリストを表示します。</p> <pre>sp_dboption Settable database options database_options ----- abort tran on log full allow nulls by default allow wide dol rows async log service auto identity dbo use only ddl in tran delayed commit deferred table allocation enforce dump tran sequence full logging for all</pre>

```
full logging for alter table
full logging for reorg rebuild
full logging for select into
identity in nonunique index
no chkpt on recovery
no free space acctg
read only
scratch database
select into/bulkcopy/pllsort
single user
trunc log on chkpt
trunc. log on chkpt.
unique auto_identity index
```

例 2 データベース pubs2 を読み取り専用にします。

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

```
Database option 'read only' turned ON for database 'pubs2'.
Running CHECKPOINT on database 'pubs2' for option 'read only' to take effect.
(return status = 0)
```

read という文字列は、使用可能なすべてのデータベース・オプションの中から read only オプションをユニークに識別します。キーワード read が、引用符で囲まれていることに注意してください。

例 3 データベース pubs2 を再度書き込み可能にしますが、*dockpt* オプションに 0 を指定しているため、「変更されたデータベースで CHECKPOINT コマンドを実行してください。」というメッセージが表示されます。

```
1> use pubs2
2> go
1> master..sp_dboption pubs2, "read", false, 0
2> go
```

```
Database option 'read only' turned OFF for database 'pubs2'.
Run the CHECKPOINT command in the database that was changed.
(return status = 0)
```

pubs2 に対してチェックポイントを手動で実行するには、次のように入力します。

```
1> checkpoint
2> go
```


例 4 pubs2 データベース内のテーブルでの `select into`、`bcp`、並列ソートの各オペレーションを可能にします。`select into` という文字列は、使用可能なすべてのデータベース・オプションの中から `select into/ bulkcopy` オプションをユニークに識別します。

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

注意 オプション名に埋め込みスペースがあるため、オプションを引用符で囲む必要があります。

例 5 mydb に作成された新しいテーブルに、10 桁の `IDENTITY` カラムが自動的に定義されます。`IDENTITY` カラムである `SYB_IDENTITY_COL` は、プライマリ・キーも、`unique` 制約も、`IDENTITY` カラムも指定しないで作成された、新しい各テーブル内で定義されます。

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

例 6 mydb で、`IDENTITY` カラムを持っているテーブルにインデックスを作成すると、インデックスのキーに `IDENTITY` カラムが自動的に含まれるようになります。テーブルに作成されたすべてのインデックスは、内部的にユニークになります。

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

例 7 pubs2 データベースの新しいテーブルに、ユニーク・ノンクラスタード・インデックス付きの `IDENTITY` カラムが自動的に設定されます。

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

例 8 指定されたデータベースに非同期ログ・サービス (ALS) を設定します。これによって、ユーザ・ログ・キャッシュとログ・ライター・スレッドが有効になります。

```
sp_dboption "mydb", "async log service", true
use mydb
```

例 9 指定されたデータベースで ALS を無効にします。

```
sp_dboption "mydb", "async log service", false
use mydb
```

例 10 big_db データベースに対して dump tran シーケンスを実施します。

```
sp_dboption 'big_db', 'enforce dump tran sequence',
true
```

例 11 mydb データベース内の select into コマンドと alter table コマンドの完全なロギングを有効にします。

create database コマンドは mydb を作成します。

```
create database mydb on datadev=20 log on logdev=10
go
CREATE DATABASE:allocating 10240 logical pages (20.0 megabytes) on disk
'datadev' (10240 logical pages requested).
CREATE DATABASE:allocating 5120 logical pages (10.0 megabytes) on disk
'logdev' (5120 logical pages requested).
Database 'mydb' is now online.
```

mydb 内の select into の完全ロギング・オプションをオンにします。

```
sp_dboption "mydb", "full logging for select into", "true"
go
Database option 'full logging for select into' turned ON for database
'mydb'.
Running CHECKPOINT on database 'mydb' for option 'full logging for select
into' to take effect.
(return status = 0)
```

mydb 内の alter table の完全ロギング・オプションをオンにします。

```
sp_dboption "mydb", "full logging for alter table", "true"
go
Database option 'full logging for alter table' turned ON for database
'mydb'.
Running CHECKPOINT on database 'mydb' for option 'full logging for alter
table' to take effect.
(return status = 0)
```

sp_helpdb を実行すると mydb の設定が表示されます。

```

sp_helpdb mydb
go
name db_size owner dbid created durability status
-----
mydb 30.0 MB sa 5 Dec 16, 2010 full full logging for select
into/alter table

(1 row affected)
device_fragments size usage created free kbytes
-----
datadev 20.0 MB data only Dec 16 2010 6:08PM 18696
logdev 10.0 MB log only Dec 16 2010 6:08PM not applicable

-----
log only free kbytes = 10184
(return status = 0)
1>

```

例 12 pubs2 データベースに対して遅延テーブルの作成を有効にします。

```
sp_dboption pubs2, 'deferred table allocation', true
```

使用法

- **master** データベース・オプションの設定内容は、変更できません。
- *optname* にあいまいな値を入力すると、エラー・メッセージが表示されます。たとえば、**dbo use only** と **read only** の 2 つのデータベース・オプションがありますが、*optname* パラメータに “only” と入力すると、両方のオプション名と一致するので、エラー・メッセージが表示されます。このエラー・メッセージには、入力された文字列と一致する部分を持つ完全名が出力されます。したがって、*optname* でユニークな値を指定するにはどのようにすればよいかわかります。
- データベース・オプションのリストを表示するには、**master** データベースの内部から **sp_dboption** をパラメータなしで実行します。
- 特定のデータベースに設定されているデータベース・オプションに関するレポートを表示するには、**sp_helpdb** を実行します。
- **no chkpt on recovery** オプションと **trunc log on chkpt** オプションの両方が同一データベースの **sp_dboption** で設定されている場合、前者が後者を無効にします。この競合は、**trunc log on chkpt** がデフォルトで **on** に設定されている **tempdb** データベースにおいて特に発生する可能性があります。
- データベース所有者またはシステム管理者は、**model** に対して **sp_dboption** を実行すると、すべての新しいデータベースに対する特定のデータベース・オプションを設定または設定解除できます。

- `sp_dboption` を実行した後、オプションが変更されたデータベース内で `checkpoint` コマンドが発行されるまで、変更内容は有効になりません。

完全なロギング

デフォルトでは、`select into`、特定のタイプの `alter table`、および `reorg rebuild` は、最低限のロギング・モードで実行されます。これらのコマンドを実行する前に、まず、`select into/bulk copy` データベース・オプションを `true` に設定して、Adaptive Server によってダンプ・シーケンスがブレイクされるようにします。つまり、`dump transaction` が使用されないようにするオペレーションを実行します。

“full logging for [select into | alter table | reorg rebuild | all]” オプションを使用する場合、コマンドは完全なロギングで実行されます。あらかじめ設定された `select into/bulk copy` の値は、いずれの完全なログを取るコマンドとも関係がなくなります。

高速 `bcp` と並列ソートの完全なロギングはサポートされていないため、`select into/bulk copy` が `true` に設定されていない場合、実行できません。

完全なロギングで実行されるようにオペレーションを設定すると、完全なログを取る他のオペレーションと同様に `dump transaction/load transaction` を使用してこれらのオペレーションをリカバリすることができます。

デフォルトで最低限のログを取るコマンドを完全なログを取るよう変更する構文は次のとおりです。

```
sp_dboption dbname, "full logging for  
[select into | alter table | reorg rebuild | all]",  
true | false
```

各パラメータの意味は次のとおりです。

- **full logging for select into** — `select into proxy table` の完全なログを取るには、実際のテーブルをホストするリモート・サーバ上で “full logging for select into” オプションを `true` に設定します。実際のテーブルをホストするサーバ上で `full logging for select into` オプションを `false` に設定すると、このコマンドは目的のデータベースで最低限のロギングを使用して実行され、ダンプ・トランザクション・シーケンスがブレイクします。
- **full logging for alter table** — データを移動する必要がある次の `alter table` の完全なロギングを有効にします。
 - `alter table add column not null`
 - `alter table drop column not null`
 - `alter table modify datatype of not null column`

- alter table partition

他の alter table コマンドは、常に完全なロギング・モードで実行されます。

注意 alter table lock によってロック・スキームを全ページロック・テーブルからデータページ・ロック/データロー・ロック・テーブル(またはその逆)に変更するにはデータ移動が必要ですが、この動作は full logging for alter table ではサポートされていません。

- full logging for reorg rebuild — テーブル・データを移動します。常に完全なログが取られる reorg rebuild index コマンドは影響を受けません。このパラメータによって、reorg rebuild table コマンドの完全なロギングが有効になります。このオプションを設定しない場合(このオプションを false に設定した場合)、Adaptive Server は最低限のロギングで reorg rebuild table コマンドを実行します。
- full logging for all — 上記のすべての完全なロギング・オプションを有効にします。all を false に設定すると、完全なロギング・オプションは無効になります。

注意 この構文では、完全なログを取る対象を指定する必要があります。“full logging”は単独では有効なオプションではありません。

どの full logging for オプションを使用した場合も、コマンドは完全なロギングで実行されます。完全なロギングを設定することにより、select into/bulk copy/plsort の設定とは無関係にコマンドが実行できます。但し、高速 bcp と並列ソートの完全なロギングはサポートされていないため、select into/bulk copy/plsort が true に設定されていない場合、実行できません。

完全なロギングで実行されるようにオペレーションを設定すると、完全なログを取る他のオペレーションと同様に dump transaction/load transaction を使用してこれらのオペレーションをリカバリすることができます。

dboption は “full logging for all” で、単なる “full logging” ではありません。

注意 完全なログを取る select into、alter table、または reorg rebuild コマンドを実行する場合、トランザクション・ログを格納するための大容量のログ領域が必要になる場合があります。

ログの縮小

完全なロギングが有効な場合、`select into`、`alter table`、および `reorg rebuild` を発行すると、特に大きなサイズのテーブルについて、ログ領域の需要が大幅に増加する可能性があります。そのため、ログ領域の拡張が必要になる場合があります。コマンドが完了した後、拡張したログ領域を削除するには、`alter database log off` コマンドを使用します。`alter database` および『システム管理ガイド 第1巻』の「ログ領域の縮小」を参照してください。

次のデータベースについては、`select into`、`alter database`、または `reorg rebuild` で完全なロギングを設定できません。

- master データベース
- インメモリ・データベース

次のデータベースへも設定は可能です。

- ログとデータのセグメントが混在したデータベース。ただし、データベースがログとデータのセグメントが混在しないように変更されるまで、オプションは無視されます。
- 持続性レベルが `full` でないデータベース。ただし、データベースの持続性が `full` に変更されるまで、オプションは無視されます。

このような制約は、`dump transaction` コマンドを実行できるデータベースがない場合に適用されます。`dump transaction` は、完全な形にリカバリできる DDL を使用すると、有効になります。

長いローの使用

長い可変長のデータオンリー・ロック (DOL) ローを許可するようにデータベースを設定するには、`allow wide dol rows` を使用します。

- `allow wide dol rows` はデータベースごとに個別に有効にする必要があります。
- `allow wide dol rows` データベース・オプションは、ユーザ・データベースでのみ設定できます。`master` データベースについては、`allow wide dol rows` データベース・オプションを設定できません。
- `allow wide dol rows` は、ページ・サイズが 8K 以下に設定された Adaptive Server で有効にしても効果はありません。

- `allow wide dol rows` を無効にすると、Adaptive Server ではワイド・データを含んでいる可変長の DOL ローは作成されませんが、このようなローを含むデータを選択することはできます。ただし、`allow wide dol rows` を有効にしないと、ワイド・データを含んでいるローを変更できません。変更するとワイド・データが含まれないローが生成されます。
- テンポラリ・データベースでは、`allow wide dol rows` 設定を有効にしないと、ワイド・データを含んでいる DOL ワークテーブルを使用できません。`tempdb` グループを使用する場合、`allow wide dol rows` をグループ内のすべてのデータベースについて有効にするか、または無効にします。この結果、ワークテーブルとクエリ処理動作は、特定のユーザ・セッションが関連付けられる `tempdb` に関係なく、グループ全体で一貫したものになります。

非同期ログ・サービス (ALS) オプション

`async log service (ALS)` は、ハイエンド対称マルチプロセッサ・システムのロギング・サブシステムに高いスループットを与えることで、Adaptive Server に大きなスケーラビリティの向上をもたらします。

- ALS オプションは、デフォルトでは無効です。
- `master` や `model` などのシステム・データベースでは、ALS オプションを有効にできません。
- ALS オプションは永続的です。指定したデータベースに対して ALS をいったん有効にした後は、ALS に無効にせずにデータベースをダンプしたり再ロードしたりできます。この機能を無効にするには、`sp_dboption` を使用してパラメータを `false` に設定します。

`enforce dump tran sequence` の考慮事項

`enforce dump tran sequence` は、後続のダンプ・トランザクションを妨害するオペレーションを防止します。

- `false` - (デフォルト) ダンプ・トランザクションを妨害するオペレーションを防止しません。
- `true` - ダンプ・トランザクションを妨害するオペレーションを禁止します。

このオプションは、データベースが次の場合にのみ `true` に設定できます。

- ログとデータのセグメントが混在していないデータベース。
- アーカイブ・データベースではない。
- ローカルまたはグローバルのテンポラリ・データベースではない。

- 読み取り専用ではない。
- スタンバイ・アクセス用にオンライン状態ではない。
- 持続性レベルが `full` である。持続性が `at_shutdown` および `no_recovery` のデータベースでは許可されていません。
- `select into/bulk copy/pllsort` または `trunc log on chkpt` が `false` に設定されている。これらのオプションは、いずれかが `true` に設定されている場合、自動的に `false` に再設定されます。
- `dump database` が不要な状態である。以下のオペレーションを実施した場合は、このデータベース・オプションを `true` に設定する前に、`ダンプ・データベース`を実行してください。
 - `select into`、`alter table modify`、`reorg rebuild`、`fast bcp`、`writetext` などの一部のログを取る更新が実行された。
 - トランザクション・ログがトランケートされた。
 - 新規に作成されたか、アップグレードされたデータベースである。

データベース・オプション `enforce dump tran sequence` が `true` の場合、次の内容は実行できません。

- `select into/bulk copy/pllsort` を `true` に設定する。部分ロギングを使用するコマンドは使用できません。
- `trunc log on chkpt` を `true` に設定する。ログは、チェックポイント・プロセスではトランケートされません。
- `dump tran with truncate_only` または `dump tran with no_log` を実行する。ログを、トランケートするには、アーカイブ・デバイスにダンプする必要があります。
- データベースを読み取り専用にする。
- 持続性レベルを `full` から `at_shutdown` または `no_recovery` に変更する。
- `mixed-log-and-data` データベースに変更する。`load database` や `dbcc findstranded` などの場合、データベースはログとデータが混在するデータベースに変更される場合があります。

データベース・オプション

- **abort tran on log full** オプションは、指定されたデータベースのログ・セグメントでラストチャンス・スレッショルドが超過されたときに、実行中のトランザクションのそれ以降の処理を決定します。デフォルト値は **false** です。この場合、トランザクションは中断され、領域が解放されている場合にだけ再開されます。設定を **true** に変更すると、トランザクション・ログに書き込む必要があるすべてのユーザ・クエリは、ログ内の領域が解放されるまで強制終了されます。

- **allow nulls by default** オプションを **true** に設定すると、SQL 標準に準拠して、カラムのデフォルト値が **not null** から **null** に変更されます。Transact-SQL のカラム用デフォルト値は **not null** でなので、**null** が **create table** または **alter table** のカラム定義内で指定されていないかぎり、**null** 値は許容されません。**allow nulls by default true** はそれを逆にします。

allow nulls by default を使用しても、**select into** 文の実行時にカラムで **null** 値が可能かどうかを変更することはできません。代わりに、**convert** を使用して、結果のカラムで **null** 値が可能かどうかを指定します。

- **auto identity** オプションが **true** (オン) に設定されている間は、10 桁の **IDENTITY** カラムが、プライマリ・キー、一意性制約、または **IDENTITY** カラムのいずれも指定しないで作成されている新しいテーブル内で定義されます。**select *** 文ですべてのカラムを選択した場合でも、このカラムは表示されません。このカラムを検索するには、**select** リスト内で、カラム名 **SYB_IDENTITY_COL** を明示的に指定してください。

自動 **IDENTITY** カラムの精度を設定するには、**size of auto identity column** 設定パラメータを使用します。

auto identity データベース内で **true** を **tempdb** に設定することはできませんが、このカラムは認識されず、使用されることもありません。また、このデータベースに作成されるテンポラリー・テーブルに **IDENTITY** カラムが自動的に追加されることもありません。

IDENTITY カラムを持つ特定のテーブルのインデックスに関するレポートを得るには、**sp_helpindex** を実行します。

- **dbo use only** オプションが **true** (オン) に設定されている間、データベースを使用できるのはデータベース所有者だけです。

- `ddl in tran option` が `true` (オン) に設定されているときは、トランザクションで特定のデータ定義言語コマンドを使用できます。特定のデータベースで `ddl in tran` が `true` になっている場合、`create table`、`grant`、`alter table` などのコマンドは、そのデータベース内のトランザクションの中で使用できます。`model` データベースで `ddl in tran` が `true` になっている場合、これらのコマンドは、`ddl in tran` が `model` 内で設定された後で作成されたすべてのデータベースのトランザクション内で使用できます。

警告! データ定義言語コマンドは、`sysobjects` などのシステム・テーブル上でロックを保持します。トランザクション内部ではデータ定義言語コマンドを使用しないでください。どうしても使用しなければならぬ場合は、トランザクションを短くしてください。

トランザクション内の `tempdb` でデータ定義言語コマンドを使用すると、システムが停止することがあります。`tempdb` では、常に `ddl in tran` を `false` に設定しておいてください。

- `ddl in tran` オプションが `true` に設定されている場合は、ユーザ定義トランザクション内で次のコマンドを使用できます。

<code>alter table</code> — <code>partition</code> と <code>unpartition</code>	<code>drop default</code>
以外の句が使用可能です。	<code>drop index</code>
<code>create default</code>	<code>drop procedure</code>
<code>create index</code>	<code>drop rule</code>
<code>create procedure</code>	<code>drop table</code>
<code>create rule</code>	<code>drop trigger</code>
<code>create schema</code>	<code>drop view</code>
<code>create table</code>	<code>grant</code>
<code>create trigger</code>	<code>revoke</code>
<code>create view</code>	

- 次のコマンドは、どのような状況でも、ユーザ定義トランザクション内では使用できません。

<code>alter database</code>	<code>dump transaction</code>
<code>alter table...lock</code>	<code>drop database</code>
<code>alter table...partition</code>	<code>load transaction</code>
<code>alter table...unpartition</code>	<code>load database</code>
<code>create database</code>	<code>select into</code>
<code>disk init</code>	<code>truncate table</code>
<code>dump database</code>	<code>update statistics</code>

さらに、テンポラリ・テーブルを作成するシステム・プロシージャ、または **master** データベースを変更するシステム・プロシージャを、ユーザ定義トランザクション内で使用することはできません。

- **deferred table allocation** は、**model** データベースに対しては有効にできますが、その他のシステム・データベース (**master**、**sybsystemprocs**、**sybsystemdb** など) やテンポラリ・データベースに対しては一切有効にできません。
- **identity in nonunique index** オプションを指定すると、テーブルのインデックス・キーに **IDENTITY** カラムが自動的に組み込まれます。これによって、テーブルに作成されるインデックスはすべてユニークになります。このデータベース・オプションは、論理的にユニークでないインデックスを内部的にユニークにし、これらのインデックスを、更新可能なカーソルと独立性レベル 0 の読み込み処理に使用できるようにします。

identity in nonunique index オプションが正常に動作するには、テーブルがすでに **IDENTITY** カラムを持っていなければなりません。これには、**create table** 文を使用するか、またはテーブルを作成する前に **auto identity** データベース・オプションを **true** に設定します。

ユニークでないインデックスを持つテーブルでカーソルと独立性レベル 0 の読み込みを使用する場合は、**identity in nonunique index** を使用します。ユニーク・インデックスは、次にそのカーソルに **fetch** を実行したときに、カーソルが正しい位置に置かれるようにします。ユニークなインデックスと任意の独立性レベルを持つテーブル上でカーソルを使用する場合は、**unique auto_identity index** オプションを使用します。

identity in nonunique index オプションと **unique auto_identity index** とを混同しないでください。後者は、新しいテーブルにユニークなノンクラスタード・インデックス付きの **IDENTITY** カラムを追加するためのものです。

IDENTITY カラムを持つ特定のテーブルのインデックスに関するレポートを得るには、**sp_helpindex** を実行します。

- **no free space acctg** オプションは、データ・セグメントに対する空き領域の計算とスレッショルド・アクションの実行を抑制します。**no free space acctg** を **true** に設定すると、データ・セグメントの空き領域が再計算されなくなるため、リカバリ時間が短くなります。

- **no chkpt on recovery** オプションは、データベースの最新のコピーを保管する場合に、**true** (オン) に設定します。保管作業を行う環境では、プライマリ・データベースとセカンダリ・データベースが用意します。最初にプライマリ・データベースがダンプされ、セカンダリ・データベースにロードされます。次に、一定の間隔で、プライマリ・データベースのトランザクション・ログがダンプされ、セカンダリ・データベースにロードされます。

このオプションを **false** (オフ) に設定すると (デフォルト状態)、**Adaptive Server** の再起動時にデータベースがリカバリされてから、チェックポイントのレコードがデータベースに追加されます。このチェックポイントは、リカバリ・メカニズムが不必要に再実行されないようにするもので、シーケンス番号を変更して、プライマリ・データベースからの後続トランザクション・ログのロードを実行できないようにします。

セカンダリ・データベースについてこのオプションを **true** (オン) に設定すると、プライマリ・データベースからの後続トランザクション・ログのダンプがセカンダリ・データベースにロードされるように、セカンダリ・データベースがリカバリ処理からチェックポイントを取得しないようにします。

- **read only** オプションを選択すると、ユーザはデータベースからデータを取り出すことはできても、データを変更することはできません。
- トランザクションの完全なレコードをログ内に保管しないオペレーションを実行するには、**select into/bulkcopy/pilsort** を **on** に設定する必要があります。次のようなオペレーションが該当します。
 - **writetext** ユーティリティの使用。
 - 永久テーブルへの **select into** の実行。
 - **bcp** を使った高速「バルク・コピー」。デフォルトでは、インデックスを持たないテーブルに対しては高速 **bcp** が使用されます。
 - 並列ソートの実行。

トランザクション・ログのダンプでは、このような最小限のログしか取らないオペレーションはリカバリできないため、ダンプ・デバイスへの **dump transaction** は禁止されています。それでも、**dump transaction...with no_log** と **dump transaction...with truncate_only** の使用は可能です。ログを取らないオペレーションが完了した後、**select into/bulk copy/pilsort** を **false** (オフ) に設定して **dump database** を発行します。

ログを取らない変更が、`select into`、バルク・コピー、並列ソートによって加えられた後に `dump transaction` 文を発行すると、代わりに `dump database` を使用するように指示するエラー・メッセージが生成されます (`writetext` コマンドには、この保護はありません)。

`tempdb` がリカバリされることはありません。したがって、テンポラリー・テーブルに `select into` を実行するために `select into/bulkcopy/pllsort` オプションを `true` に設定する必要はありません。インデックスを持つテーブルに `bcp` を実行するために、このオプションを `true` に設定する必要はありません。インデックスを持つテーブルは常にバルク・コピーの低速バージョンでコピーされてログが記録されるからです。

`select into/bulkcopy/pllsort` オプションを設定しても、ログのダンプがブロックされることはありませんが、最低限のログを取る変更をデータに加えると、通常の `dump transaction` の使用はブロックされます。

デフォルトでは、新しく作成されるデータベースでは `select into/bulkcopy/pllsort` オプションは `off` に設定されます。このデフォルトを変更するには、`model` データベース内でこのオプションを `on` にします。

- `single user` を `true` に設定すると、一度に 1 人のユーザだけがデータベースにアクセスできます (シングル・ユーザ・モード)。

ユーザ・データベース内では、ストアド・プロシージャ内から `single user` を `true` に設定することはできません。ユーザがデータベースをオープンにしている間も設定できません。 `tempdb` 用に `single user` を `true` に設定することはできません。

- `trunc log on chkpt` オプションを選択すると、コミットされたトランザクションの 50 ロウを超えるトランザクション・ログがある場合、`checkpoint` の確認処理が実行されるたびに (通常は毎分 2 回以上)、トランザクション・ログがトランケートされる (コミットされたトランザクションが削除される) こととなります。ただし、データベース所有者が手動で `checkpoint` を実行する場合、ログはトランケートされません。開発作業中は、このオプションをオンにしてログが増えないようにしておくとう便利です。

`trunc log on chkpt` オプションがオンになっている間は、ダンプ・デバイスへのダンプ・トランザクション (`dump transaction`) は禁止されます。これは、トランケートされたトランザクション・ログからのダンプをメディア障害からのリカバリに使用できないためです。 `dump transaction` 文を発行すると、代わりに `dump database` を使用するように指示するエラー・メッセージが生成されます。

`trunc log on chkpt` は新しく作成されたデータベースではオフになっています。このデフォルトを変更するには、`model` データベース内でこのオプションを `on` にします。

警告！ `model` データベース内で `trunc log on chkpt` をオンにした場合に、新しく作成されたデータベースにデータベースとトランザクション・ログをまとめてロードしなければならない場合は、新しいデータベースでこのオプションを必ずオフにしてください。

- `delayed commit` オプションは、デフォルトでは無効です。これが有効になっているときには、すべてのローカル・トランザクションで遅延コミットが使用されます。つまり、コミットの時点で、ログ・ページの I/O の完了を待たずに、クライアントに制御が戻り、遅延コミット・トランザクションの最後のログ・バッファでは I/O は発行されません。データベースに対し、`delayed commit` と `ALS` オプションの両方が有効になっている場合には、遅延コミットは使用されません。
- `unique auto_identity index` オプションを `true` に設定すると、新しいテーブルに、ユニークなノンクラスタード・インデックスが付きの `IDENTITY` カラムが追加されます。デフォルトでは、`IDENTITY` カラムは 10 桁の数値データ型ですが、`size of auto identity column` 設定パラメータによってこのデフォルトを変更できます。`auto identity` と同様、すべてのカラムを `select *` 文によって選択した場合は、`IDENTITY` カラムを参照することはできません。このカラムを検索するには、`select` リスト内で、カラム名 `SYB_IDENTITY_COL` を明示的に指定してください。

カーソルや独立性レベル 0 読み込みをユニークでないインデックスに使用する必要がある場合は、`identity in nonunique index` オプションを使用します。

`tempdb` 内で `unique auto_identity index` を `true` に設定することはできませんが、このオプションは認識されず、使用されることもありません。また、このデータベースに作成されるテンポラリ・テーブルに、ユニークなインデックスを持つ `IDENTITY` カラムが自動的に追加されることはありません。

`unique auto_identity index` オプションを利用すると、ユニークなインデックスが付いた自動 `IDENTITY` カラムを持つテーブルを作成できるようになります。このユニークなインデックスは、更新可能カーソルとともに使用できます。テーブル上にユニークなインデックスがあれば、`fetch` の実行後にカーソルが確実に正しいローに位置付けられるようになります (独立性レベル 0 の読み込みを行うために、論理的にユニークでないインデックスを内部的にユニークにして更新可能カーソルを処理できるようにするには、`identity in nonunique index` オプションを使用します)。

場合によっては、`unique auto_identity index` オプションによってハロウィーン問題を回避できます。その理由は次のとおりです。

- ユーザは `IDENTITY` カラムを更新できません。したがって、`IDENTITY` カラムをカーソル更新時に使用することはできません。
- `IDENTITY` カラムは、ユニークなノンクラスタード・インデックス付きで自動的に作成されます。したがって、このカラムを更新可能なカーソルのスキャンに使用できます。

ハロウィーン問題、`IDENTITY` カラム、カーソルの詳細については、『`Transact-SQL ユーザーズ・ガイド`』の「第 18 章 カーソル：データのアクセス」と、『`パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン`』の「第 8 章 カーソルの最適化」を参照してください。

`unique auto_identity index` オプションと `identity in nonunique index` オプションとを混同しないでください。後者を使用するのは、テーブル内にあるすべてのインデックスをユニークにするために、そのテーブルのインデックス・キーに `IDENTITY` カラムを追加する場合です。

パーミッション

`sp_dboption` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データベース所有者であるか、データベースに対する <code>own database</code> 権限を持つユーザであることが必要。 データベース・オプションは、すべてのユーザが表示できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。 データベース・オプションは、すべてのユーザが表示できる。データベース所有者にエイリアスを割り当てているユーザは、 <code>sp_dboption</code> を実行してデータベース・オプションを変更することはできません。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル データベース・オプションの詳細については、『システム管理ガイド』を参照してください。

コマンド [checkpoint](#)、[select](#)

システム・プロシージャ [sp_configure](#)、[sp_helpdb](#)、[sp_helpindex](#)、[sp_helpjoints](#)

ユーティリティ [bcp](#)

sp_dbrecovery_order

説明	ユーザ・データベースがリカバリされる順序を指定し、データベースまたはすべてのデータベースのユーザが定義したリカバリ順をリストします。
構文	sp_dbrecovery_order [<i>database_name</i> [, <i>rec_order</i> [, <i>force</i> [, <i>relax</i> <i>strict</i>]]]]
パラメータ	<p><i>database_name</i> リカバリ順序を割り当てられているデータベース、またはユーザ定義のリカバリ順序がリストされるデータベースの名前です。</p> <p><i>rec_order</i> データベースがリカバリされる順序です。<i>rec_order</i> が -1 の場合は、ユーザ定義のリカバリ・シーケンスから、指定したデータベースが削除されます。</p> <p><i>force</i> リカバリ・シーケンスが終わった時点でなくても、ユーザはデータベースを現行のリカバリ・シーケンスに挿入できます。</p> <p><i>relax</i> リカバリが完了するとデータベースが作成されることを指定します(デフォルト)。 デフォルトは <i>relax</i> で、データベースはリカバリ完了直後にオンラインになります。</p> <p><i>strict</i> データベースが指定されたリカバリ順序でオンラインになることを指定します。</p>
例	<p>例 1 pubs2 データベースを、システム障害後にリカバリされる最初のユーザ・データベースになります。</p> <pre>sp_dbrecovery_order pubs2, 1</pre> <p>例 2 pubs3 データベースを、ユーザ定義のリカバリ・シーケンスの 3 番目に挿入します。当初その 3 番目の位置にあった別のデータベースは 4 番目に移動し、このデータベースの後に続くデータベースはすべて、1 つずつ順番に移動します。</p> <pre>sp_dbrecovery_order pubs3, 3, force</pre> <p>例 3 ユーザ定義のリカバリ・シーケンスから pubs2 データベースを削除します。削除後、ユーザ指定のリカバリ順序によってすべてのデータベースがリカバリされた後、pubs2 がリカバリされます。</p> <pre>sp_dbrecovery_order pubs2, -1</pre>

例 4 sp_dbrecovery_order を介して割り当てられたリカバリ順序で、すべてのデータベースの現在のリカバリ順序をリストします。

```
sp_dbrecovery_order
```

使用法

- sp_dbrecovery_order を使用してユーザ指定のリカバリ順序を入力または変更するには、master データベースをオープンしていなければなりません。任意のデータベースから、データベースのユーザ定義のリカバリ順序をリストできます。
- データベースのユーザ定義リカバリ位置を変更するには、sp_dbrecovery_order を使用してリカバリ・シーケンスからデータベースを削除し、sp_dbrecovery_order を使用して、そのデータベースを新しい位置に挿入します。
- システム・データベースは、常にユーザ・データベースより先にリカバリされます。システム・データベースとそのリカバリ順序は、次のとおりです。

```
master
model
tempdb
sybssystemdb
sybsecurity
sybssystemprocs
```

- sp_dbrecovery_order によるリカバリ順序をデータベースに割り当てなければ、システム・データベースの後に、すべてのユーザ・データベースが順番に (データベース ID 別に) リカバリされません。
- *database_name* が、
 - 指定されていても *rec_order* が特定されていない場合 — sp_dbrecovery_order は、指定されたデータベースのユーザ定義リカバリ位置を示します。
 - 指定されていない場合 — sp_dbrecovery_order は、ユーザ割り当てのリカバリ順序を使用して、すべてのデータベースのリカバリ順序をリストします。
- sp_dbrecovery_order によって割り当てられるリカバリ順序は、1 から始まり、値の間にギャップのない連続的な順序でなければなりません。リカバリ順序が割り当てられた最初のデータベースには、*rec_order 1* を割り当ててください。3 つのデータベースにリカバリ順序の 1、2、3 が割り当てられている場合、次のデータベースにリカバリ順序の 5 を割り当てることはできません。

パーミッション `sp_dbrecovery_order` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

sp_dbremap

説明	Adaptive Server に、 <code>alter database</code> が行った変更を認識させます。Adaptive Server のメッセージで指示された場合のみ、このプロシージャを実行してください。
構文	<code>sp_dbremap dbname</code>
パラメータ	<code>dbname</code> <code>alter database</code> コマンドが中断されたデータベースの名前です。
例	<code>alter database</code> コマンドはデータベース <code>sample_db</code> を変更しました。このコマンドによって、Adaptive Server から変更内容を参照できます。 <pre>sp_dbremap sample_db</pre>
使用法	<ul style="list-style-type: none"> • ダンプ処理中のデータベース上で発行された <code>alter database</code> 文に割り込みが発生すると、Adaptive Server は、<code>sp_dbremap</code> を実行するようにユーザに指示するメッセージを出力します。 • データベース・ダンプまたはトランザクション・ダンプ中に <code>sysusages</code> に加えられた変更内容は、ダンプが完了するまでアクティブなメモリにはコピーされません。これは、ダンプ中にデータベースのマッピングが変更されないようにするためです。<code>alter database</code> を実行すると、システム・テーブルの変更がただちにディスクに反映されます。メモリ内割り付けは、ダンプが完了するまで変更されません。したがって、<code>alter database</code> は一時停止します。 <code>sp_dbremap</code> を実行する場合、このコマンドはダンプ処理が完了するまで待機しなければなりません。 • <code>sp_dbremap</code> を実行する指示があってもかかわらず、このコマンドが実行されなかった場合、Adaptive Server は、次回再起動されるまで、<code>alter database</code> によって割り付けられている領域を使用できません。
パーミッション	<code>sp_dbremap</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、データベース所有者であるか、データベースに対する <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [alter database](#)、[dump database](#)、[dump transaction](#)

sp_defaultloc

説明 (コンポーネント統合サービスのみ) ローカル・データベース内での、オブジェクトのデフォルトの記憶位置を定義します。

構文 `sp_defaultloc dbname, defaultloc, defaulttype`

パラメータ

dbname

リモート記憶ロケーションにマップされるデータベースの名前です。 `create database` 文を使用して、データベースを先に定義しておいてください。システム・データベースはリモート・ロケーションにマップできません。

defaultloc

データベースがマップされるリモート記憶ロケーションです。既存のデフォルト・マッピングを削除するようにサーバに指示するには、このパラメータを `NULL` にします。 `defaultloc` 値は、次のようにピリオド(.)で終わります。

```
server.dbname.owner.
```

defaulttype

`object_loc` によって命名されたオブジェクトのフォーマットを指定する値の1つです。有効な値は次のとおりです。 `defaulttype` 値は引用符で囲んでください。

- `table` — `object_loc` によって命名されたオブジェクトがリモート・サーバからアクセスできるテーブルであることを示します。この値は `defaulttype` のデフォルトです。
- `view` — `object_loc` によって命名されたオブジェクトがリモート・サーバの管理するビューであり、テーブルとして処理されることを示します。
- `rpc` — `object_loc` によって命名されたオブジェクトは、リモート・サーバが管理する `RPC` であり、`RPC` からの結果セットを読み取り専用テーブルとして処理することを示します。

例 **例 1** `sp_defaultloc` は、リモートの記憶領域ロケーション `pubs.dbo` をリモート・サーバ `SYBASE` に定義します。これは、`pubs` データベースをリモート・ロケーションにマップします。 `create table book1` 文は、リモート・ロケーションに `book1` という名前のテーブルを作成します。 `bookN` 用の `create existing table` 文では、`pubs.dbo.bookN` がすでにリモート・ロケーションに存在していることが必須の条件です。また、`bookN` テーブルについての情報は、`bookN` ローカル・テーブル内に保管されます。

```
sp_defaultloc pubs, SYBASE.pubs.dbo., table
```

```
create table pubs.dbo.book1 (bridges char(15))
```

例 2 リモート・ロケーションへのデータベース `pubs` のマッピングを削除します。

```
sp_defaultloc pubs, NULL
```

例 3 リモート記憶ロケーションの `wallst.nasdaq.dbo` を識別します。ここで、“`wallst`” は `server_name` に提供された値であり、“`nasdaq`” は `database` に、“`dbo`” は `owner` に提供されます。RPC `sybase` は、リモート・ロケーションにすでに存在していなければなりません。`create existing table sybase` 文は、ローカル・テーブル `ticktape` 内に、RPC `sybase` からの結果セットについての情報を格納します。RPC `sybase` からの結果セットは、読み込み専用テーブルとみなされます。RPC については、挿入、更新、削除はサポートされていません。

```
sp_defaultloc ticktape, wallst.nasdaq.dbo., rpc
create existing table sybase (bestbuy integer)
```

使用法

- `sp_defaultloc` は、ローカル・データベース内にテーブルのデフォルト記憶ロケーションを定義します。データベース内のテーブル名をリモート・ロケーションにマップします。すべての `create table` および `create existing table` コマンドの前に `sp_addobjectdef` コマンドを発行するのではなく、ユーザがすべてのデータベースにデフォルトを確立することを許可します。
- `defaulttype` が `table`、`view`、`rpc` のいずれかの場合、`defaultloc` パラメータのフォーマットは次のようになります。

```
server_name.dbname.owner.
```

- `defaultloc` の指定はピリオド (.) で終了することに注意してください。
- `server_name` は、`sp_addserver` によって `sys.servers` に追加済みのサーバを表します。`server_name` パラメータは必須です。
- `dbname` は必須ではありません。サーバ・クラスによっては `dbname` をサポートしていないものもあります。
- `owner` を常に提供して、指定があいまいになることを避けてください。`owner` が提供されない場合、外部ログインがリモート・オブジェクト所有者に一致するかどうかによって、実際に参照されるリモート・オブジェクトが変わります。

- `create table` 文または `create existing table` 文を発行する前に、`sp_defaultloc` を発行してください。どちらかの文が使用されているとき、サーバは `sysattributes` テーブルを使って、作成または定義しようとしているオブジェクトに対し、テーブル・マッピングが指定されたかどうかを判断します。マッピングが指定されている場合、`create table` 文は、`object_loc` によって指定されたロケーションにテーブルが作成されるように指示します。`create existing table` 文は、既存のリモート・オブジェクトの情報をローカル・テーブルに格納します。
- `sp_defaultloc` を `defaulttype view` 付きで発行し、続いて `create table` を発行すると、コンポーネント統合サービスによってリモート・サーバ上に、ビューではなく新しいテーブルが作成されます。
- データベースのデフォルト・ロケーションを変更しても、以前に異なるデフォルト・ロケーションにマップされたテーブルに影響はありません。
- データベースのテーブルが作成されると、その後のすべての (`select`、`insert`、`delete`、`update` により行われる) `dbname` 内のテーブルへの参照は、正しいロケーションにマップされます。

パーミッション

`sp_defaultloc` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド `create existing table`、`create table`

システム・プロシージャ `sp_addobjectdef`、`sp_addserver`、`sp_helpserver`

sp_deletesmobj

説明 IBM Tivoli Storage Manager (TSM) から指定されたバックアップ・オブジェクトを削除します。

注意 sp_deletesmobj は、サイトで TSM がライセンスされている場合にのみサポートされます。

構文 sp_deletesmob "syb_tsm", "server_name"{, "database_name", "object_type", "dump_type", "until_time", "bs_name"}

パラメータ

syb_tsm

TSM との通信を有効にする libsyb_tsm.so モジュールを起動するキーワードです。

server_name

削除対象の TSM のバックアップ・オブジェクトに関連付けられている Adaptive Server の名前です。

database_name

削除対象の TSM のバックアップ・オブジェクトに関連付けられているデータベースの名前です。アスタリスク (*) はすべてのデータベースを表します。

object_name

dump database コマンドか dump transaction コマンドに指定された TSM のバックアップ・オブジェクトの名前です。このパラメータを省略すると、すべてのバックアップ・オブジェクトが削除されます。アスタリスク (*) はすべてのバックアップ・オブジェクトを表します。

dump_type

削除するバックアップ・オブジェクトです。値は次のとおりです。

- DB — dump database コマンドによって作成されたデータベース・バックアップ・オブジェクト。
- XACT — dump transaction コマンドによって作成されたデータベース・バックアップ・オブジェクト。
- * — すべてのデータベース・バックアップ・オブジェクト。これがデフォルト値です。

until_time

データのタイムスタンプ・フィールドです。指定された基準を満たし、until_time の日付より前に作成されたすべてのバックアップ・オブジェクトが削除されます。

bs_name

リモート Backup Server の名前です。 *bs_name* を省略すると、SYB_BACKUP がデフォルトになります。

例 **例 1** Adaptive Server “svr1” で作成されたすべての TSM のバックアップ・オブジェクトを削除します。

```
sp_deletesmobj "syb_tsm", "svr1"
```

例 2 2009 年 5 月 20 日、10:51:43:866am よりも前に “svr1” により作成された *testdb* データベースのすべてのバックアップ・オブジェクトを削除します。バックアップ・オブジェクト名は “obj1.dmp” です。

```
sp_deletesmobj "syb_tsm", "svr1", "testdb", "obj1.dmp",
    "*", "may 20, 2009 10:51:43:866am"
```

例 3 2009 年 5 月 21 日、10:51:43:866 a.m. よりも前に “svr1” により作成された *dump database* 型の “testdb” データベースのすべてのバックアップ・オブジェクトを削除します。

```
sp_deletesmobj "syb_tsm", "svr1", "'testdb'", "*", "DB",
    "may 21, 2009 10:51:43:866am"
```

例 4 2009 年 5 月 20 日、10:51:43:866 a.m. よりも前に “svr1” により作成された *dump transaction* 型の “testdb” データベースのすべてのバックアップ・オブジェクトを削除します。

```
sp_deletesmobj "syb_tsm", "svr1", "testdb",
    "/tmp/obj1.dmp", "*", "XACT", "may 21, 2009
    10:51:43:866am"
```

使用法 TSM に対する Sybase のサポートの詳細については、『THE TSM BOOK』（英語）を参照してください。

パーミッション *sp_dbremap* のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <i>dump any database</i> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <i>sa_role</i> または <i>oper_role</i> を持つユーザであることが必要。

監査 *sysaudits* テーブルの *event* カラムと *extrainfo* カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none">• <i>Roles</i> – 現在のアクティブな役割• <i>Keywords or options</i> – NULL• <i>Previous value</i> – NULL• <i>Current value</i> – NULL• <i>Other information</i> – すべての入力パラメータ• <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

[sp_querysmobj](#)

sp_depends

説明

データベース・オブジェクトの依存性の情報を表示します。これは、指定したテーブルまたはビューに依存するデータベース内のビュー、トリガ、ユーザ定義関数、プロシージャ、述部を表示したり、指定したビュー、トリガ、プロシージャ、述部が依存するデータベース内のテーブルとビューを表示します。ビューでは述部を付与できません。

また、テーブル・カラムの依存性の情報も表示します。*column_name* が指定されている場合には指定されたカラムで、*column_name* が指定されていない場合にはテーブル内のすべてのカラムでそれぞれ定義されているインデックス、デフォルト、検査制約、ルール、参照整合性制約、述部を表示します。

構文

```
sp_depends objname[, column_name]
```

パラメータ

objname

依存関係を調べられるテーブル、ビュー、Transact-SQL ストアド・プロシージャ、SQLJ ストアド・プロシージャ、SQLJ 関数、またはトリガの名前です。データベース名は指定できません。オブジェクト所有者がコマンドを実行しているユーザでない場合、またデータベース所有者でもない場合は、所有者名を使用してください。

column_name

依存性を調べられるカラムの名前です。

例

例 1 sysobjects テーブルに依存するデータベース・オブジェクトをリストします。

```
sp_depends sysobjects
```

例 2 titleview ビューに依存するデータベース・オブジェクト、および titleview ビューが依存するデータベース・オブジェクトをリストします。

```
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.tvview2	view

例 3 ユーザ "mary" が所有している titles テーブルに依存するデータベース・オブジェクトをリストします。ピリオドは特殊文字であるため、引用符が必要です。

```
sp_depends "mary.titles"
```

例 4 述部 pred1 とそれが参照するテーブルの間の依存性を表示します。

```
sp_depends pred1
Things the object references in the current database.
object          type          updated      selected
-----
dbo.tab1        user table      no           no
dbo.tab2        user table      no           no
```

例 5 述部とテーブル tab1 およびカラム col1 の間の依存性を表示します。

```
sp_depends tab1, col1
Things inside the current database that reference the object.
object          type
-----
dbo.pred1       predicate
Dependent objects that reference column col1.
Columns referenced in stored procedures, views or triggers are not included
in this report.
Type            Property
Object Names or Column Names
Also see/Use command
-----
-----
permission      permission
column permission
sp_helprotect, grant/revoke
```

例 6 sysobjects テーブルのすべてのカラムについて、カラムレベルの依存性を表示します。

```
sp_depends sysobjects
Things inside the current database that reference the object.
object          type
-----
dbo.sp_dbupgrade      stored procedure
dbo.sp_procxmode      stored procedure
```

Dependent objects that reference all columns in the table. Use sp_depends on each column to get more information.

Columns referenced in stored procedures, views or triggers are not included in this report.

Column	Type	Object Names or Column Names
cache	permission	column permission
ckfirst	permission	column permission
crdate	permission	column permission
deltrig	permission	column permission
expdate	permission	column permission
id	index	sysobjects (id)
id	logical RI	From syscolumns (id) To sysobjects (id)
id	logical RI	From syscomments (id) To sysobjects (id)
id	logical RI	From sysdepends (id) To sysobjects (id)
id	logical RI	From sysindexes (id) To sysobjects (id)
id	logical RI	From syskeys (depid) To sysobjects (id)
id	logical RI	From syskeys (id) To sysobjects (id)
id	logical RI	From sysobjects (id) To sysprocedures (id)
id	logical RI	From sysobjects (id) To sysprotects (id)
id	logical RI	sysobjects (id)
id	permission	column permission
indexdel	permission	column permission
instrig	permission	column permission
loginame	permission	column permission
name	index	ncsysobjects (name, uid)
name	permission	column permission
objspare	permission	column permission
schemacnt	permission	column permission
seltrig	permission	column permission
sysstat	permission	column permission
sysstat2	permission	column permission
type	permission	column permission
uid	index	ncsysobjects (name, uid)
uid	logical RI	From sysobjects (uid) To sysusers (uid)
uid	permission	column permission
updtrig	permission	column permission
userstat	permission	column permission
versionts	permission	column permission

例 7 sysobjects テーブルの id カラムについて、カラムレベルの依存性の詳細を表示します。

```
sp_depends sysobjects, id
```

```
Things inside the current database that reference the object.
object                                     type
```

```

-----
dbo.sp_dbupgrade                                stored procedure
dbo.sp_procxmode                                stored procedure
Dependent objects that reference column id.
Columns referenced in stored procedures, views or triggers are not included
in this report.

```

Type	Property	Object Names or Column Names Also see/Use command
index	index	sysobjects (id) sp_helpindex, drop index, sp_helpconstraint, alter table drop constraint
logical RI	primary	sysobjects (id) sp_helpkey, sp_dropkey
logical RI	foreign	From syskeys (id) To sysobjects (id) sp_helpkey, sp_dropkey
logical RI	common	From syscolumns (id) To sysobjects (id) sp_helpkey, sp_dropkey
logical RI	common	From sysdepends (id) To sysobjects (id) sp_helpkey, sp_dropkey
logical RI	common	From sysindexes (id) To sysobjects (id) sp_helpkey, sp_dropkey
logical RI	common	From syskeys (depid) To sysobjects (id) sp_helpkey, sp_dropkey
logical RI	common	From syscomments (id) To sysobjects (id) sp_helpkey, sp_dropkey
logical RI	common	From sysobjects (id) To sysprotects (id) sp_helpkey, sp_dropkey
logical RI	common	From sysobjects (id) To sysprocedures (id) sp_helpkey, sp_dropkey
permission	permission	column permission sp_helprotect, grant/revoke

例 8 ユーザが作成したテーブル (titles) のすべてのカラムについて、カラムレベルの依存性を表示します。

```
1> sp_depends titles
```

```
Things inside the current database that reference the object.
```

object	type
dbo.delttitle	trigger
dbo.history_proc	stored procedure
dbo.title_proc	stored procedure
dbo.titleid_proc	stored procedure
dbo.titleview	view

```
dbo.totalsales_trig          trigger
```

Dependent objects that reference all columns in the table. Use sp_depends on each column to get more information. Columns referenced in stored procedures, views or triggers are not included in this report.

Column	Type	Object Names or Column Names
pub_id	logical RI	From titles (pub_id) To publishers (pub_id)
pubdate	default	datedflt
title	index	titleind (title)
title	statistics	(title)
title_id	index	titleidind (title_id)
title_id	logical RI	From roysched (title_id) To titles (title_id)
title_id	logical RI	From salesdetail (title_id) To titles (title_id)
title_id	logical RI	From titleauthor (title_id) To titles (title_id)
title_id	logical RI	titles (title_id)
title_id	rule	title_idrule
title_id	statistics	(title_id)
type	default	typedflt

例 9 ユーザが作成した titles テーブルの pub_id カラムについて、カラムレベルの依存性の詳細を表示します。

```
sp_depends titles, pub_id
```

Things inside the current database that reference the object.

object	type
dbo.delttitle	trigger
dbo.history_proc	stored procedure
dbo.title_proc	stored procedure
dbo.titleid_proc	stored procedure
dbo.titleview	view
dbo.totalsales_trig	trigger

Dependent objects that reference column pub_id. Columns referenced in stored procedures, views or triggers are not included in this report.

Type	Property	Object Names or Column Names Also see/Use command
logical RI	foreign	From titles (pub_id) To publishers (pub_id) sp_helpkey, sp_dropkey

使用法

- `sp_depends` を実行すると、*objname* に依存する、現在のデータベース内のすべてのオブジェクトと、*objname* が依存するすべてのオブジェクトが表示されます。たとえば、ビューは1つ以上のテーブルに依存しますが、そのビューに依存するプロシージャまたは他のビューを持つことができます。別のオブジェクトを参照するオブジェクトは、そのオブジェクトに依存します。現在のデータベースの外のオブジェクトへの参照は、レポートされません。
- カラムを修正または削除する前に、修正による影響を受ける可能性のある従属オブジェクトがテーブル内にあるかどうかを確認するために、`sp_depends` を使用します。たとえば、新しいデータ型を使用するためにカラムを修正すると、テーブルに結合しているオブジェクトを再定義して、カラムの新しいデータ型と一致させる必要があります。
- `sp_depends` プロシージャは、`sysdepends` テーブルを調べて依存性を判断します。

オブジェクトが正しい順序で作成されなかった場合(たとえば、ビューを作成する前にそのビューを使用するプロシージャが作成されている場合など)、その依存性を示すローは `sysdepends` 内にはありません。また、`sp_depends` はその依存性をレポートしません。

- `sp_depends` からのレポートの `updated` カラムと `selected` カラムは、レポートされたオブジェクトがストアド・プロシージャかトリガの場合に意味を持ちます。更新された `updated` カラムの値は、ストアド・プロシージャがそのオブジェクトを更新するかどうかを示します。選択された `selected` カラムは、オブジェクトが読み込みカーソルまたはデータ修正文のために使用されるかどうかを示します。
- `sp_depends` は、次のような Adaptive Server の規則に従ってオブジェクトを見つけます。
 - ユーザが所有者名を指定せず、コマンドを実行しているユーザが指定された名前のオブジェクトを所有している場合は、そのオブジェクトが使用されます。
 - ユーザが所有者名を指定せず、その名前のオブジェクトを所有していない場合、データベース所有者がその名前のオブジェクトを所有していれば、データベース所有者のオブジェクトが使用されます。
 - ユーザもデータベース所有者もその名前のオブジェクトを所有していない場合、その名前のオブジェクトがデータベースに存在していても、所有者が異なると、コマンドはエラー条件をレポートします。

- 指定された名前のオブジェクトをユーザとデータベース所有者がともに所有していて、ユーザがデータベース所有者のオブジェクトをアクセスしたい場合は、*dbo.objectname* のように名前を指定しなければなりません。
- コマンドを実行しているユーザとデータベース所有者以外のデータベース・ユーザが所有しているオブジェクトは、例 3 に示すように、常に所有者の名前で修飾しなければなりません。
- **SQLJ 関数と SQLJ ストアド・プロシージャは、SQL ラップでラップされている Java メソッドです。** 詳細については、『**Adaptive Server Enterprise における Java**』を参照してください。
- **SQLJ 関数と SQLJ ストアド・プロシージャは、従属オブジェクトをリストできるデータベース・オブジェクトです。** SQLJ ストアド・プロシージャと SQLJ 関数の従属オブジェクトのみが Java クラスとなります。
- *objname* が SQLJ ストアド・プロシージャまたは SQLJ 関数である場合、`sp_depends` は、`create` 文で宣言されたルーチンの外部名内に Java クラスをリストします。パラメータ・リスト内でリターン・タイプまたは `return` データ型として指定されているクラスはリストされません。
- SQLJ ストアド・プロシージャと SQLJ 関数は、他のデータベース・オブジェクトの従属オブジェクトとしてリストされます。

パーミッション

すべてのユーザが `sp_depends` を実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [create procedure](#)、[create table](#)、[create view](#)、[execute](#)

システム・プロシージャ [sp_help](#)

sp_deviceattr

説明 (UNIX プラットフォームのみ) 既存のデータベース・デバイス・ファイルのデバイス・パラメータ設定を変更します。

構文 `sp_deviceattr logicalname, optname, optvalue`

パラメータ

logicalname

既存のデータベース・デバイスの論理名です。デバイスは、オペレーティング・システム・ファイルまたはロー・パーティションのどちらかに格納できますが、ロー・パーティションの `dsync` 設定は無視されます。

optname

変更する属性の名前です。有効な値は `directio` と `dsync` です。

- `directio` — Adaptive Server がオペレーティング・システムのバッファ・システムをバイパスしてディスクに直接書き込めるようにします。
- `dsync` — デバイスに対する更新を記憶メディアに対して直接行うか、UNIX ファイル・システムによってバッファされるかを指定できます。

注意 `directio` オプションと `dsync` オプションは、いずれか1つのみを指定できます。両方同時に "true" に設定することはできません。

optvalue

“true” または “false” を指定できます。

例 “file_device1” という名前のデバイスについて `dsync` をオンに設定します。

```
sp_deviceattr file_device1, dsync, true
```

使用法

- UNIX ファイルに格納されたデータベース・デバイスの場合、`dsync` は、デバイスに対する更新が記憶メディアに対して直接行われるか、UNIX ファイル・システムによってバッファされるかを指定します。

`dsync` がオンの場合、データベース・デバイスへの書き込みは物理記憶メディアに対して直接行われ、Adaptive Server は、システム障害が発生したときにデバイス上のデータをリカバリできます。

`dsync` がオフの場合、データベース・デバイスへの書き込みは UNIX ファイル・システムによってバッファされることがあります。UNIX ファイル・システムは、物理メディアがまだ修正されていない場合でも、更新が完了したものとしてマークを付けることがあります。このため、システム障害時にデータ更新要求が物理メディアに対して行われるという保証はなく、Adaptive Server はデータベースをリカバリできない場合があります。

- (UNIX のみ) ロー・デバイスでは、`sp_deviceattr` ストアド・プロシージャによって `directio` または `dsync` を `true` に設定することはできません。

注意 HP-UX の場合、`dsync` オプションのみが適用されます。

これを行うと、次のようなメッセージが返されます。

```
You cannot set option dsync for raw device 'dev/raw/raw235'
```

または

```
You cannot set attribute dsync for raw device 'myrawdsk1'
```

- `sp_deviceattr` を使用して `dsync` または `directio` 設定を変更した後、Adaptive Server をリブートして、この変更内容を有効にしてください。
- データベース・デバイス・ファイルについて `dsync` オプションが無効になっている場合、`sp_deviceattr` は警告メッセージを表示します。

`directio` 設定を変更した後、Adaptive Server をリブートして、この変更内容を有効にしてください。

- `dsync` は、マスタ・デバイス・ファイルに対して常にオンに設定されます。マスタ・デバイス・ファイルの `dsync` 設定は、`sp_deviceattr` によって変更できません。したがって、マスタ・デバイスに `directio` オプションは設定できません。
- `dsync` の値は、システム障害後にデバイス上のデータベースをリカバリする必要がない場合にのみオフにします。たとえば、`tempdb` データベースだけを格納するデバイスの場合は、`dsync` をオフに設定できます。
- Adaptive Server は、ロー・パーティションに格納されたデバイスについては、`dsync` の設定を無視します。これらのデバイスへの更新は、`dsync` の設定に関係なくバッファされません。
- `dsync` は、Windows NT プラットフォームでは使用できません。

パーミッション sp_deviceattr のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage disk 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_helpdevice](#)

sp_diskdefault

説明	ユーザがデータベース・デバイスを指定しない場合や、 create database または alter database コマンドで default を指定する場合に、データベースの記憶領域用にデータベース・デバイスが使用できるかどうかを指定します。
構文	<code>sp_diskdefault <i>logicalname</i>, {defaulton defaultoff}</code>
パラメータ	<p><i>logicalname</i></p> <p><code>master.dbo.sysdevices.name</code> 内で特定されたデバイスの論理名です。特定するデバイスは、ダンプ・デバイスではなく、データベース・デバイスにしてください。</p> <p><code>defaulton defaultoff</code></p> <p>データベース・デバイスをデフォルト・データベース・デバイスとする場合は <code>defaulton</code> であり、指定のデータベース・デバイスをデフォルト・データベース・デバイスとしない場合は <code>defaultoff</code> です。</p> <p><code>defaulton</code> は、データベース・デバイスを disk init によってシステムに追加した後に使用します。<code>defaultoff</code> は、マスタ デバイスのデフォルトのステータスを変更するために使用します (このデバイスは、Adaptive Server を初めてインストールしたときのデフォルト・デバイスとして指定されています)。</p>
例	<p>create database または alter database がマスタ・デバイスをデータベースのデフォルトの記憶領域として使用できないようにします。</p> <pre>sp_diskdefault master, defaultoff</pre>
使用法	<ul style="list-style-type: none">デフォルト・データベース・デバイスは、ユーザがデータベース・デバイス名を指定しない場合や、<code>default</code> キーワードを指定する場合に、データベース記憶領域として create database や alter database で使用されるデバイスです。デフォルト・デバイスを複数持つことができます。これらのデバイスは、<code>master.dbo.sysdevices</code> テーブルにある順序 (アルファベット順) で使用されます。最初のデフォルト・デバイスがいっぱいになると、2 番目以降のデフォルト・デバイスが順に使用されます。

- Adaptive Server を初めてインストールしたときは、マスタ・デバイスが唯一のデフォルト・データベース・デバイスです。

注意 ユーザ・データベースを保管するためにデバイスを初期化したら、`sp_diskdefault` を使用してマスタ・デバイスのデフォルト・ステータスをオフにしてください。このようにすると、ユーザがマスタ・デバイス上に誤ってデータベースを作成することがなくなり、`master` データベースのリカバリが容易になります。

- どのデータベース・デバイスがデフォルト・データベース・デバイスかを調べるには、`sp_helpdevice` を実行します。

パーミッション

`sp_diskdefault` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage disk</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [alter database](#)、[create database](#)、[disk init](#)
 システム・プロシージャ [sp_helpdevice](#)

sp_displayaudit

説明	監査オプションのステータスを表示します。
構文	<pre>sp_displayaudit ["procedure" "object" "login" "database" "global" "default_object" "default_procedure" [, "name"]]</pre>
パラメータ	<p>procedure 指定されたストアド・プロシージャまたはトリガの監査オプションのステータスを表示します。<i>name</i> に値を指定しない場合、sp_displayaudit は現在のデータベースにあるすべてのプロシージャとトリガのアクティブな監査オプションを表示します。</p> <p>object 指定されたテーブルまたはビューの監査オプションのステータスを表示します。<i>name</i> に値を指定しない場合、sp_displayaudit は現在のデータベースにあるすべてのテーブルとビューのアクティブな監査オプションを表示します。</p> <p>login 指定されたユーザ・ログインの監査オプションのステータスを表示します。<i>name</i> に値を指定しない場合、sp_displayaudit は、master データベースにあるすべてのログインのアクティブな監査オプションを表示します。</p> <p>database 指定されたデータベースの監査オプションのステータスを表示します。<i>name</i> の値を指定しない場合、sp_displayaudit は、サーバ上のすべてのデータベースのアクティブな監査オプションを表示します。</p> <p>global 指定されたグローバル監査オプションのステータスを表示します。<i>name</i> に値を指定しない場合、sp_displayaudit は現在のデータベースにあるすべてのプロシージャとトリガのアクティブな監査オプションを表示します。</p> <p>default_object 指定されたデータベース上に作成された新しいテーブルまたはビューに使用されるデフォルト監査オプションを表示します。<i>name</i> に値を指定しない場合、sp_displayaudit はアクティブなデフォルト監査設定とすべてのデータベースのデフォルト監査オプションを表示します。</p>

default_procedure

指定されたデータベース上に作成された新しいプロシージャまたはトリガに使用されるデフォルト監査オプションを表示します。

name に値を指定しない場合、`sp_displayaudit` はアクティブなデフォルト監査設定とすべてのデータベースのデフォルト監査オプションを表示します。

name

指定されたパラメータの情報です。各パラメータについては次の表で説明しています。

パラメータ	<i>name</i> の値
procedure	プロシージャ名またはトリガ名
object	テーブル名またはビュー名
login	ユーザ・ログイン
database	データベース名
global	グローバル監査オプション
default_object	データベース名
default_procedure	データベース名

例

例 1 パラメータが指定されない場合、各カテゴリーのステータスとすべての監査オプションを表示します。

`sp_displayaudit`

```

Procedure/Trigger      Audit Option  Value Database
-----
dbo.sp_altermessage    exec_procedure on    sybssystemprocs
dbo.sp_help            exec_procedure on    sybssystemprocs
dbo.sp_who             exec_procedure on    sybssystemprocs
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
keycustodian_role    off
navigator_role        off

```

```
oper_role                off
replication_role        off
rpc                      off
sa_role                  off
security                 off
sso_role                 off
```

例 2 プロシージャ名が指定されない場合、すべてのプロシージャ監査オプションのステータスを表示します。

```
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

例 3 プロシージャ名が指定された場合、そのプロシージャのステータスだけを表示します。

```
sp_displayaudit "procedure", "sp_who"
```

Procedure/Trigger	Audit Option	Value	Database
dbo.sp_who	exec_procedure	on	sybsystemprocs

例 4 グローバル監査オプションが指定されない場合、すべてのグローバル監査オプションのステータスを表示します。

```
sp_displayaudit "global"
```

Option Name	Value
adhoc	off
dbcc	off
disk	off
errors	off
login	off
logout	off
keycustodian_role	off
navigator_role	off
oper_role	off
replication_role	off
rpc	off
sa_role	off
security	off
sso_role	off

- 使用法
- sp_displayaudit は、監査オプションのステータスを表示します。
 - 次の表は、各パラメータについて有効な監査オプションを示しています。

オブジェクト・タイプ・パラメータ	有効な監査オプション
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、encryption_key、func_dbaccess、grant、load、revoke、setuser、truncate、unbind
global	adhoc、dbcc、disk、errors、login、logout、navigator_role、oper_role、replication_role、rpc、keycustodian_role、sa_role、security、sso_role
default_object	delete、func_obj_access、insert、reference、select、update
default_procedure	exec_procedure、exec_trigger

- 最初にオブジェクト・タイプ・パラメータを指定しなければ、名前用の値を指定することはできません。

パーミッション sp_displayaudit のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage auditing 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sso_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照 マニュアル 監査の設定の詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_audit](#)

ユーティリティ [bcp](#)

sp_displaylevel

説明 `sp_configure` の出力に Adaptive Server のどの設定パラメータを表示するかを設定します。または、その設定を表示します。

構文 `sp_displaylevel [loginame [, level]]`

パラメータ

loginame

表示レベルを設定または表示するユーザの Adaptive Server ログインです。

level

表示レベルを次のいずれかに設定します。

- “basic” (基本) 表示レベルは、最も基本的な設定パラメータだけを表示します。一般的なサーバのチューニングに適しています。
- “intermediate” (中間) 表示レベルは、basic レベルのすべてのパラメータのほかに、より複雑な設定パラメータも表示します。このレベルは、あまり複雑でないサーバのチューニングに適しています。
- “comprehensive” (包括) 表示レベルは、最も複雑な設定パラメータも含め、すべての設定パラメータを表示します。このレベルは非常に細かいサーバのチューニングに適しています。

例 **例 1** `sp_displaylevel` を実行したユーザの現在の表示レベルを表示します。

```
sp_displaylevel
```

```
The current display level for login 'sa' is 'comprehensive'.
```

例 2 ユーザ “jerry” の現在の表示レベルを表示します。

```
sp_displaylevel jerry
```

```
The current display level for login 'jerry' is 'intermediate'.
```

例 3 ユーザ “jerry” の表示レベルを “comprehensive” に設定します。

```
sp_displaylevel jerry, comprehensive
```

```
The display level for login 'jerry' has been changed to 'comprehensive'.
```

使用法 表示レベルと設定パラメータの詳細については、『システム管理ガイド』を参照してください。

パーミッション sp_displaylevel のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage server configuration 権限を持つユーザーであることが必要。 どのユーザーも sp_displaylevel を実行して自分の表示レベルを設定および表示できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sso_role が付与されたユーザーであることが必要。 どのユーザーも sp_displaylevel を実行して自分の表示レベルを設定および表示できる。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_configure](#)

sp_displaylogin

説明 ログイン・アカウント情報を表示します。ワイルドカード(%)を使用することで、一致するログイン情報を取得することもできます。また、ディスクに格納されているログイン・パスワードの暗号化のバージョンを表示します。

構文 sp_displaylogin ['*user_id*' | ['*loginame* | *wildcard*']

パラメータ *user_id*
サーバ・ユーザ ID です。

loginame
情報を表示したい、自分以外のユーザのログイン・アカウントです。他のユーザのログイン・アカウント情報を取得するには、システム・セキュリティ担当者またはシステム管理者の権限が必要です。

wildcard
ログイン・アカウントの情報を取得するために使用するワイルドカード式です。

例 **例 1** 使用しているサーバ・ログイン・アカウント情報を表示します。パスワードの有効期間が“0”に設定されており、パスワードが無期限になります。

```
1> sp_displaylogin 'sa'
2> go

Suid:121
Loginame : sa
Fullname:
Configured Authorization:
    sa_role (default ON)
    sso_role (default ON)
    oper_role (default ON)
    sybase_ts_role (default ON)
Locked:NO
Date of Last Password Change :Aug 10 2010 11:17AM
Password expiration interval:0
Password expired:NO
Minimum password length:6
Maximum failed logins:0
Current failed login attempts:
Login password encryption:SYB-PROP, SHA-256
Last login date:Aug 17 2010 5:55PM
Login Profile :emp_lp
```

例 2 ログイン・アカウント “susanne” の情報を表示します。表示される情報は、sp_displaylogin を実行するユーザの役割によって異なります。ユーザ “susanne” にはパスワードの有効期間が設定されていないため、パスワードが無期限になります。

```
sp_displaylogin susanne

Suid:12
Loginame:susanne
Fullname:
Configured Authorization:
    supervisor (default OFF)
Locked:NO
Date of Last Password Change :July 26 2010 10:42AM
Login Profile :emp_lp
```

例 3 ログインに対して設定されているログイン・セキュリティ関連のパラメータと、指定された認証メカニズムを表示します。パスワードの期限は、2010年11月29日、3:46PMに切れ、5日後の2010年12月5日、3:46PMに切れます。

```
sp_displaylogin joe

Suid:294
Loginame:joe
Fullname:Joseph Resu
Configured Authorization:
    intern_role (default OFF)
Locked:NO
Date of Last Password Change :Nov 24 2010 3:46PM
Password expiration interval:5
Password expired:NO
Minimum password length:4
Maximum failed logins:10
Current failed logins:3
Login password encryption:SHA-256
Login Profile :emp_lp
```

例 4 サーバ・ユーザ ID が 1 のログイン・アカウント情報を表示します。

```
sp_displaylogin '1'
-----
Suid:1
Loginame:sa
Fullname:
Configured Authorization:
    sa_role (default ON)
    sso_role (default ON)
    oper_role (default ON)
```



```

        sybase_ts_role (default ON)
Locked:NO
Date of Last Password Change :Dec 18 2010
Password expiration interval:0
Login Profile :emp_lp

```

例5 ワイルドカードを使用して、自分のサーバ・ログイン・アカウントではなく、他のサーバ・ログイン・アカウントを示すことができます。

```
sp_displaylogin '%'
```

```

-----
Suid Loginname Fullname Locked Date of Last Password Change
Password expiration interval Password expired Minimum password length Maximum
failed logins Current failed login attempts Authenticate with Login
Profile Configured Authorization
-----
-----
-----
-----
-----
-----

```

```

2 probe NULL sybssystemdb NULL NULL NO Jan  8 2010 7:13AM 1 NO 6 0 0 NONE
NULL
1 sa NULL master NULL NULL NO Jan  8 2010 6:46AM 1 NO 6 0 0 NONE

```

例6 ディスク上のログイン・パスワードが、Sybase独自の古い暗号化アルゴリズムとSHA-256アルゴリズムの両方を使用して暗号化されて格納されています。

```

1> sp_displaylogin 'mylogin'
2> go

```

```

Suid:121
Loginname:mylogin
Fullname:
Configured Authorization:
        sa_role (default ON)
        sso_role (default ON)
        oper_role (default ON)
        sybase_ts_role (default ON)
Locked:NO
Date of Last Password Change :Aug 10 2006 11:17AM
Password expiration interval:0
Password expired:NO
Minimum password length:6
Maximum failed logins:0
Current failed login attempts:
Login password encryption:SYB-PROP, SHA-256

```

```
Last login date:Aug 17 2010 5:55PM
Login Profile :emp_lp
```

```
(return status = 0)
```

ログイン・パスワードが **SHA-256** アルゴリズムのみを使用してディスクに格納されている場合は、`sp_displaylogin` の出力には “Login password encryption: SHA-256” 行が表示されます。

```
1> sp_displaylogin 'mylogin'
2> go

Suid:121
Loginname:mylogin
...
Authenticate with:NONE
Login password encryption:SHA-256
Last login date:Aug 17 2010 5:55PM
Login Profile :emp_lp
```

```
(return status = 0)
```

バージョン 15.0.2 以前の **Adaptive Server** からのアップグレード後にログインが発生しなかったアカウントにはまだ古い暗号化が残っているので、`sp_displaylogin` の出力には “Login password encryption: SYB-PROP” 行が表示されます。

```
1> sp_displaylogin 'mylogin'
2> go

Suid:121
Loginname:mylogin
...
Authenticate with:NONE
Login password encryption:SYB-PROP
Last login date:Aug 17 2006 5:55PM
(return status = 0)
```

ログインがロックされている場合、`sp_displaylogin` は、アカウントがロックされた日付、理由、ログイン情報を表示します。`lastlogindate` の値も次のように表示されます。

```
1> sp_displaylogin 'mylogin'
2> go

Suid:121
Loginname:mylogin
Fullname:
Configured Authorization:
    sa_role (default ON)
```

```
sso_role (default ON)
oper_role (default ON)
sybase_ts_role (default ON)
Locked: YES
Date when locked: Aug 18 2010 9:15AM
Reason: Account locked by Adaptive Server due to failed
login attempts reaching max failed logins.
Locking suid: mylogin
Date of Last Password Change : Aug 10 2010 11:17AM
Password expiration interval: 0
Password expired: NO
Minimum password length: 6
Maximum failed logins: 3
Current failed login attempts: 3
Login password encryption: SYB-PROP, SHA-256
Last login date: Aug 17 2010 5:55PM
Login Profile: emp_lp
(return status = 0)
```

例 7 ログインに使用された暗号化のバージョンを表示します。この出力には、Adaptive Server が使用しているディスク上のログイン・パスワード暗号化方式に関する情報が含まれています。

```
sp_displaylogin sa
go
Suid: 1
Loginname: sa
Fullname:
Configured Authorization:
  sa_role (default ON)
  sso_role (default ON)
  oper_role (default ON)
  sybase_ts_role (default ON)
Locked: NO
Date of Last Password Change : Mar  8 2010 3:04PM
Password expiration interval: 0
Password expired: NO
Minimum password length: 6
Maximum failed logins: 0
Current failed login attempts:
Login Password Encryption: SHA-256
Login Profile : emp_lp
```

Adaptive Server が、ダウングレード期間にバージョン 15.0.2 以前の Adaptive Server または現在のリリースの暗号化アルゴリズムを使用する場合は、sp_displaylogin は、以前の Sybase 独自の暗号化アルゴリズムと新しいアルゴリズム SHA-256 を表示します。

```
Login password encryption:SYB-PROP, SHA-256
```

例 8 現在のログイン・アカウントのログインおよびパスワード・ポリシーのオプションを表示します。

```
sp_displaylogin
go
Suid:5
Loginname:tammi
Fullname:
Configured Authorization:
    sa_role (default ON)
    sso_role (default ON)
    oper_role (default ON)
    sybase_ts_role (default ON)
Locked:NO
Date of Last Password Change :Mar  8 2010 3:04PM
Password expiration interval:0
Password expired:NO
Minimum password length:6
Maximum failed logins:0
Current failed login attempts:
Authenticate with:ANY
Login Password Encryption:SHA-256
Exempt inactive lock:0
Login Profile:emp_lp
```

例 9 56 という suid を持つユーザのログイン・アカウントを表示します。

```
sp_displaylogin '56'
```

“st” で始まるすべてのユーザのログイン・アカウントの情報を表示します。

```
sp_displaylogin 'st%'
```

使用法

- パスワード有効期限、最大失敗ログイン回数、およびパスワード長に関連するログイン情報を表示するときに、sp_passwordpolicy セキュリティ・オプションが考慮されます。
- sp_displaylogin は、ログインに使用された暗号化のバージョンを表示します。たとえば、パスワード・ダウングレード期間に新旧両暗号化が使用されているときには、sp_displaylogin の出力に含まれている新しい “Password encryption” の行にその情報が表示されません。
- sp_displaylogin は、役割が set コマンドによってアクティブでない状態にされていたとしても、設定されている役割を表示します。

- 該当のログインに関連するログイン・トリガは、ログイン・プロファイルによって指定されています。詳細については、『システム管理ガイド』の「ログイン・アカウントとログイン・プロファイルの管理」を参照してください。
- `sp_displaylogin` を使用して自分自身のアカウントに関する情報を取得する場合、`loginame` パラメータは不要です。`sp_displaylogin` によって表示される項目は、使用しているサーバ・ユーザ ID、ログイン名、ログイン・プロファイル、フルネーム、各自に付与されたすべての役割、最後のパスワード変更日付、および使用しているアカウントがロックされているかどうかの情報です。
- システム・セキュリティ担当者かシステム管理者の権限を持っていれば、`loginame` パラメータを使用して、すべてのアカウントの情報にアクセスできます。

パーミッション

`sp_displaylogin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any login</code> 権限または <code>manage sever</code> 権限を持つユーザであることが必要。 どのユーザも <code>sp_displaylogin</code> を実行して自分のログイン・アカウントの情報を表示できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> または <code>sso_role</code> が付与されたユーザであることが必要。 どのユーザも <code>sp_displaylogin</code> を実行して自分のログイン・アカウントの情報を表示できる。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

ストアド・プロシージャ [sp_activeroles](#)、[sp_displayroles](#)、[sp_helprotect](#)

sp_displayroles

説明

他の役割に付与されたすべての役割、ログインまたはログイン・プロフィール、役割の階層ツリー全体 (表形式)、指定された役割に対して設定されている他のログイン・セキュリティ関連のパラメータ (役割がロックされた日付、ロックの理由、および役割をロックしたログイン・サーバ・ユーザ ID (suid) を含む) を表示します。パスワードで保護されている役割の場合、役割のパスワードの暗号化バージョンも表示されます。

関連付けられたログイン・プロフィールによりログインに付与された役割を表示します。出力の **grantee** カラムには、該当するログイン・プロフィール名が表示されます。このカラムは、ログインに役割が付与された関連ログイン・プロフィールがある場合のみ表示されません。ログイン・プロフィールは、直接関連付けられるか、デフォルトのログイン・プロフィールによって関連付けられます。

構文

```
sp_displayroles [grantee_name [, mode]]
```

パラメータ

grantee_name

役割についての情報を取得するユーザのログイン名またはログイン・プロフィール名、または情報を取得する役割名です。

mode

次のいずれかを指定します。

- **expand_up** — 親レベルの役割階層ツリーを表示します。
- **expand_down** — 子レベルの役割階層ツリーを表示します。
- **display_info** — 指定された役割に対して設定されているログイン・セキュリティ関連のパラメータを表示します。

例 1 コマンドを発行するユーザに付与されているすべての役割を表示します。

```
sp_displayroles

Role Name
-----
supervisor_role
```

例 2 supervisor_role に付与されているすべての役割を表示します。

```
sp_displayroles "supervisor_role"

Role Name
-----
clerk
```

例 3 ログイン “susanne” に付与された役割と、階層内でそれより下位にある役割を表示します。

```
sp_displayroles susanne, expand_down
```

Role Name	Parent Role Name	Level
supervisor_role	NULL	1
clerk_role	supervisor_role	2

例 4 intern_role に付与された役割と、階層内でそれより上位の役割を表示します。

```
sp_displayroles "intern_role", expand_up
```

例 5 特定の役割用に設定されたログイン・セキュリティ関連のパラメータを表示します。

```
sp_displayroles physician_role, "display_info"
```

```
Role name = physician_role
Locked: YES
  Date when locked: Jul 14 2007 9:15AM
  Reason: Role locked by Adaptive Server due to failed login
  attempts reaching max failed logins.
  Locking suid: dr_john
```

```
Date of Last Password Change : Oct 31 1999 3:33PM
Password expiration interval = 5
Password expired: NO
Minimum password length = 4
Maximum failed logins = 10
Current failed logins = 3
Password encryption version: SHA-256
```

例 6 “sec_profile” という名前のログイン・プロファイルに関連付けられたログイン “tom” に付与された役割を表示します。

```
grant role sec_role to sec_profile

create login tom with password C0mp13x login profile sec_profile

grant role emp_role to tom
go
sp_displayroles tom
go
```

```
Role Name                               Grantee
-----
```

```
emp_role          tom
sec_role          sec_profile
```

使用法 オプション・パラメータの `expand_up` または `expand_down` を指定すると、指定された役割名に含まれる、または指定された役割名を含む、直接付与されたすべての役割が表示されます。

`Grantee` カラムは、ログインに関連ログイン・プロファイルがある場合、またはデフォルトのログイン・プロファイルでログインに役割が付与されている場合にのみ表示されます。

パーミッション `sp_displayroles` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage roles</code> または <code>manage server</code> 権限を持つユーザであることが必要。 どのユーザも <code>sp_displayroles</code> を実行して自分自身に付与された役割を表示できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sso_role</code> が付与されたユーザであることが必要。 どのユーザも <code>sp_displayroles</code> を実行して自分自身に付与された役割を表示できる。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — すべての入力パラメータ <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 マニュアル 詳細については、『システム管理ガイド』の「ユーザ定義のログイン・セキュリティ」を参照してください。

コマンド `alter role`、`create role`、`drop role`、`grant`、`revoke`、`set`

システム・プロシージャ `sp_activeroles`、`sp_displaylogin`、`sp_helprotect`

sp_downgrade

説明	(master データベースのみ)以前の 15.0.x リリースへのダウングレードの準備ができていかどうかを検証します。Adaptive Server 15.0.2 によって変更されたシステム・カタログの変更もダウングレードされます。
構文	<pre>sp_downgrade @cmd = {'prepare' 'downgrade' 'help'}, @toversion = 'n'[, @verbose = 0 1][, @override = 0 1]</pre>
パラメータ	<p>prepare Adaptive Server 15.0.2 がダウングレードの準備ができていかどうかを検証するために最初に使用します。</p> <p>downgrade 以前にインストールした 15.x バージョンの Adaptive Server にダウングレードする準備ができてい場合、prepare パラメータの後に使用します。サーバはシングルユーザ・モード (-m オプションで始める) にしてください。</p> <p>toversion 15.0 または 15.0.1 を指定できます。具体的には “150” または “15.0”、“1501” あるいは “15.0.1” と指定してください。</p> <p>verbose 詳細度を指定します。有効なオプションは 0 (指定しない) または 1 (指定する) です。</p> <p>override この時点で書き込み可能でないデータベースを省略するかどうかを指定します。有効なオプションは 0 (指定しない) または 1 (指定する) です。</p>
例	<p>例 1 この例は、sp_downgrade を実行して生成された出力を示しています。</p> <pre>00:0006:00000:00006:2011/06/29 02:16:44.35 server Preparing ASE downgrade from 15.7.0.0 to 15.5.0.0. 00:0006:00000:00006:2011/06/29 02:16:44.37 server Starting downgrading ASE. 00:0006:00000:00006:2011/06/29 02:16:44.37 server Downgrade : Marking stored procedures to be recreated from text. 00:0006:00000:00006:2011/06/29 02:16:45.34 server Downgrade : Removing full logging modes from sysattributes. 00:0006:00000:00006:2011/06/29 02:16:45.34 server Downgrade : Downgrading data-only locked table rows. 00:0006:00000:00006:2011/06/29 02:16:45.34 server Downgrade : Removing full logging modes from sysattributes. 00:0006:00000:00006:2011/06/29 02:16:45.34 server Downgrade : Removing column sysoptions.number. 00:0006:00000:00006:2011/06/29 02:16:45.34 server Downgrade : Removing srvprincipal column from syssservers system table 00:0006:00000:00006:2011/06/29 02:16:45.34 server Downgrade : Removing 'automatic</pre>

```
master key access' configuration parameter.
00:0006:00000:00006:2011/06/29 02:16:45.35 server Downgrade : Removing DualControl
sysattribute rows
00:0006:00000:00006:2011/06/29 02:16:45.35 server Downgrade : ダウングレード
sysattributes system table.
00:0006:00000:00006:2011/06/29 02:16:45.37 server Downgrade : ダウングレード
syscomments system table.
00:0006:00000:00006:2011/06/29 02:16:45.42 server Downgrade : Truncated role
password, locked role and removed columns locksuid, lockreason, lockdate from
sysssrvroles
00:0006:00000:00006:2011/06/29 02:16:45.43 server Downgrade : Removing catalog
changes for RSA Keypair Regeneration Period and Login Profile
00:0006:00000:00006:2011/06/29 02:16:45.43 server Downgrade : Turning on database
downgrade indicator.
00:0006:00000:00006:2011/06/29 02:16:45.43 server Downgrade : Resetting database
version indicator.
00:0006:00000:00006:2011/06/29 02:16:45.43 server ASE downgrade completed.
```

例 2 ダウングレードの準備ができていのかどうかについてデータベースを確認します。

```
1> sp_downgrade 'prepare', '15.5', 1
2> go
Downgrade from 15.7.0.0 to 15.5.0.0 (command: 'prepare')
```

Checking databases for downgrade readiness.

```
There are no errors which involve encrypted columns.
sp_downgrade 'prepare' completed.
(return status = 0)
```

例 3 Adaptive Server バージョン 15.7 から 15.5 にダウングレードします。

```
1> sp_downgrade 'downgrade', '15.5', 1
2> go

Downgrade from 15.7.0.0 to 15.5.0.0 (command: 'downgrade')
```

Checking databases for downgrade readiness.

There are no errors which involve encrypted columns.

```
Executing downgrade step 2 [dbcc markprocs(@dbid)] for :
- Database:master (dbid:1)
sql comman is: dbcc markprocs(@dbid)
```

DBCC execution completed.If DBCC printed error messages, contact a user with System Administrator (SA) role.

```
- Database:tempdb (dbid: 2)
sql comman is: dbcc markprocs(@dbid)
```

DBCC execution completed.If DBCC printed error messages, contact a user with System Administrator (SA) role.

```
- Database:model (dbid: 3)
sql comman is: dbcc markprocs(@dbid)
```

DBCC execution completed.If DBCC printed error messages, contact a user with System Administrator (SA) role.

```
- Database:sybsystemdb (dbid: 31513)
sql comman is: dbcc markprocs(@dbid)
```

DBCC execution completed.If DBCC printed error messages, contact a user with System Administrator (SA) role.

```
- Database:sybsystemprocs (dbid: 31514)
sql comman is: dbcc markprocs(@dbid)
```

DBCC execution completed.If DBCC printed error messages, contact a user with System Administrator (SA) role.

Executing downgrade step 17 [delete sysattributes where class = 38] for :

```
- Database:master (dbid:1)
sql comman is: delete sysattributes where class = 38
```

```
Executing downgrade step 18 [declare @ret int select @ret =
dol_downgrade_check(':DBNAME:', @toversid) print "Database :DBNAME: table downgrade
status: %1!", @ret if @ret != 0 begin print "*** Tables in database ':DBNAME:' cannot
be downgraded." print "*** See the server error log for details." select
@exec_error_count = @exec_error_count + 1 end] for :
```

```
- Database:master (dbid:1)
sql comman is: declare @ret int select @ret = dol_downgrade_check('master', @toversid)
print "Database master table downgrade status: %1!", @ret if @ret != 0 begin print
"*** Tables in database 'master' cannot be downgraded." print "*** See the server
error log for details." select @exec_error_count = @exec_error_count + 1 end
```

Database master table downgrade status: 0

```
- Database:tempdb (dbid: 2)
```

```
sql comman is: declare @ret int select @ret = dol_downgrade_check('tempdb', @toversid)
print "Database tempdb table downgrade status: %1!", @ret if @ret != 0 begin print
"*** Tables in database 'tempdb' cannot be downgraded." print "*** See the server
error log for details." select @exec_error_count = @exec_error_count + 1 end
```

Database tempdb table downgrade status: 0

```
- Database:model (dbid: 3)
```

```
sql comman is: declare @ret int select @ret = dol_downgrade_check('model', @toversid)
print "Database model table downgrade status: %1!", @ret if @ret != 0 begin print
"*** Tables in database 'model' cannot be downgraded." print "*** See the server
error log for details." select @exec_error_count = @exec_error_count + 1 end
```

Database model table downgrade status: 0

```
- Database:sybsystemdb (dbid: 31513)
```

```
sql comman is: declare @ret int select @ret = dol_downgrade_check('sybssystemdb',
@toversid) print "Database sybssystemdb table downgrade status: %!", @ret if @ret !=
0 begin print "**** Tables in database 'sybssystemdb' cannot be downgraded." print
"**** See the server error log for details." select @exec_error_count =
@exec_error_count + 1 end
```

Database sybssystemdb table downgrade status: 0

- Database: sybssystemprocs (dbid: 31514)

```
sql comman is: declare @ret int select @ret = dol_downgrade_check('sybssystemprocs',
@toversid) print "Database sybssystemprocs table downgrade status: %!", @ret if @ret
!= 0 begin print "**** Tables in database 'sybssystemprocs' cannot be downgraded."
print "**** See the server error log for details." select @exec_error_count =
@exec_error_count + 1 end
```

Database sybssystemprocs table downgrade status: 0

Executing downgrade step 19 [delete sysattributes where class = 38] for :

- Database:master (dbid:1)

```
sql comman is: delete sysattributes where class = 38
```

Executing downgrade step 20 [delete syscolumns where id = object_id('sysoptions') and name='number'] for :

- Database:master (dbid:1)

```
sql comman is: delete syscolumns where id = object_id('sysoptions') and name='number'
```

Executing downgrade step 21 [delete syscolumns where id = object_id('syssservers') and name = 'srvprincipal'] for :

- Database:master (dbid:1)

```
sql comman is: delete syscolumns where id = object_id('syssservers') and name =
'srvprincipal'
```

Executing downgrade step 22 [delete sysconfigures where config = 503] for :

- Database:master (dbid:1)

```
sql comman is: delete sysconfigures where config = 503
```

Executing downgrade step 23 [delete sysattributes where class = 25 and attribute in (2, 3)] for :

- Database:master (dbid:1)

```
sql comman is: delete sysattributes where class = 25 and attribute in (2, 3)
```

```
Executing downgrade step 24 [update :DBNAME:..sysattributes set object_cinfo2 = null,
object_datetime = null where object_cinfo2 is not null or object_datetime is not null
delete :DBNAME:..syscolumns where id = 21 and name in ('object_cinfo2',
'object_datetime')] for :- Database:master (dbid:1)
```

```
sql comman is: update master..sysattributes set object_cinfo2 = null, object_datetime
= null where object_cinfo2 is not null or object_datetime is not null delete
master..syscolumns where id = 21 and name in ('object_cinfo2', 'object_datetime')
```

```
- Database:tempdb (dbid: 2)
```

```
sql comman is: update tempdb..sysattributes set object_cinfo2 = null, object_datetime
= null where object_cinfo2 is not null or object_datetime is not null delete
tempdb..syscolumns where id = 21 and name in ('object_cinfo2', 'object_datetime')
```

```
- Database: model (dbid: 3)
```

```
sql comman is: update model..sysattributes set object_cinfo2 = null, object_datetime
= null where object_cinfo2 is not null or object_datetime is not null delete
model..syscolumns where id = 21 and name in ('object_cinfo2', 'object_datetime')
```

```
- Database:sybsystemdb (dbid: 31513)
```

```
sql comman is: update sybsystemdb..sysattributes set object_cinfo2 = null,
object_datetime = null where object_cinfo2 is not null or object_datetime is not null
delete sybsystemdb..syscolumns where id = 21 and name in ('object_cinfo2',
'object_datetime')
```

```
- Database:sybsystemprocs (dbid: 31514)
```

```
sql comman is: update sybsystemprocs..sysattributes set object_cinfo2 = null,
object_datetime = null where object_cinfo2 is not null or object_datetime is not null
delete sybsystemprocs..syscolumns where id = 21 and name in ('object_cinfo2',
'object_datetime')
```

```
Executing downgrade step 25 [update :DBNAME:..syscomments set encrkeyid = null where
encrkeyid is not null delete:DBNAME:..syscolumns where id = 6 and name = 'version'
delete :DBNAME:..syscolumns where id = 6 and name = 'encrkeyid'] for :
```

```
- Database:master (dbid:1)
```

```
sql comman is: update master..syscomments set encrkeyid = null where encrkeyid is not
null delete master..syscolumns where id = 6 and name = 'version' delete
master..syscolumns where id = 6 and name = 'encrkeyid'
```

```
- Database:tempdb (dbid: 2)
```

```
sql comman is: update tempdb..syscomments set encrkeyid = null where encrkeyid is not
null delete tempdb..syscolumns where id = 6 and name = 'version' delete
tempdb..syscolumns where id = 6 and name = 'encrkeyid'
```

```
- Database:model (dbid: 3)
```

```
sql comman is: update model..syscomments set encrkeyid = null where encrkeyid is not
null delete model..syscolumns where id = 6 and name = 'version' delete
model..syscolumns where id = 6 and name = 'encrkeyid'
```

```
- Database:sybsystemdb (dbid: 31513)
```

```
sql comman is: update sybsystemdb..syscomments set encrkeyid = null where encrkeyid
is not null delete sybsystemdb..syscolumns where id = 6 and name = 'version' delete
sybsystemdb..syscolumns where id = 6 and name = 'encrkeyid'
```

```
- Database:sybsystemprocs (dbid: 31514)
sql comman is: update sybsystemprocs..syscomments set encrkeyid = null where encrkeyid
is not null delete sybsystemprocs..syscolumns where id = 6 and name = 'version' delete
sybsystemprocs..syscolumns where id = 6 and name = 'encrkeyid'
```

```
Executing downgrade step 26 [delete statistics sysssrvroles(password) if exists
(select 1 from sysssrvroles where password is not null) begin print "Truncating
password and locking following role(s)" select name from sysssrvroles where password
is not null update sysssrvroles set password = null, status = (status | @lockrole)
```

```
where password is not null end update syscolumns set length = 30 where id =
object_id('sysssrvroles') and name = 'password' update sysssrvroles set locksuid = null,
lockreason = null, lockdate = null where locksuid is not null or lockreason is not
null or lockdate is not null delete syscolumns where id = object_id('sysssrvroles')
and name in ('locksuid', 'lockreason', 'lockdate')] for :
```

```
- Database:master (dbid:1)
sql comman is: delete statistics sysssrvroles(password) if exists (select 1 from
sysssrvroles where password is not null) begin print "Truncating password and locking
following role(s)" select name from sysssrvroles where password is not null update
sysssrvroles set password = null, status = (status | @lockrole) where password is not
null end update syscolumns set length = 30 where id = object_id('sysssrvroles') and
name = 'password' update sysssrvroles set locksuid = null, lockreason = null, lockdate
= null where locksuid is not null or lockreason is not null or lockdate is not null
delete syscolumns where id = object_id('sysssrvroles') and name in ('locksuid',
'lockreason', 'lockdate')
```

Truncating password and locking following role(s)

```
name
-----
doctor_role
```

```
Executing downgrade step 27 [delete sysattributes where class = 35 delete
sysattributes where class = 39 update syslogins set lpid = null, crsuid = null where
lpid is not null or crsuid is not null delete syscolumns where id =
object_id('syslogins') and name in ('lpid', 'crsuid') delete syslogins where (status
& @lp_status) = @lp_status update syslogins set status = status & ~(@exempt_lock)
where (status & @exempt_lock) = @exempt_lock] for :
```

```
- Database:master (dbid:1)
sql comman is: delete sysattributes where class = 35 delete sysattributes where class
= 39 update syslogins set lpid = null, crsuid = null where lpid is not null or crsuid
is not null delete syscolumns where id = object_id('syslogins') and name in ('lpid',
'crsuid') delete syslogins where (status & @lp_status) = @lp_status update syslogins
set status = status & ~(@exempt_lock) where (status & @exempt_lock) = @exempt_lock
```

```
Executing downgrade step 998 [declare @d int, @stat4 int select @stat4=convert(int,
dbinfo_get('master','status4')) select @d=dbinfo_update(1, 'status4', 32 | @stat4)]
for :
```

```
- Database:master (dbid:1)
```

```
sql comman is: declare @d int, @stat4 int select @stat4=convert(int,
dbinfo_get('master','status4')) select @d=dbinfo_update(1, 'status4', 32 | @stat4)
```

```
Executing downgrade step 999 [declare @d int select @d=dbinfo_update(@dbid,
'ASEvers', 15500)] for :
- Database:master (dbid:1)
sql comman is: declare @d int select @d=dbinfo_update(@dbid, 'ASEvers', 15500)

- Database:tempdb (dbid: 2)
sql comman is: declare @d int select @d=dbinfo_update(@dbid, 'ASEvers', 15500)

- Database: model (dbid: 3)
sql comman is: declare @d int select @d=dbinfo_update(@dbid, 'ASEvers', 15500)

- Database:sybsystemdb (dbid: 31513)
sql comman is: declare @d int select @d=dbinfo_update(@dbid, 'ASEvers', 15500)

- Database:sybsystemprocs (dbid: 31514)
sql comman is: declare @d int select @d=dbinfo_update(@dbid, 'ASEvers', 15500)

(return status = 0)
```

使用法

以前にインストールした Adaptive Server 15.0.x リリースに戻す場合に使用します。この時点では、Adaptive Server 15.0.2 に戻すことはできません。

役割パスワードと sp_downgrade

15.7 より前のバージョンの Adaptive Server にダウングレードする場合、sp_downgrade システム・プロシージャによって、sp_passwordpolicy [prepare | downgrade] とダウングレードする Adaptive Server のバージョン番号が内部的に呼び出されます。

sp_downgrade を実行すると、Adaptive Server は次のタスクを実行します。

- 役割パスワードのトランケートと役割のロック。
- クラス 35 未満の sysattributes に新しく追加された属性の削除。
- sysattributes に新しく追加されたクラス 35 の削除。

- `sysssrvroles` からの新しい `locksuid` カラム、`lockreason` カラム、および `lockdate` カラムの削除。

注意 Adaptive Server をバージョン 15.7 から 15.0.2 より前のバージョンにダウングレードすると、役割とログインの両方のパスワードがダウングレードされます。ただし、バージョン 15.0.2 にダウングレードすると、Adaptive Server は役割パスワードのみをトランケートして、ロックします。

役割パスワードのダウングレード方法の詳細については、使用しているプラットフォームの『インストール・ガイド』のダウングレードの項を参照してください。

パーミッション

`sp_downgrade` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>sa_serverprivs_role</code> と <code>sybase_ts_role</code> を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sso_role</code> 、 <code>sa_role</code> 、および <code>sybase_ts_role</code> を持つユーザであることが必要。

sp_dropalias

説明 `sp_dropalias` によって設定されたエイリアス・ユーザ名 ID を削除します。

構文 `sp_dropalias loginame [, force]`

パラメータ `loginame`
別のユーザにエイリアスされたユーザの名前 (master.dbo.syslogins 内にある名前) です。

force
データベース・オブジェクトを所有するエイリアスも削除できるようにします。

例 **例 1** “victoria” が現在のデータベースで (たとえばデータベース所有者の) エイリアスである場合、この文は “victoria” を、エイリアスを割り当てられたユーザとしてデータベースから削除します。

```
sp_dropalias victoria
```

例 2 プロシージャ `namelist` を所有するエイリアス “harry” を削除します。エイリアスが Adaptive Server によって削除されますが、次の警告メッセージが発行されます。

```
sp_dropalias harry, force
Warning: You have forced the drop of the alias for login
'harry' which owns objects in the database. This may
result in errors when those objects are accessed from
or contain references to another database.
Alias user dropped.
```

```
(return status = 0)
```

使用法

- `sp_dropalias` プロシージャを実行すると、`sysalternates` テーブルから、ユーザの代替 `suid` マッピングが削除されます。
- ユーザのエイリアスが削除されると、そのユーザは、エイリアスが作成されていたデータベースにアクセスできなくなります。
- データベース内でオブジェクトを所有しているユーザのエイリアスも削除できます。ログインを削除する前にオブジェクトを削除する必要はありません。

パーミッション `sp_dropalias` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any user</code> 権限を持つユーザであることが必要。
---------------	--

細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者、 <code>sso_role</code> を持つユーザ、または <code>sa_role</code> を持つユーザであることが必要。
---------------	---

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_addalias](#)、[sp_adduser](#)、[sp_droplogin](#)、[sp_dropuser](#)、[sp_helpuser](#)

sp_drop_all_qplans

説明	抽象プラン・グループ内のすべての抽象プランを削除します。
構文	<code>sp_drop_all_qplans name</code>
パラメータ	<i>name</i> すべてのプランが削除される抽象プラン・グループの名前です。
例	<code>sp_drop_all_qplans dev_test</code>
使用法	<ul style="list-style-type: none"> 個々のプランを削除するには、sp_drop_qplan を使用します。 現在のデータベースにある抽象プラン・グループの名前を表示するには、sp_help_qpgroup を使用します。 <code>sp_drop_all_qplans</code> は、システム管理者またはデータベース所有者によって実行されると、指定のユーザに属するグループ内のすべてのプラン、またはグループ内のすべてのプランを暗黙的に削除します。

パーミッション `sp_drop_all_qplans` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザであることが必要。 どのユーザも <code>sp_drop_all_qplans</code> を実行して各自のプランを削除できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。 どのユーザも <code>sp_drop_all_qplans</code> を実行して各自のプランを削除できる。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — すべての入力パラメータ <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_drop_qplan](#)、[sp_drop_qpgroup](#)

sp_drop_qpgroup

説明	抽象プラン・グループを削除します。		
構文	<code>sp_drop_qpgroup group</code>		
パラメータ	<i>group</i> 削除する抽象プラン・グループの名前です。		
例	抽象プラン・グループ “dev_test” を削除します。 <code>sp_drop_qpgroup dev_test</code>		
使用法	<ul style="list-style-type: none"> デフォルト・グループの <code>ap_stdin</code> と <code>ap_stdout</code> は削除できません。 プランが含まれているグループを削除することはできません。グループ内のプランをすべて削除するには、sp_drop_all_qpplans を使用します。グループのリストとグループに格納されているプランの数を表示するには、sp_help_qpgroup を使用します。 <code>sp_drop_qpgroup</code> は、トランザクション内では実行できません。 		
パーミッション	<code>sp_drop_qpgroup</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。		
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザであることが必要。		
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。		
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。		
		監査対象のコマンドまたはアクセス	extrainfo の情報
イベント	監査オプション		
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名
参照	システム・プロシージャ sp_drop_all_qpplans 、 sp_help_qpgroup		

sp_drop_qplan

説明	抽象プランを削除します。
構文	<code>sp_drop_qplan id</code>
パラメータ	<i>id</i> 削除する抽象プランの ID です。
例	指定した ID の抽象プランが削除されます。 <code>sp_drop_qplan 1760009301</code>
使用法	<ul style="list-style-type: none"> プランの ID を検索するには、sp_help_qpgroup、sp_help_qplan、または sp_find_qplan を使用します。また、プラン ID は <code>create plan</code> によって返され、<code>showplan</code> の出力に含まれます。 グループ内の抽象プランをすべて削除するには、sp_drop_all_qplans を使用します。
パーミッション	<code>sp_drop_qplans</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザであることが必要。 どのユーザも <code>sp_drop_qplans</code> を実行して各自のプランを削除できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。 どのユーザも <code>sp_drop_qplans</code> を実行して各自のプランを削除できる。
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — すべての入力パラメータ <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照	<p>コマンド create plan</p> <p>システム・プロシージャ sp_drop_all_qplans、sp_find_qplan、sp_help_qpgroup、sp_help_qplan</p>
----	---

sp_drop_resource_limit

説明 Adaptive Server から 1 つ以上のリソースの制限を削除します。

構文

```
sp_drop_resource_limit { name, appname }  
                    [, rangename, limittype, enforced, action, scope]
```

パラメータ

name

制限が適用される Adaptive Server ログインです。特定のアプリケーションのすべてのユーザに適用されているリソースの制限を削除するには、*appname* と NULL の *name* を指定します。

appname

制限が適用されるアプリケーションです。指定したログインで使用されるすべてのアプリケーションに適用されているリソースの制限を削除するには、ログイン名と NULL の *appname* を指定します。特定のアプリケーションに適用される制限を削除するには、クライアント・プログラムがログイン・パケット内で Adaptive Server に渡すアプリケーション名を指定します。

rangename

制限が実施される時間範囲です。*systimeranges* システム・テーブルに保管されている既存の時間範囲を指定します。または、*rangename* に関係なく、指定した *name*、*appname*、*limittype*、*action*、*scope* に対応するすべてのリソース制限を削除する場合は NULL を指定してください。

limittype

制限が設定されているリソースのタイプです。これは、次のいずれかでなければなりません。

- **row_count** — クエリで返せるローの数を限定する制限だけを削除します。
- **elapsed_time** — クエリ・バッチまたはトランザクションを実行できる秒数を限定する制限だけを削除します。
- **io_cost** — 実際のクエリ処理コストまたは見積もりのクエリ処理コストを限定する制限だけを削除します。
- **tempdb_space** — 1 つのセッションで使用または保持できる tempdb データベースのページ数の制限だけを削除します。
- **NULL** — 指定した *name*、*appname*、*rangename*、施行時間、*action*、*scope* によるすべてのリソース制限を、*limittype* に関係なく削除します。

enforced

クエリの実行前、または実行中に制限を実施するかどうかを指定します。次の表に、制限タイプ別の有効値を示します。

実施コード	説明	制限タイプ
1	実行の見積もりコストが指定の制限値を超えたときに実行される動作に対する制限だけを削除する。	io_cost
2	実際のロー数、経過時間、実行のコストが指定の制限値を超えたときに実行される動作に対する制限だけを削除する。	row_count elapsed_time io_cost
3	見積もりコスト (1) または実際のコスト (2) のどちらかが指定の制限値を超えたときに実行される動作に対する制限だけを削除する。	io_cost
NULL	指定された <i>name</i> 、 <i>appname</i> 、 <i>rangename</i> 、 <i>limittype</i> 、 <i>scope</i> によるすべてのリソース制限を、 <i>action</i> が実施される時間には関係なく削除する。	

action

制限値を超過したときに実行される動作です。これは、次のいずれかであればなりません。

動作コード	説明
1	警告を出す制限だけを削除する。
2	クエリ・バッチをアボートする制限だけを削除する。
3	トランザクションをアボートする制限だけを削除する。
4	セッションを強制終了する制限だけを削除する。
NULL	指定した <i>name</i> 、 <i>appname</i> 、 <i>rangename</i> 、 <i>limittype</i> 、 <i>実行時間</i> 、 <i>scope</i> によるすべてのリソース制限を、実行される動作 (<i>action</i>) には関係なく削除する。

scope

制限が適用されるスコープです。これは、次のいずれかであればなりません。

スコープのコード	説明
1	クエリに使用する制限だけを削除する。
2	クエリ・バッチに適用される制限だけを削除する。
4	トランザクションに適用される制限だけを削除する。
6	クエリ・バッチとトランザクションの両方に適用される制限だけを削除する。
NULL	指定した <i>name</i> 、 <i>appname</i> 、 <i>rangename</i> 、 <i>limittype</i> 、 <i>実行時間</i> 、 <i>action</i> によるすべてのリソース制限を、そのスコープ (<i>scope</i>) には関係なく削除する。

例 **例 1** 過剰な実行時間 I/O コストをもたらす *friday_afternoon* 時間範囲中に *payroll* アプリケーションを使用してクエリを実行したときに必ず、セッションを強制終了するリソース制限を 1 つだけ削除します。

```
sp_drop_resource_limit joe, payroll, friday_afternoon, io_cost, 2, 4, 1
```

注意 リソースの制限値がこれらの選択基準に一致しない場合、`sp_drop_resource_limit` はエラーなしで返されます。

例 2 `joe` が使用する *payroll* アプリケーションに適用されるすべての制限を削除します。

```
sp_drop_resource_limit joe, payroll
```

例 3 ユーザ “`joe`” に適用されるすべての制限を削除します。

```
sp_drop_resource_limit joe
```

例 4 *payroll* アプリケーションに適用されるすべてのリソース制限を削除します。

```
sp_drop_resource_limit NULL, payroll
```

例 5 *payroll* アプリケーションに対してセッションの強制終了動作を実行する、すべてのリソース制限を削除します。

```
sp_drop_resource_limit NULL, payroll, NULL, NULL, NULL, 4, NULL
```

使用法

- どのリソース制限をユーザ、アプリケーション、または日時に適用するかを決めるには、`sp_help_resource_limit` システム・プロシージャを使用します。
- Adaptive Server のログインを削除するために `sp_droplogin` を使用すると、そのログインに対応するすべてのリソース制限も削除されます。
- リソースの制限を削除すると、そのログインとアプリケーションまたはいずれかに対する各セッションの制限値が、そのセッションで次回実行されるクエリ・バッチの最初の部分に再度バインドされます。

パーミッション

`sp_drop_resource_limit` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage resource limit</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 **マニュアル** リソース制限の詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_add_resource_limit](#)、[sp_droplogin](#)、[sp_help_resource_limit](#)、[sp_modify_resource_limit](#)

sp_drop_time_range

説明 Adaptive Server からユーザ定義の時間範囲を削除します。

構文 `sp_drop_time_range name`

パラメータ *name*
削除される時間範囲の名前です。

例 “evenings” という時間範囲を削除します。

```
sp_drop_time_range evenings
```

使用法

- “at all times” 時間範囲は削除できません。
- 時間範囲にリソースの制限値が設定されている場合は、時間範囲を削除できません。
- 時間範囲を削除しても、現在進行しているセッションのアクティブな時間範囲への影響はありません。

パーミッション `sp_drop_time_range` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage resource limit</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 マニュアル 時間範囲の詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_add_resource_limit](#)、[sp_add_time_range](#)、[sp_modify_time_range](#)

sp_dropdevice

説明	Adaptive Server のデータベース・デバイスまたはダンプ・デバイスを削除します。
構文	<code>sp_dropdevice logicalname</code>
パラメータ	<i>logicalname</i> master.dbo.sysdevices.name 内にリストされているデバイスの名前です。
例	<p>例 1 tape5 という名前のデバイスを Adaptive Server から削除します。</p> <pre>sp_dropdevice tape5</pre> <p>例 2 fredsdta という名前のデータベース・デバイスを Adaptive Server から削除します。削除されるデバイスは、どのデータベースでも使用していないデバイスでなければなりません。</p> <pre>sp_dropdevice fredsdta</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_dropdevice</code> プロシージャは、Adaptive Server からデバイスを削除して、そのデバイスのエントリを <code>master.dbo.sysdevices</code> から消去します。 • <code>sp_dropdevice</code> は、データベース・デバイスとして削除されるファイルを消去しません。そのファイルを Adaptive Server からアクセスできないようにするだけです。<code>sp_dropdevice</code> を使用した後でファイルを消去するには、オペレーティング・システムのコマンドを使用してください。
パーミッション	<code>sp_dropdevice</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage disk</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 [コマンド drop database](#)

システム・プロシージャ [sp_addumpdevice](#)、[sp_helpdb](#)、[sp_helpdevice](#)

sp_dropengine

説明	sp_dropengine は、スレッド・モードでは動作しません。
プロセス・モードの考慮事項	指定したエンジン・グループからエンジンを削除します。そのエンジンがグループ内の最後のエンジンである場合は、そのエンジン・グループを削除します。
構文	sp_dropengine engine_number [,engine_group] [,instance_id]
パラメータ	<p>engine_number グループから削除するエンジンの番号です。番号は、0 と最大値の間で、最大値は設定されたオンライン・エンジン数から 1 を引いた値です。</p> <p>engine_group エンジンを削除するエンジン・グループの名前です。</p> <p>instance_id (クラスタ環境のみ) エンジンまたはエンジン・グループを削除するインスタンスの ID です。</p>
例	<p>例 1 DS_GROUP というグループからエンジン番号 2 を削除します。このエンジンがグループ内の最後のエンジンである場合は、そのグループも削除されます。</p> <pre>sp_dropengine 2, DS_GROUP</pre> <p>例 2 (クラスタ環境のみ) インスタンス ID 8 からエンジン番号 5 を削除します。</p> <pre>sp_dropengine 5, 8</pre>
使用法	<ul style="list-style-type: none"> • sp_dropengine は、master データベースからしか呼び出せません。 • engine_number が engine_group 内の最後のエンジンであれば、Adaptive Server は engine_group も削除します。 • クラスタ環境では、sp_cluster set system_view が cluster に設定されている場合、クラスタ内のインスタンスからエンジンまたはエンジン・グループを削除できます。system_view がインスタンスに設定されている場合、ローカル・インスタンスからのみエンジンまたはエンジン・グループを削除できます。 • sp_dropengine は、sp_procxmode を使用してトランザクション・モードを anymode に変更した後に、連鎖トランザクションを使用するセッションで実行できます。 • 指定する engine_number が engine_group 内に存在していなければなりません。

パーミッション sp_dropengine のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage any execution class 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addengine](#)

sp_dropexeclass

説明 ユーザ定義の実行クラスを削除します。

構文 `sp_dropexeclass classname`

パラメータ `classname`
削除されるユーザ定義の実行クラスの名前です。

例 ユーザ定義実行クラス `DECISION` を削除します。

```
sp_dropexeclass 'DECISION'
```

使用法

- 実行クラスは、Adaptive Server がタスクの処理で使用する実行優先度を定義するとき便利です。実行クラスと実行属性の詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。
- `classname` を、どのクライアント・アプリケーション、ログイン、ストアド・プロシージャ、デフォルトの実行クラスにもバインドしないでください。最初に `sp_unbindexeclass` を使用して実行クラスのバインドを解除し、次に `sp_dropexeclass` を使用して実行クラスを削除します。
- システム定義実行クラスは削除できません。

パーミッション `sp_dropexeclass` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any execution class</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ `sp_addexeclass`、`sp_bindexeclass`、`sp_showexeclass`、`sp_unbindexeclass`

sp_dropextendedproc

説明	ESP (拡張ストアド・プロシージャ) を削除します。
構文	<code>sp_dropextendedproc esp_name</code>
パラメータ	<code>esp_name</code> 削除される拡張ストアド・プロシージャの名前です。
例	<code>xp_echo</code> を削除します。 <code>sp_dropextendedproc xp_echo</code>
使用法	<ul style="list-style-type: none"> <code>sp_dropextendedproc</code> は、master データベースから実行してください。 <code>esp_name</code> では大文字と小文字を区別します。ESP が作成されたときの名前と正確に一致させてください。
パーミッション	<code>sp_dropextendedproc</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any ESP</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド drop procedure](#)
[システム・プロシージャ sp_addextendedproc](#)、[sp_freelld](#)、[sp_helpextendedproc](#)

sp_dropexternlogin

説明	(コンポーネント統合サービスのみ) sp_addexternlogin によってあらかじめ定義されているリモート・ログインの定義を削除します。
構文	<code>sp_dropexternlogin server [, loginame [, rolename]]</code>
パラメータ	<p>server ローカル・サーバがアカウント・アクセスを削除するリモート・サーバの名前です。リモート・サーバは、<code>master.dbo.sys.servers</code> テーブルのエントリによって、ローカル・サーバに認識されていなければなりません。</p> <p>loginame ローカル・サーバに認識されるログイン・アカウントです。 <code>loginame</code> が指定されていない場合は、現在のアカウントが使用されます。<code>loginame</code> は、<code>master.dbo.syslogins</code> テーブルに存在していなければなりません。</p> <p>rolename <code>Adaptive Server</code> のユーザに割り当てられた役割です。</p>
例	<p>例 1 リモート・サーバ CIS1012 への外部ログインの定義を “bobj” から削除します。“bobj” に対するリモート・ログインを追加または修正できるのは、“bobj” アカウントと “sa” アカウントだけです。</p> <pre>sp_dropexternlogin CIS1012, bobj</pre> <p>例 2 リモート・サーバ SSB への外部ログインの定義を、<code>sa_role</code> を持つユーザから削除します。</p> <pre>sp_dropexternlogin SSB, NULL, sa_role</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_dropexternlogin</code> は、sp_addexternlogin によってローカル・サーバに対してあらかじめ定義されているリモート・ログインの定義を削除します。 • <code>sp_dropexternlogin</code> を、トランザクション内から実行することはできません。 • sp_addserver を使用して、リモート・サーバをローカル・サーバに対して定義してください。 • ローカル・サーバ・ユーザを追加または削除するには、システム・プロシージャの sp_addalias と sp_droplogin を使用します。

パーミッション `sp_dropexternlogin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効
 細密なパーミッションが有効の場合、`manage any remote login` 権限を持つユーザであることが必要。
`sp_dropexternlogin` は、`loginname` にバインドされたユーザが実行できる。

細密なパーミッションが無効
 細密なパーミッションが無効の場合、`sa_role` が付与されたユーザであることが必要。
`sp_dropexternlogin` は、`loginname` にバインドされたユーザが実行できる。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ `sp_addexternlogin`、`sp_helpexternlogin`、`sp_addlogin`、`sp_droplogin`

sp_dropglockpromote

説明	テーブルまたはデータベースからロック・プロモーション値 (ロックの拡大値) を削除します。
構文	<code>sp_dropglockpromote {"database" "table"}, objname</code>
パラメータ	<p><code>database table</code> ロック・プロモーション・スレッシュホールドをデータベースから削除するか、テーブルから削除するかを指定します。ロック・プロモーション・スレッシュホールドは Transact-SQL キーワードであるため、引用符で囲みます。</p> <p><code>objname</code> ロック・プロモーション・スレッシュホールドを削除するテーブルまたはデータベースの名前です。</p>
例	<p><code>titles</code> からロック・プロモーション値 (ロックの拡大値) を削除します。<code>titles</code> のロック・プロモーション (ロックの拡大) が、データベースワイドまたはサーバワイドな値を使用するようになります。</p> <pre>sp_dropglockpromote "table", titles</pre>
使用法	<ul style="list-style-type: none"> • sp_setpglockpromote によって設定されたロック・プロモーション値を削除するには、sp_dropglockpromote を使用します。 • データベースのロック・プロモーション・スレッシュホールドを削除すると、ロック・プロモーション・スレッシュホールドが設定されていないテーブルでは、サーバワイドな値が使用されます。 • テーブルの値が削除されたときに、Adaptive Server は、データベースのロック・プロモーション・スレッシュホールドが設定されていればその値を使用し、設定されていない場合にはサーバワイドな値を使用します。 • サーバワイドな値は sp_setpglockpromote によって変更できますが、削除はできません。
パーミッション	<code>sp_dropglockpromote</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage lock promotion threshold</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの 実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_setpglockpromote](#)

sp_dropgroup

説明	データベースからグループを削除します。
構文	<code>sp_dropgroup grpname</code>
パラメータ	<code>grpname</code> 現在のデータベース内にあるグループの名前です。
例	<p>“purchasing”グループは“accounting”グループとマージされています。次のコマンドは、グループを削除する前に、“purchasing”グループのメンバである“martha”と“george”を他のグループに移動します。“public”というグループ名は予約語であるため、引用符で囲まれています。</p> <pre>sp_changegroup accounting, martha sp_changegroup "public", george sp_dropgroup purchasing</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_dropgroup</code> を実行すると、データベースの <code>sysusers</code> テーブルからグループ名が削除されます。 • メンバを持っているグループは削除できません。各メンバに対して <code>sp_changegroup</code> を実行してからグループを削除してください。
パーミッション	<code>sp_dropgroup</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any user</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者、 <code>sa_role</code> を持つユーザ、または <code>sso_role</code> を持つユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addgroup](#)、[sp_changegroup](#)、[sp_helpgroup](#)

sp_dropkey

説明	<code>sp_primarykey</code> 、 <code>sp_foreignkey</code> 、または <code>sp_commonkey</code> を使用して定義されていたキーを <code>syskeys</code> テーブルから削除します。
構文	<code>sp_dropkey keytype, tablename [, deptabname]</code>
パラメータ	<p><i>keytype</i> 削除されるキーのタイプです。 <i>keytype</i> は <code>primary</code>、<code>foreign</code>、または <code>common</code> でなければなりません。</p> <p><i>tablename</i> 削除されるキーが入っているキー・テーブルまたはビューの名前です。</p> <p><i>deptabname</i> <i>keytype</i> が <code>foreign</code> または <code>common</code> である場合は、関連するテーブルから 2 番目のテーブルの名前を指定します。 <i>keytype</i> が <code>primary</code> である場合、このパラメータは不要です。 <code>primary</code> キーは従属テーブルを持っていないからです。 <i>keytype</i> が <code>foreign</code> の場合、これはプライマリ・キー・テーブルの名前です。 <i>keytype</i> が <code>common</code> の場合は、<code>sp_helpkey</code> によって表示される順序で 2 つのテーブル名を指定します。</p>
例	<p>例 1 <code>employees</code> テーブルのプライマリ・キーを削除します。 <code>employees</code> 用のプライマリ・キーに依存していた外部キーもすべて削除されます。</p> <pre>sp_dropkey primary, employees</pre> <p>例 2 <code>employees</code> テーブルと <code>projects</code> テーブルの間の共通キーを削除します。</p> <pre>sp_dropkey common, employees, projects</pre> <p>例 3 <code>titleauthor</code> テーブルと <code>titles</code> テーブルの間の外部キーを削除します。</p> <pre>sp_dropkey foreign, titleauthor, titles</pre>
使用法	<ul style="list-style-type: none">• <code>sp_dropkey</code> を実行すると、指定されたキーが <code>syskeys</code> から削除されます。 キーをテーブルから削除できるのは、そのテーブルの所有者だけです。• キーは、データベース設計での暗黙的な論理関係を明示的にするために作成されます。 この情報は、アプリケーションで使用できません。• プライマリ・キーを削除すると、それに対応するすべての外部キーが自動的に削除されます。 外部キーを削除しても、そのテーブル上で指定されているプライマリ・キーには影響を与えません。

- [sp_commonkey](#)、[sp_primarykey](#)、または [sp_foreignkey](#) を実行すると、`syskeys` システム・テーブルにキーが追加されます。定義されているキーについてのレポートを表示するには、[sp_helpkey](#) を実行します。

パーミッション

`sp_dropkey` を実行するには、テーブルの所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_commonkey](#)、[sp_foreignkey](#)、[sp_helpkey](#)、[sp_primarykey](#)

sp_droplanguage

説明	サーバから代替言語を削除し、代替言語のローを <code>master.dbo.syslanguages</code> から消去します。
構文	<code>sp_droplanguage language [, dropmessages]</code>
パラメータ	<p><i>language</i> 削除する言語の公式名です。</p> <p><i>dropmessages</i> <i>language</i> にある Adaptive Server のシステム・メッセージをすべて削除します。Adaptive Server のシステム・メッセージも削除しないと、対応しているシステム・メッセージを持つ言語を削除することはできません。</p>
例	<p>例 1 対応するメッセージがない場合に、使用可能な代替言語からフランス語を削除します。</p> <pre>sp_droplanguage french</pre> <p>例 2 対応するメッセージがある場合に、使用可能な代替言語からフランス語を削除します。</p> <pre>sp_droplanguage french, dropmessages</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_droplanguage</code> を実行すると、言語のエントリが <code>master.dbo.syslanguages</code> テーブルから消去されます。これによって代替言語のリストからその言語が削除されます。 • システム・メッセージを持つ言語の削除を試みても、<code>dropmessages</code> パラメータが指定されていなければ、削除の要求は実行されません。
パーミッション	<code>sp_droplanguage</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addlanguage](#)、[sp_helplanguage](#)

sp_droplogin

説明	Adaptive Server 15.7 以降では、このシステム・プロシージャは推奨されていません。Adapter Server のログイン・アカウントを削除するには、drop login コマンドを使用します。
構文	なし
パラメータ	なし
使用法	なし

sp_dropmessage

説明	ユーザ定義メッセージを <code>sysusermessages</code> から削除します。
構文	<code>sp_dropmessage message_num [, language]</code>
パラメータ	<p><i>message_num</i> 削除されるメッセージのメッセージ番号です。メッセージ番号は、20000 以上の値でなければなりません。</p> <p><i>language</i> 削除されるメッセージの言語です。</p>
例	<p><code>sysusermessages</code> から、フランス語バージョンの 20002 番のメッセージを削除します。</p> <pre>sp_dropmessage 20002, french</pre>
使用法	<ul style="list-style-type: none"> <i>language</i> パラメータは省略可能です。このパラメータを入力した場合は、指示した言語の指定メッセージ番号 (<i>message_num</i>) が付いているメッセージだけが削除されます。<i>language</i> を指定しなければ、指定メッセージ番号 (<i>message_num</i>) が付いているすべてのメッセージが削除されます。
パーミッション	<code>sp_dropmessage</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、メッセージを作成したユーザ、データベース所有者、または現在のデータベースに対して <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、メッセージを作成したユーザ、データベース所有者、または <code>sa_role</code> を持つユーザであることが必要。
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — すべての入力パラメータ <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addmessage](#)、[sp_getmessage](#)

sp_dropobjectdef

説明 (コンポーネント統合サービスのみ) ローカル・オブジェクトに関する外部の記憶マッピングを削除します。

構文 `sp_dropobjectdef tablename`

パラメータ

tablename

tablename は、`dbname.owner.object` という形式をとります。ここで、

- *dbname* は、削除する記憶ロケーションがあるオブジェクトを格納しているデータベースの名前です。*dbname* は省略可能ですが、これを指定する場合は、現在のデータベースの名前にしてください。また、*owner* またはプレースホルダが必要です。
- *owner* は、削除する記憶ロケーションがあるオブジェクトの所有者の名前です。*owner* は省略可能ですが、*dbname* が指定されている場合は、*owner* が必要です。
- *object* は、外部記憶マッピングが削除されるローカル・テーブルの名前です。

例 **例 1** `personnel` データベース内で `colleges` としてサーバに認識されているテーブルに外部記憶マッピングを提供している `sysattributes` から、エントリを削除します。

```
sp_dropobjectdef "personnel.dbo.colleges"
```

例 2 `andrea.fishbone` オブジェクトに外部記憶マッピングを提供している `sysattributes` から、エントリを削除します。ここで、`andrea` は `owner` で、ローカル・テーブル名は `fishbone` です。

```
sp_dropobjectdef "andrea.fishbone"
```

使用法

- `sp_dropobjectdef` は、ローカル・オブジェクトに提供された外部記憶マッピングを削除します。これは、`sp_droptabledef` を置き換えます。
- `drop table` でリモート・テーブルを削除してから、`sp_dropobjectdef` を使用してください。
- `sysattributes` テーブルからのマッピング情報が `sp_addobjectdef` によって追加されている場合、テーブルを削除してもマッピング情報は消去されません。これは、`sp_dropobjectdef` を使用して明示的に消去する必要があります。
- *tablename* は、次のいずれかの形式をとることができます。
 - *object*
 - *owner.object*

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

パーミッション `sp_dropobjectdef` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、オブジェクト所有者であるか <code>drop any table</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、オブジェクト所有者、データベース所有者、または <code>sa_role</code> を持つユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド create existing table、create table、drop table](#)
 システム・プロシージャ [sp_addobjectdef](#)

sp_dropremotelogin

説明	リモート・ユーザ・ログインを削除します。
構文	<code>sp_dropremotelogin remoteserver [, loginame [, remotename]]</code>
パラメータ	<p><i>remoteserver</i> 削除されるリモート・ログインが割り当てられているサーバの名前です。</p> <p><i>loginame</i> <code>sysremotelogins</code> テーブル内のリモート・サーバに対応しているローカル・サーバのユーザ名です。</p> <p><i>remotename</i> リモート・サーバからログインするときに <i>loginame</i> にマップされるリモート・ユーザ名です。</p>

例 **例 1** GATEWAY という名前のリモート・サーバのエントリを削除します。

```
sp_dropremotelogin GATEWAY
```

例 2 リモート・サーバ GATEWAY からのリモート・ログインを “churchy” という名前のローカル・ユーザにマップするためのエントリを削除します。

```
sp_dropremotelogin GATEWAY, churchy
```

例 3 “churchy” という名前のローカル・ユーザにマップされた、リモート・サーバ GATEWAY 上のリモート・ユーザ “pogo” のログインを削除します。

```
sp_dropremotelogin GATEWAY, churchy, pogo
```

使用法	<ul style="list-style-type: none"> • <code>sp_dropremotelogin</code> を実行すると、リモート・サーバからユーザ・ログインが削除され、ユーザのエントリが <code>master.dbo.sysremotelogins</code> から消去されます。 • リモート・ログインの詳細については、sp_addremotelogin を参照してください。 • ローカル・サーバ・ユーザを追加または削除するには、システム・プロシージャの sp_addlogin と sp_droplogin を使用します。
-----	--

パーミッション `sp_dropremotelogin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any remote login</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addlogin](#)、[sp_addremotelogin](#)、[sp_addserver](#)、[sp_droplogin](#)、[sp_helpremotelogin](#)、[sp_helpserver](#)

sp_drowlockpromote

説明	データベースまたはテーブルから、ロー・ロック・プロモーション・スレッシュールド値を削除します。
構文	<code>sp_drowlockpromote {"database" "table"}, objname</code>
パラメータ	<code>database table</code> ロー・ロック・プロモーション・スレッシュールドをデータベースから削除するか、テーブルから削除するかを指定します。 <code>objname</code> ロー・ロック・プロモーション・スレッシュールドを削除するデータベースまたはテーブルの名前です。
例	<code>sales</code> テーブルからロー・ロック・プロモーション値を削除します。 <code>sales</code> のロック・プロモーションは、現在データベースワイドまたはサーバワイドな値を使用しています。 <pre>sp_drowlockpromote "table", "sales"</pre>
使用法	<ul style="list-style-type: none">• sp_setrowlockpromote によって設定されたロー・ロック・プロモーション値を削除するには、<code>sp_drowlockpromote</code> を使用します。• データベースのロー・ロック・プロモーション・スレッシュールドを削除する場合、設定されたロー・ロック・プロモーション・スレッシュールドを持っていないデータロー・ロック・テーブルはサーバワイドな値を使用します。ロー・ロック・プロモーション設定パラメータの値を確認するには、<code>sp_configure</code> を使用します。• テーブルのロー・ロック・プロモーション値が削除されると、Adaptive Server は、データベースのロー・ロック・プロモーション・スレッシュールドが設定されている場合にはそのスレッシュールドを使用し、データベース用にスレッシュールドが設定されていない場合にはサーバワイドな値を使用します。• データベースに使用されるロック・プロモーション・スレッシュールドを変更するには、master データベースを使用してください。データベース内のテーブルに使用されるロック・プロモーション・スレッシュールドを変更するには、テーブルがあるデータベースを使用してください。• サーバワイドな値は、sp_setrowlockpromote によって変更できます。これによって、ロー・ロック・プロモーション設定パラメータ内の値が変更されます。したがって、<code>sp_drowlockpromote</code> に対応するサーバ・オプションはありません。

パーミッション `sp_droprolockpromote` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage lock promotion threshold</code> 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザーであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_setrowlockpromote](#)

sp_dropsegment

説明	データベースからセグメントを削除するか、特定のデータベース・デバイスからセグメントのマップを解除します。
構文	<code>sp_dropsegment segname, dbname [, device]</code>
パラメータ	<p>segname 削除されるセグメントの名前です。</p> <p>dbname セグメントを削除するデータベースの名前です。</p> <p>device <i>segname</i> セグメントを削除するデータベース・デバイスの名前です。このパラメータは省略可能ですが、データベース・デバイスからシステム・セグメントの <code>system</code>、<code>default</code>、または <code>logsegment</code> が削除される場合は省略できません。</p>
例	<p>例 1 indexes セグメントを <code>pubs2</code> データベースから削除します。</p> <pre>sp_dropsegment indexes, pubs2</pre> <p>例 2 dev1 データベース・デバイスから <code>indexes</code> セグメントのマップを解除します。</p> <pre>sp_dropsegment indexes, pubs2, dev1</pre>
使用法	<ul style="list-style-type: none">セグメントを削除できるのは、指定したデータベース内のテーブル、インデックス、またはパーティションがそのセグメントを参照していない場合です。オプションの引数の <i>device</i> を指定しない場合、セグメントは、指定されたデータベースから削除されます。<i>device</i> 名を指定した場合は、指定されたデータベース・デバイスへのセグメントのマップが解除されますが、セグメントは削除されません。セグメントを削除すると、そのセグメントに対応するすべてのスレッシュホールドが削除されます。

- `sp_dropsegment` は、シングルユーザ・モードで `logsegment` システム・セグメントに対してのみ実行できます。

注意 このコマンドは、非常に大きなデータベースでは、完了までに時間がかかることがあります。

- 1 つ以上のデバイスからセグメントのマップを解除すると、`Adaptive Server` はそのセグメントの合計領域を超えるスレッシュホールドをすべて削除します。1 つ以上のデバイスから `logsegment` のマップを解除すると、`Adaptive Server` はラストチャンス・スレッシュホールドを再計算します。
- `sp_placeobject` は、テーブルまたはインデックスに今後割り当てる領域を、あるセグメントから別のセグメントに変更し、元のセグメントから参照を削除します。`sp_placeobject` を使用した後、`sp_dropsegment` によって元のセグメント名を削除できます。
- システム・セグメントの `system`、`default`、`logsegment` については、セグメントを削除するデバイスの名前を指定してください。

パーミッション

`sp_dropsegment` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ `sp_addsegment`、`sp_addthreshold`、`sp_helpsegment`、`sp_helpthreshold`、`sp_placeobject`

sp_dropserver

説明 登録されているサーバのリストからサーバを削除します。また、関連するリモート・ログインと外部ログインを同時に削除できます。

構文 `sp_dropserver server [, droplogins]`

パラメータ `server`
削除されるサーバの名前です。

`droplogins`
`server` のあらゆるリモート・ログインの削除を指示します。

例 **例 1** リモート・サーバ GATEWAY を削除します。

```
sp_dropserver GATEWAY
```

例 2 リモート・サーバ RDBAM_ALPHA のエントリを削除し、そのサーバのリモート・ログインと外部ログインをすべて削除します。

```
sp_dropserver RDBAM_ALPHA, droplogins
```

使用法

- `sp_dropserver` を実行すると、認識されているサーバのリストからサーバが削除され、`master.dbo.sysservers` テーブルからエントリが消去されます。
- `master.dbo.sysremotelogins` テーブル内に対応するエントリが割り当てられているサーバ上で `sp_dropserver` を実行すると、サーバを削除する前にリモート・ユーザを削除しなければならないことを示すエラー・メッセージが表示されます。サーバを削除するときにそのサーバのすべてのリモート・ログインを削除するには、`droplogins` を使用します。
- `sysattributes` テーブル内に対応するエントリが割り当てられているサーバに対して `droplogins` を指定しないで `sp_dropserver` を実行すると、エラーになります。リモート・ログインと外部ログインを削除してから、サーバを削除してください。
- コンポーネント統合サービスが設定されると、外部ログインとサーバへのデフォルト・マッピングについて、`sysattributes` の確認が行われます。

パーミッション	<p><code>sp_dropserver</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。</p>
細密なパーミッションが有効	<p>細密なパーミッションが有効の場合、<code>manage server</code> 権限を持つユーザであることが必要。</p> <p><code>droplogins</code> を指定する場合は <code>manage any remote login</code> 権限を持つユーザであることが必要。</p> <p>Adaptive Server 高可用性－<code>manage server</code> 権限と <code>ha_role</code> を持つユーザであることが必要。<code>droplogins</code> を指定する場合は <code>manage any remote login</code> 権限を持つユーザであることが必要。</p> <p>Adaptive Server 共有ディスク・クラスター <code>manage server</code> 権限と <code>manage cluster</code> 権限を持つユーザであることが必要。<code>droplogins</code> を指定する場合は <code>manage any remote login</code> 権限を持つユーザであることが必要。</p>
細密なパーミッションが無効	<p>細密なパーミッションが無効の場合、<code>sso_role</code> が付与されたユーザであることが必要。</p> <p>Adaptive Server 高可用性－<code>sso_role</code> パーミッションと <code>ha_role</code> を持つユーザであることが必要。</p> <p>Adaptive Server 共有ディスク・クラスター <code>sso_role</code> と <code>sa_role</code> を持つユーザであることが必要。</p>

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addserver](#)、[sp_dropremotelogin](#)、[sp_helpremotelogin](#)、[sp_helpserver](#)

sp_droptreshold

説明	セグメントから空き領域スレッシュホールドを削除します。
構文	<code>sp_droptreshold dbname, segname, free_space</code>
パラメータ	<p>dbname スレッシュホールドを削除するデータベースです。これは、現在のデータベースの名前でなければなりません。</p> <p>segname 空き領域がスレッシュホールドによってモニタされているセグメントです。デフォルトのセグメント(“default”)を指定するときは、引用符を使用してください。</p> <p>free_space スレッシュホールドを超過した時点での空きページの数です。</p>
例	<p>mydb の segment1 からスレッシュホールドを削除します。スレッシュホールドを識別するには、データベース、セグメント、および空き領域を指定してください。</p> <pre>sp_droptreshold mydb, segment1, 200</pre>
使用法	<ul style="list-style-type: none"> ログ・セグメントからラストチャンス・スレッシュホールドを削除することはできません。 sp_droptreshold の代わりに、sp_dboption の no free space acctg オプションを使用できます。このオプションは、ログ・セグメント以外のセグメントに対する空き領域の計算を無効にします。ログ・セグメントの空き領域の計算を無効にすることはできません。
パーミッション	sp_droptreshold のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、manage database 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か sa_role が付与されたユーザーであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addthreshold](#)、[sp_dboption](#)、[sp_helpthreshold](#)、[sp_thresholdaction](#)

sp_droptype

説明	ユーザ定義データ型を削除します。
構文	<code>sp_droptype <i>typename</i></code>
パラメータ	<i>typename</i> ユーザが所有するユーザ定義データ型の名前です。
例	<code>birthday</code> という名前のユーザ定義データ型を削除します。 <code>sp_droptype birthday</code>
使用法	<ul style="list-style-type: none"> • <code>sp_droptype</code> を実行すると、ユーザ定義データ型が <code>systypes</code> から削除されます。 • テーブルまたはその他のデータベース・オブジェクトがユーザ定義データ型を参照している場合は、そのデータ型を削除することはできません。
パーミッション	<code>sp_droptype</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データ型の所有者が <code>manage database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データ型所有者またはデータベース所有者であることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 [データ型 ユーザ定義型](#)
[システム・プロシージャ `sp_addtype`、`sp_rename`](#)

sp_dropuser

説明	現在のデータベースからユーザを削除します。
構文	<code>sp_dropuser name_in_db</code>
パラメータ	<code>name_in_db</code> 現在のデータベースの <code>sysusers</code> テーブルにあるユーザの名前です。
例	現在のデータベースからユーザ “albert” を削除します。これによって、ユーザ “albert” はデータベースを使用できなくなります。 <code>sp_dropuser albert</code>
使用法	<ul style="list-style-type: none"> • <code>sp_dropuser</code> を実行すると、<code>sysusers</code> からユーザのローが消去されることによって、現在のデータベースからユーザが削除されます。 • 次のものは削除できません。 <ul style="list-style-type: none"> • データベース内でオブジェクトを所有しているユーザ。 • 他のユーザにパーミッションを付与しているユーザ。 • データベース所有者であるユーザ。 • 削除されるユーザに複数のユーザのエイリアスが設定されている場合は、それらのエイリアスも削除されます。これらのユーザはデータベースにアクセスできなくなります。 • そのデータベース内の実行クラスにバインドされているストアド・プロシージャを所有しているユーザを、データベースから削除することはできません。sp_bindexeclss を参照してください。 • <code>sp_dropuser</code> は、現在のデータベース内の指定されたユーザの <code>sysencryptkeys</code> のすべてのキー・コピーを削除します。ユーザがデータベースで暗号化キーを所有している場合、<code>sp_dropuser</code> は失敗します。『暗号化カラム・ユーザ・ガイド』を参照してください。
パーミッション	<code>sp_dropuser</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any user</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データ型の所有者、 <code>sa_role</code> を持つユーザ、または <code>sso_role</code> を持つユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [grant](#)、[revoke](#)、[use](#)

システム・プロシージャ [sp_addalias](#)、[sp_adduser](#)、[sp_bindexclass](#)、[sp_droplogin](#)

sp_dump_history

説明	ダンプ履歴ファイルからレコードを消去できます。元のダンプ履歴ファイルは、ファイル名にタイムスタンプのサフィックスを付けて保存されます。
構文	<pre>sp_dump_history [@operation = {'list' 'purge' 'listfiles' 'help'} [, @until_time = 'date'] [, @database_name = 'database_name'] [, @dump_type = {'DATABASE' 'TRAN[SACTION]' 'CONFIG[URATION]'}] [, @status = {'success' 'fail'}] [, @file = 'filename']</pre>
パラメータ	<p>@operation = {'list' 'purge' 'listfiles' 'help'} 必須のパラメータです。次のいずれかのオプションを指定します。</p> <ul style="list-style-type: none"> • list — ダンプ履歴ファイルのレコードを表示します。デフォルトでは、リストに次の項目が含まれます。 <ul style="list-style-type: none"> • データベース名 - サーバ設定名 • ダンプ・タイプ • ダンプ・ストライプの総数。 • 圧縮レベル • リモート Backup Server 名 • ダンプ作成日 • エラー番号 - Adaptive Server 上のエラー番号 • purge — ダンプ履歴ファイルからレコードを消去します。消去されるレコードは、sp_dump_history の他のパラメータで指定した条件に基づいて選択されます。 • listfiles — ダンプ履歴ファイル名のリストを表示します。 • help — sp_dump_history の構文を表示します。 <p>@until_time = 'date' (オプション) 日付と時刻を指定できます。その時刻より前に作成されたすべてのダンプ・オブジェクトが Backup Server で消去されます。デフォルトでは、すべてのダンプ・レコードが消去されます。</p> <p>@database_name = 'database_name' (オプション) 表示または消去するダンプ履歴ファイルのダンプ・レコードが格納されているデータベースの名前です。デフォルトでは、すべてのデータベースのダンプ・レコードが対象となります。</p>

@dump_type'

(オプション) 表示または消去するダンプ・レコードのタイプを指定します。次の型が有効です：

- 'DATABASE' — dump database で作成されたデータベース・ダンプ・オブジェクト。
- 'TRAN[SACTION]' — dump transaction で作成されたトランザクション・ダンプ・オブジェクト。
- 'CONFIG[URATION]' — dump configuration で作成されたサーバ設定オブジェクト。

デフォルトでは、すべてのタイプのダンプ・レコードが表示または消去されます。

@status = {'success' | 'fail'}

(オプション) 成功か失敗のどちらのダンプ・レコードを表示または消去するかを指定します。デフォルトでは、成功したダンプ・レコードのみが対象となります。

@file = 'filename'

(オプション) ダンプ履歴ファイルの名前です。デフォルトでは、設定されているダンプ履歴ファイル名が使用されます。

例 **例 1** ダンプ履歴ファイルのすべてのダンプ・レコードを表示します。

```
sp_dump_history 'list'
go
```

例 2 指定した時刻より前に作成された、指定したデータベースのダンプ・レコードを表示します。

```
sp_dump_history 'list', "mar 20, 2010 10:51:43:866am",
'testdb'
go
```

使用法 `sp_dump_history` をパラメータなしで実行した場合のデフォルトの動作は、`list` パラメータによる出力を表示します。

データベース・ダンプとトランザクション・ダンプの出力は構成ファイルのものとは異なります。

パーミッション `sp_dump_history` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage dump configuration</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> または <code>oper_role</code> を持つユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
			<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 マニュアル ダンプ・オペレーションの詳細については、『システム管理ガイド』を参照してください。

コマンド dump configuration、dump database、load database

関数

システム・プロシージャ sp_config_dump

sp_dumpoptimize

説明	データベースのダンプ・オペレーション中に Backup Server によってダンプされるデータの量を指定します。
構文	<pre>sp_dumpoptimize ['archive_space = {maximum minimum default }'] sp_dumpoptimize ['reserved_threshold = {nnn default }'] sp_dumpoptimize ['allocation_threshold = {nnn default }']</pre>
パラメータ	<p>archive_space ダンプされるデータベースの量を指定します。</p> <p>maximum 各ページが割り付けられているかどうかを判断しないで、データベース全体をダンプします。アーカイブ・イメージ(1 つまたは複数)が使用する合計領域は、データベースのサイズと同じです。このオプションを使用すると、オプションの <code>reserved_threshold=0</code> と <code>allocation_threshold=0</code> を使用した場合と同じ結果が得られます。</p> <p>minimum 割り付けられたページだけをダンプします。この結果、可能なかぎり小さいサイズのアーカイブ・イメージが作成されます。このオプションは、スループットが QIC テープ・ドライブなどのデータベース・デバイスよりもはるかに少ないアーカイブ・デバイスにダンプするときに便利です。このオプションを使用すると、オプションの <code>reserved_threshold=100</code> と <code>allocation_threshold=100</code> を使用した場合と同じ結果が得られます。</p> <p>default デフォルト値の使用を指定します。以下のオプションと使用します。</p> <ul style="list-style-type: none">• <code>archive_space</code> とともに使用すると、<code>reserved_threshold</code> オプションと <code>allocation_threshold</code> オプションがデフォルト値に設定された状態でデータベースがダンプされます。このオプションは、Backup Server をデフォルトの設定にリセットするために使用します。• <code>reserved_threshold</code> とともに使用すると、<code>default</code> は 85 % を指定します。• <code>allocation_threshold</code> とともに使用すると、<code>default</code> は 40 % を指定します。

reserved_threshold

データベース・ディスク内の予約ページの割合が *nnn* 以上のときは、データベース・ディスク内のそのデータベースに属するすべてのページをダンプします。たとえば、*nnn* に 60 を指定した場合、データベース・ディスクの予約ページの割合が 60 % 以上であると、そのディスク内のどのページが割り付けられているかを判別することなく、ディスク全体がダンプされます。このオプションのデフォルト値は 85 % です。

nnn

スレッシュホールドの値を表わす 0 ~ 100 の間の整数値です。この値を指定して、ダンプするデータの量を決めます。

このオプションを **reserved_threshold** とともに使用すると、ディスク内の予約ページの割合が指定値よりも大きい場合、データベース・ディスク内にあるデータベースのすべてのページがダンプされます。

このオプションを **allocation_threshold** とともに使用すると、アロケーション・ユニット内の割り付けられたページの割合が **allocation_threshold** に指定された割合よりも大きい場合、アロケーション・ユニット内のすべてのページがダンプされます。

allocation_threshold

アロケーション・ユニット内で割り付けられているページの割合が *nnn* 以上である場合、そのユニット内のすべてのページをダンプします。たとえば、*nnn* に 70 が指定されている場合、アロケーション・ユニット内の割り付けられたページの割合が 70 % 以上であると、そのアロケーション・ユニット内のページが割り付けられているかどうかを判別することなく、そのアロケーション・ユニット全体がダンプされます。**reserved_threshold** の設定によってディスク全体がダンプされる場合、そのディスクの **allocation_threshold** 設定は無視されます。このオプションのデフォルト値は 40 % です。

例**例 1** データベース全体がダンプされます。

```
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%.
```

例 2 割り付けられたページだけがダンプされます。したがって、最小のアーカイブ・イメージが作成されます。

```
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%.

例 3 予約スレッシュヨルドが 85 % に設定され、割り付けスレッシュヨルドが 40 % に設定されます。

```
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%.

例 4 予約ページの割合が 60 % 以上のデータベース内のディスクは、このディスク上のアロケーション・ページを読み込むことなくダンプされます。残りのディスクでは、アロケーション・ページが読み込まれ、`allocation_threshold` に最後に設定された値が使用されます。Backup Server の起動後に `allocation_threshold` が設定されていなかった場合は、`allocation_threshold` のデフォルト値 40 % が使用されます。

```
sp_dumpoptimize 'reserved_threshold=60'
```

Backup Server:4.172.1.3:The value of 'reserved pages threshold' has been set to 60%.

例 5 この例では、予約スレッシュヨルドが 85 % に設定されます。アロケーション・ページのスレッシュヨルドへの影響はありません。

```
sp_dumpoptimize 'reserved_threshold=default'
```

Backup Server:4.172.1.3:The value of 'reserved pages threshold' has been set to 85%.

例 6 アロケーション・ページの読み込みは、ディスクの予約ページの割合が、`reserved_threshold` 用に最後に設定された値未満であるディスクに対して行われます。また、アロケーション・ユニットに 80 % 以上のページが割り付けられている場合は、そのアロケーション・ユニット全体がダンプされます。

```
sp_dumpoptimize 'allocation_threshold=80'
```

Backup Server:4.172.1.4:The value of 'allocated pages threshold' has been set to 80%.

例 7 この例では、アロケーション・ページのスレッシュヨルドがデフォルト値の 40 % に設定されます。予約ページのスレッシュヨルドへの影響はありません。

```
sp_dumpoptimize 'allocation_threshold=default'
```

Backup Server:4.172.1.4:The value of 'allocated pages threshold' has been set to 40%.

例 8 予約ページの割合が 60 % 以上のデータベース内のディスクは、このディスク上のアロケーション・ページを読み込むことなくダンプされます。残りのディスクでは、アロケーション・ページが読み込まれます。また、アロケーション・ユニットに 30 % 以上のページが割り付けられている場合は、そのアロケーション・ユニット全体がダンプされます。

```
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%.
```

例 9 この例では、スレッシュホルドの現在の値が表示されます。

```
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%.
```

使用法

- `sp_dumpoptimize` を使用してスレッシュホルドを設定すると、このスレッシュホルドは、データベースが存在する個々のデバイスに対して機能します。
- `sp_dumpoptimize` によって値が設定された場合、これらの値はただちに有効になります。`Backup Server` を再起動する必要はありません。ただし、変更内容は、`Backup Server` が起動している間だけ有効になります。`Backup Server` を再起動すると、デフォルト値が使用されます。
- `sp_dumpoptimize` を複数回発行した場合は、一番最後に指定されたスレッシュホルドが、後で実行されるダンプで使用されます。たとえば、初めに `reserved_threshold` 値を設定して、その後に `archive_space=maximum` を発行すると、その値は `reserved_threshold` に設定した前の値を上書きします。
- `sp_dumpoptimize` 値を変更してから各データベースのダンプを実行することによって、データベースごとにダンプのスレッシュホルドを変えることができます。

- スレッシュホルドの最適値は、データベースによって異なることがあります。したがって、ダンプのパフォーマンスは、I/O 設定とデータベースで使用されるスペースの量によって異なります。**DBA** は、異なる値を使用してダンプを試行してダンプ所要時間が最も短い値を選びます。これによって、データベースの適切な設定が決められます。
- `sp_dumpoptimize` は、ローカル・ダンプにもリモート・ダンプにも使用できます。
- `sp_dumpoptimize` を使用しても、トランザクション・ログのダンプまたはロードのパフォーマンスへの影響はありません。したがって、`dump transaction`、`load database`、または `load transaction` オペレーションの前にこれを発行する必要はありません。
- パラメータを付けずに `sp_dumpoptimize` を発行すると、スレッシュホルドの現在の値がクライアントに表示されます。
- アーカイブ・デバイスのスループットが、すべてのデータベース・ディスクの累積スループット以上である設定では、`archive_space=maximum` を使用すると、ダンプ速度が速くなる場合があります。ただし、アーカイブ・デバイスのスループットが、すべてのデータベース・ディスクの累積スループット未満である設定でこのオプションを使用すると、ダンプ速度が低下するおそれがあります。
- このプロシージャのオプション名と値は、それぞれの名前と値を識別するユニークな部分文字列に省略することができます。たとえば、オプション `archive_space=maximum` をユニークに識別するには、`ar = ma` で十分です。
- オプション文字列の等号 (=) の前後には、ブランクのスペース文字を付けても付けなくてもかまいません。
- オプションの名前と値は、大文字と小文字を区別しません。

スレッシュホルド

スレッシュホルドのデフォルト値を次に示します。

- 予約ページ：85%
- アロケーション・ページ：40%

データベースのデバイス・フラグメントの予約ページの割合によって処理が異なります。

- データベースのデバイス・フラグメントの予約ページの割合が予約スレッシュホールドと等しい場合、または予約スレッシュホールドよりも大きい場合、このデータベースに関連するこのデバイス上のすべてのブロックがダンプされます。
- 予約スレッシュホールドよりも小さい場合、**Backup Server** は、このデバイス上の各アロケーション・ユニットのアロケーション割合のチェックを開始します。累積アロケーションの割合によって処理が異なります。
 - 累積アロケーションの割合がアロケーション・スレッシュホールドよりも小さい場合、データが書き込まれたページだけがダンプされます。
 - 累積アロケーションの割合がアロケーション・スレッシュホールドよりも大きい場合、アロケーション・ユニット全体がダンプされます。

パーミッション

`sp_dumpoptimize` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、ダンプするデータベースに対して <code>dump database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データ型の所有者、 <code>sa_role</code> を持つユーザ、または <code>sso_role</code> を持つユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル アロケーション・ページの詳細については、『システム管理ガイド』を参照してください。

コマンド `dump database`、`dump transaction`、`load database`、`load transaction`

sp_encryption

説明	暗号化の情報をレポートします。
構文	<pre> sp_encryption help helpkey sp_encryption help helpkey [, key_name wildcard] [, all_dbs key_copy display_cols] sp_encryption help 'helpkey', {'master' 'dual master'} [, 'display_keys' 'all_dbs'] sp_encryption 'help'[, 'servicekeyname' [, 'display_objs']] sp_encryption 'helpextpasswd' sp_encryption helpcol [, table_name column_name] sp_encryption helpuser [, user_name wildcard][, key_copy login_passwd_check] sp_encryption 'mkey_startup_file'[, {'new_path' 'default_location' 'null'} [, {sync_with_mem sync_with_qrm}]] sp_encryption 'downgrade_kek_size' [, 'true' 'false'] sp_encryption system_encr_passwd, 'newpasswd' [, 'oldpasswd'] </pre>
パラメータ	<p>helpkey</p> <p>次のような暗号化キー・プロパティをリストします。</p> <ul style="list-style-type: none"> データベースに暗号化キーが格納されているかどうか sso_role を持つユーザ、キー管理者、または DBO が実行する場合：キー名、キー所有者、キー値、キー・アルゴリズム、キー・タイプ、埋め込み、初期化ベクトル、キーの暗号化に使用されるパスワードのタイプ、キー・リカバリが有効になっているかどうか、キー・コピー数。出力は、owner.keyname でソートされます。権限のないユーザが実行すると、このコマンドによって keyname、keyowner、keytype がリストされます。 <p>help</p> <p>下位互換性を提供します。出力は helpkey と同じです。</p> <p>key_name</p> <p>調べるキーの名前です。key_name に定義されたプロパティをリストします。key_name が省略された場合、すべてのキーのプロパティをリストします。</p> <p>wildcard</p> <p>現在のデータベース内でワイルドカード・パターンに一致するキーのプロパティをリストします。ワイルドカードの使用の詳細については、『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。</p>

all_dbs

すべての使用可能なデータベースの暗号化キーの情報をリストします。SSO のみが **all_dbs** を実行できます。

key_copy

現在のデータベース内の指定したキーのすべてのユーザ・コピーをリストします。出力は、**key_owner.key_name** でソートされます。次の情報を示します。

- ベース・キーの所有者
- キー・コピーがリカバリ・キー・コピーであるかどうか
- コピーを所有しているユーザ
- ユーザ暗号化パスワード、ログイン・パスワード、またはログイン関連付け (Login Access によって示される) のシステム暗号化パスワードを使用してコピーが暗号化されるかどうか

login_passwd_check

一致するユーザに割り当てられているキー・コピーがユーザのログイン・パスワードと適切に同期されているかどうかを示します。つまり、キー・コピーの最終更新日がログイン・パスワードの日付より新しいかどうかを示します。キー・コピーは、ユーザのログイン・パスワードまたはログイン関連付けを使用して暗号化されます。

display_keys

system_encr_passwd とともに使用され、システム暗号化パスワードを使って暗号化されているキーとキー・コピーを表示します。**master** または **dual master** とともに使用され、マスタ・キーまたはデュアル・マスタ・キーを使って暗号化されているキーとキー・コピーを表示します。

`sp_encryption helpkey, master | 'dual master', display_keys` を実行して、マスタ・キーまたはデュアル・マスタ・キーで保護された暗号化キーを表示するには、システム・セキュリティ担当者、キー管理者、またはデータベース所有者である必要があります。

display_cols

キー名、現在のデータベース内のすべてのキー (または一致するワイルドカード・キー)、キーが暗号化するカラムを表示します。SSO が **display_cols** を含めた場合は、すべての使用可能なデータベースでキーによって暗号化されたカラムを表示します。**sso_role** のないユーザが **display_cols** を実行すると、現在のデータベース内でキーによって暗号化されたカラムのみが表示されます。データは、**key_name**、**key_owner**、**database_name**、**table_owner**、**table_name**、**column_name** でソートされます。

master

マスタ・キーに関する情報をレポートします。

dual master

デュアル・マスタ・キーに関する情報をレポートします。

servicekeyname

`syb_extpasswdkey` または `syb_syscommkey%` に設定します。
`display_objs` と一緒に使用して、サービス・キーで暗号化されたオブジェクトを表示します。

display_objs

オブジェクト所有者を表示します。

`sp_encryption helpkey, keyname, display_objs` を実行して、`syb_extpasswdkey` サービス・キーまたは `syb_syscommkey` サービス・キーで保護された現在のデータベース内のオブジェクトを表示するには、システム・セキュリティ担当者、キー管理者、またはデータベース所有者である必要があります。

helpextpasswd

ステータス・カラムの外部パスワードの暗号化ステータスを表示します。暗号化ステータスは、次のいずれかです。

- **FIPS 暗号化** — パスワードは、FIPS に準拠した暗号化アルゴリズムを使用した `syb_extpasswdkey` サービス・キーで保護されません。
- **必要とするリセット** — システムによってパスワードが削除されたことを示します。手動でパスワードを再設定する必要があります。
- **従来の暗号化** — パスワードは、Adaptive Server 15.7 より前のバージョンのアルゴリズムで保護されます。

`sp_encryption helpextpasswd` を実行して、外部パスワードのステータスをチェックできるのは、システム・セキュリティ担当者だけです。

helpcol column_name

カラム名と、カラムの暗号化に使用されるキーを表示します。SSO が `helpcol` を含めると、キーが現在のデータベースに存在しない場合でも、キー名が出力されます。SSO 以外のユーザが `helpcol` を含めると、キーが現在のデータベースに存在しない場合、`key_name` は省略され、キーの `keyid` が出力されます。出力には次のものが含まれます。`owner.table.column`、`database.owner.keyname`。情報は、`owner.table.column` でソートされます。

helpuser

現在のデータベース内でユーザによって所有されている、またはユーザに割り当てられているキーを表示します。

mkey_startup_file

マスタ・キーのスタートアップ・ファイル名とパスを表示または設定します。sp_encryption は、マスタ・キーのスタートアップ・ファイルを *new_path* またはデフォルト・ロケーションに設定します。null を指定した場合、またはロケーションがない場合、sp_encryption は、現在のマスタ・キーのスタートアップ・ファイル名とパスを表示します。

sync_with_mem

(Cluster Edition のみ) サーバ・メモリにあるマスタ・キーの暗号化キーをマスタ・キーのスタートアップ・ファイルに書き込みます。現在のマスタ・キーの暗号化キーが存在する場合は、そのキーを置き換えます。自動マスタ・キー・アクセスがオフに設定されると、sync_with_mem も無効になります。

マスタ・キーのスタートアップ・ファイルを表示、設定、または同期できるのは、システム・セキュリティ担当者だけです。

sync_with_qrm

ローカル・マスタ・キーのスタートアップ・ファイルをクォーラム・デバイス内のバージョンで更新します。

マスタ・キーのスタートアップ・ファイルを表示、設定、または同期できるのは、システム・セキュリティ担当者だけです。

downgrade_kek_size

downgrade kek size 設定を表示または設定します。true は Adaptive Server が downgrade kek size モードであることを示し、false はこのモードを無効にします。

引数を指定しない場合、sp_encryption は、downgrade_kek_size の現在の値を表示します。

このコマンドを実行するには、システム・セキュリティ担当者またはキー管理者である必要があります。

system_encr_passwd

現在のデータベースでシステム暗号化パスワードを使用して暗号化された暗号化キーおよびキー・コピーを表示します。

system_encr_passwd, all_dbs

各データベースで設定されているシステム暗号化パスワードのプロパティを表示します。出力は、データベース名でソートされます。このコマンドを実行できるのは、システム・セキュリティ担当者のみです。すべてのデータベースにシステム暗号化パスワードが設定されていない場合、Adaptive Server は、メッセージ 19782 を生成します。

The system encryption password has not been set for all available databases

例 **例 1** helpkey パラメータを使用して、現在のデータベースのキー情報を表示します。すべてのキーまたは特定のキーの情報を取得できます。sp_encryption の 2 番目のパラメータは、キー名を表示します。このパラメータには、SQL パターン一致文字を含めることができます。データベース所有者ではなく、sso_role および keycustodian_role がない場合、sp_encryption で表示されるカラムは少なくなります。

このコマンドを SSO、キー管理者、または DBO が実行すると、現在のデータベース内のすべてのベース暗号化キーのプロパティが表示されます。

```
sp_encryption helpkey
```

Key Name	Key Owner	Key Length	Key Algorithm	Key Type	Pad
Init Vector	Protected By	Key Recovery	# of Key Copies		
tinnap_key	tinnap	128	AES	symmetric key	0
1	system encryption password			0	0
tinnap_key1	tinnap	128	AES	symmetric default key	0
1	user Passwd			1	3
sample_key1	dbo	192	AES	symmetric key	1
1	login Passwd			1	2

ユーザ “tinnap” が実行すると、次に示す、現在のデータベース内のすべてのベース暗号化キーのプロパティが表示されます。

```
sp_encryption helpkey
```

Key Name	Key Owner	Key Type
tinnap_key	tinnap	symmetric key
tinnap_key1	tinnap	symmetric default key
sample_key1	dbo	symmetric key

システム・セキュリティ担当者でない場合、または `keycustodian_role` がない場合、クエリは、ユーザが現在のデータベースで所有するすべてのベース・キーを表示します。2番目のパラメータとして `user_name` を指定しない場合、クエリは、ユーザが所有するベース・キーを表示します。

例 2 現在のデータベース内の SSO、キー管理者、または DBO が実行すると、ベース暗号化キー `sample_key1` のプロパティが表示されます。

```
sp_encryption helpkey, sample_key1
```

Key Name	Key Owner	Key Length	Key Algorithm	Key Type
Pad	Init Vector	Protected By	Key Recovery	# of Key Copies
sample_key1	dbo	192	AES	symmetric Key
1	1	Login	1	2

権限のないユーザ “tinnap” がこのコマンドを実行すると、現在のデータベース内のベース暗号化キー `sample_key1` の次のプロパティが表示されます。

```
sp_encryption helpkey, sample_key1
```

Key Name	Key Owner	Key Type
sample_key1	dbo	ymmetric key

例 3 すべての使用可能なデータベース内のすべてのベース暗号化キーのプロパティを表示します (SSO だけがこのコマンドを実行できます)。

```
sp_encryption helpkey, NULL, all_dbs
```

Db.Owner.Keyname	Key Length	Key Algorithm	Key Type
Pad	Init Vector	Protected By	#of Key Copies
keydb.dbo.cc_key	256	AES	symmetric default key
1	1	system encr passwd	0
0	0	system encr passwd	4
keydb1.tinnap.tinnap_key	128	AES	symmetric key
0	1	system encr passwd	0
keydb1.tinnap.tinnap_key1	128	AES	symmetric default key
0	1	user password	3
keydb1.dbo.sample_key1	192	AES	symmetric key
1	1	login passwd	2

例 4 `all_dbs` は、すべてのデータベースにわたるキーの情報が必要であることを示します。`all_dbs` パラメータを使用するには、`sso_role` が必要です。

すべての使用可能なデータベース内で `%key` に類似した名前を持つすべてのベース暗号化キーのプロパティが表示されます。

```
sp_encryption helpkey, '%key', all_dbs
```

Db.Owner.Keyname	Key Length	Key Algorithm	Key Type
Pad Init Vector	Protected By	Key Recovery	#of Key Copies
keydb.dbo.cc_key	256	AES	symmetric default key
1 1	system encr passwd	0	0
keydb1.tinnap.tinnap_key	128	AES	symmetric key
0 1	system encr passwd	0	0

例 5 `SSO`、キー管理者、または `DBO` が実行すると、現在のデータベース内で “`tinnap%`” と類似する名前を持つすべてのベース暗号化キーのプロパティが表示されます。

```
sp_encryption helpkey, "tinnap%"
```

Key Name	Key Owner	Key Length	Key Algorithm	Key Type
Pad Init Vector	Protected By	Key Recovery	# of Key Copies	
tinnap_key	tinnap	128	AES	symmetric key
0 1	system encr passwd	0	0	
tinnap_key1	tinnap	128	AES	symmetric default key
0 1	user passwd	1	3	

“`tinnap`” が実行した場合、“`tinnap%`” と類似する名前を持つ、現在のデータベース内のベース暗号化キーの、次のプロパティが表示されます。

```
sp_encryption helpkey, "tinnap%"
```

Key Name	Key Owner	Key Type
tinnap_key	tinnap	symmetric key
tinnap_key1	tinnap	symmetric default key

例 6 `key_copy` を 3 番目のパラメータとして使用して、キー・コピーの情報を表示します。すべてのキー・コピーの情報を表示するには、2 番目のパラメータに `keyname` の値ではなく `null` を入力します。

`keyname` パラメータでパターン一致文字を使用できます (前述の例を参照)。

```
sp_encryption helpkey, tinnap_key1, key_copy
```

Owner.Keyname	Assignee	Protected by	Key Recovery
tinnap.tinnap_key1	joesmp	user passwd	0
tinnap.tinnap_key1	samcool	user passwd	1
tinnap.tinnap_key1	billyg	user passwd	0

ユーザ “joesmp” が実行すると、現在のデータベースで、ユーザ “joesmp” に割り当てられているすべての暗号化キー・コピーと、このユーザがキー所有者であるキーのすべてのキー・コピーが表示されます。

```
sp_encryption helpkey, tinnap_key1, key_copy
```

Owner.Keyname	Assignee	Protected by	Key Recovery
tinnap.tinnap_key1	joesmp	user passwd	0

例 7 `display_cols` パラメータを使用すると、現在のデータベースのキーによって暗号化された、すべての使用可能なデータベース内のすべての暗号化カラムが表示されます。`sso_role` がない場合、クエリは、現在のデータベースのキーによって暗号化された、現在のデータベース内の暗号化カラムだけを表示します。2 番目のパラメータには、パターン一致文字または `key_name` を使用できます。`sso_role` として `key_name` にパターン一致文字を使用すると、クエリは、パターン一致 `key_name` によって暗号化された、すべての使用可能なデータベース内のすべての暗号化カラムを表示します。2 番目のパラメータに `key_name` を使用し、`sso_role` がある場合、指定した `key_name` によって暗号化された、すべての使用可能なデータベース内のすべての暗号化カラムが表示されます。

```
sp_encryption helpkey, null, display_cols
```

Key Name	Key Owner	Database Name	Table Owner	Table Name	Column Name
tinnap_key	tinnap	testdb1	tinnap	t3	c3
tinnap_key1	tinnap	testdb1	tinnap	t4	c4
sample_key1	dbo	coldb	dbo	t1	c1
sample_key1	dbo	coldb	billyg	t2	c2

例 8 現在のデータベースのシステム暗号化パスワードを使用して暗号化された暗号化キーおよびキー・コピーを表示します。これらの権限がない場合、クエリは、システム暗号化パスワードで暗号化されたユーザが所有するキー、またはそのようなユーザに割り当てられているキーを表示します。

```
sp_encryption helpkey, system_encr_passwd, display_keys

Owner.Keyname                                Assignee
-----
dbo.cc_key                                    NULL
dbo.sample_key1                              NULL
dbo.sample_key1                              tinnap
```

例 9 データベース所有者、あるいは `keycustodian_role` または `sso_role` を持つユーザが実行した場合、`helpuser` パラメータは、現在のデータベース内のユーザによって所有されているすべてのベース・キーを表示します。

```
sp_encryption helpuser

Owner.Keyname                                Protected by
-----
tinnap.tinnap_key                            system encr passwd
tinnap.tinnap_key1                          user passwd
dbo.sample_key1                              login passwd
```

ユーザ “tinnap” がこのコマンドを実行すると、現在のデータベース内でこのユーザが所有するすべてのベース・キーがリストされます。

```
sp_encryption helpuser

Owner.Keyname                                Protected by
-----
tinnap.tinnap_key                            system encr passwd
tinnap.tinnap_key1                          user passwd
```

例 10 データベース所有者、あるいは `keycustodian_role` または `sso_role` のあるユーザは、`key_copy` パラメータと `helpuser` パラメータを使用して、現在のデータベース内の 1 人または複数のユーザに割り当てられているキー・コピーを表示できます。`user` パラメータには、パターン一致文字を使用できます。現在のデータベース内のすべてのユーザのキー・コピーが表示されます。

```
sp_encryption helpuser, NULL, key_copy
Owner.Keyname      Assignee      Protected by      Key Recovery
-----
```

dbo.sample_key1	tinnap	login passwd	0
tinnap.tinnap_key1	joesmp	user passwd	0
dbo.sample_key1	joesmp	login passwd	1
tinnap.tinnap_key1	samcool	user passwd	1
tinnap.tinnap_key1	billyg	user passwd	0

データベース所有者ではなく、`keycustodian_role` および `sso_role` がない場合、このクエリは、ユーザが所有するキーおよびその他のキー所有者がユーザに割り当てたキー・コピーを表示します。たとえば、“tinnap” がこのクエリを実行すると、次の結果が表示されます。

```
sp_encryption helpuser, NULL, "key_copy"
```

Owner.Keyname	Assignee	Protected by	Key Recovery
-----	-----	-----	-----
dbo.sample_key1	tinnap	login passwd	0
tinnap.tinnap_key1	joesmp	user passwd	0
tinnap.tinnap_key1	samcool	user passwd	1
tinnap.tinnap_key1	billyg	user passwd	0

例 11 データベース所有者、あるいは `keycustodian_role` または `sso_role` を持つユーザである場合、`helpcol` は、現在のデータベース内のすべての暗号化カラムおよびカラムの暗号化に使用されたキーを表示します。これらの権限がない場合、暗号化キーが別のデータにあると、`helpcol` は、`key_name` ではなく `keyid` を表示します。

```
sp_encryption helpcol
```

Owner.Table.Column	Db.Owner.Keyname
-----	-----
dbo.t1.c1	keydb1.dbo.sample_key1
billyg.t2.c2	keydb.dbo.sample_key1
tinnap.t3.c3	coldb.dbo.sample_key2

例 12 `table_name` および `column_name` パラメータとともに `helpcol` パラメータを含めて、テーブル内のすべての暗号化カラム、または特定の暗号化カラムを表示します。`sso_role` を持つユーザが実行すると、次のクエリは、現在のデータベース内のテーブル `t3` のすべての暗号化カラムと、すべての使用可能なデータベース内でカラムの暗号化に使用されたキーを表示します。`sso_role` を持たないユーザが実行すると、キーが現在のデータベースにない場合、このクエリは、キーの名前ではなく `ID` を表示します。2 番目のパラメータには、`[database_name.][table_name.][column_name]` の組み合わせを設定できます。

```
sp_encryption helpcol, t3
```

```

Owner.Table.Column          Db.Owner.Keyname
-----
tinnap.t3.c3                coldb.dbo.sample_key2
    
```

例 13 各データベースのシステム暗号化パスワード・プロパティを表示します (このクエリを実行するには、sso_role が必要です)。

```

sp_encryption helpkey, system_encr_passwd, all_dbs
Database Type of system_encr_passwd Last modified by      Date
-----
master          persistent                sa Aug 26 2008 10:05AM
    
```

例 14 現在のデータベースのマスタ・キーで暗号化されたすべての暗号化キーを表示します (このクエリを実行するには、sso_role, keycustodian_role が必要か、データベース所有者である必要があります)。

```

sp_encryption helpkey, 'master', display_keys
Owner.Keyname      Assignee
-----
user1.key_dual     NULL
user1.key_mst      NULL
user4.key_dC_pwd   NULL
user4.key_dC_pwd   user5
user4.key_dC_pwd   user6
user4.key_dC_pwd   KC_tdb1
    
```

例 15 現在のサーバ用に設定された現在のマスタ・キーのスタートアップ・ファイルの名前とロケーションを表示します。

```

sp_encryption mkey_startup_file
Msg 19956, Level 16, State 1: Procedure 'sp_encryption',
Line 298: The current master key startup file
is: '/sybase/release/ASE-
150/init/ase_encrcols_mk_l157.dat'.
    
```

例 16 syb_syscommkey_123456 キーで暗号化された、ユーザ 1 とユーザ 2 が所有している 3 つのストアード・プロシージャを表示します。

```

sp_encryption helpkey, "syb_syscommkey%", display_objjs
Key Name          Key Owner      Database Name
-----
Object owner      Object Name
-----
syb_syscommkey_1234567890ab      dbo            testdb
                                user1         sp_mysprocl
syb_syscommkey_abcdefg hijkl123456      dbo            testdb
    
```

user1	sp_mysproc2_	
syb_syscommkey_ABCDEF123456	dbo	testdb
user2	sp_mysproc3	

使用法

- `sp_encryption` を実行するユーザに付与された権限によって出力が決定されます。詳細については、「[使用制限](#)」(358 ページ)を参照してください。
- `sp_encryption helpkey` を実行し、キーがデータベースに存在しない場合は、情報メッセージが返されます。
- キー・コピーに関する情報を取得するには、`key_copy` パラメータを指定する必要があります。`key_copy` パラメータを指定しない場合、`sp_encryption` によってベース・キーの情報のみが返されます。
- `sp_encryption helpkey, keyname, key_copy` の `keyname` が NULL である場合、SSO、キー管理者、または DBO に対しては現在のデータベースのすべてのキー・コピーがリストされます。権限のないユーザが実行すると、現在のデータベースでこのユーザに割り当てられているすべてのキー・コピーと、現在のデータベースでこのユーザが所有するキーのすべてのキー・コピーがリストされます。
- `sp_encryption helpcol` の場合、`column_name` では、`name.name.name` の形式が使用されます。パラメータの意味は次のとおりです。
 - `name` — `sp_encryption` は、この名前のテーブルを検出しなかった場合、この名前のすべてのカラムを検索します。
 - `name.name` — `owner.table` です。`sp_encryption` は、この名前のテーブルを検出しなかった場合、`table.column` という名前の 1 つのカラムを検索します。
 - `name.name.name` — `owner.table.name` です。

現在のデータベースでこれらの規則によって識別されるすべてのカラムに対して、`sp_encryption` は、カラムの暗号化に使用されたキーとともにカラム名を表示します。

`sp_encryption helpcol, column_name` の出力は、`owner.table.column` および `db.owner.keyname` です。SSO 以外のユーザが実行し、暗号化カラムとは異なるデータベースにキーが存在する場合、`keyname` は `database.keyid` として表されます。結果セットは、`owner.table.column` でソートされます。

使用制限

- SSO のみが `sp_encryption helpkey` [, *keyname* | wildcard], `all_dbs` を実行して、すべてのデータベース内のキーのプロパティを取得できます。 `sso_role` を持たないユーザがこのコマンドを実行すると、“権限のないユーザ” エラー・メッセージが返されます。 キー名またはワイルドカードの条件を満たすキーが存在しない場合、Adaptive Server は、「There are no encryption keys (key copies) like *keyname* in all databases」というメッセージを返します。
- SSO が `sp_encryption helpkey`, *keyname*, `display_cols` を実行すると、*keyname* によって暗号化された、すべての使用可能なデータベースのすべてのカラムがリストされます。 権限のないユーザが実行すると、*keyname* によって暗号化された、現在のデータベース内のカラムがリストされます。

SSO が `sp_encryption helpkey`, *keyname* を実行し、`display_cols` および *keyname* の値が NULL である場合は、すべての使用可能なデータベースのすべての暗号化カラムが表示されます。 権限のないユーザが実行すると、現在のデータベース内のすべての暗号化カラムが表示されます。

- SSO、キー管理者、または DBO が、`helpuser` パラメータの *user_name* と *key_copy* を指定せずに `sp_encryption helpuser`, *user_name*, *key_copy* を実行すると、現在のデータベースのすべてのユーザが所有するすべてのベース・キーがリストされます。 権限のないユーザが、*user_name* または *key_copy* を指定せずに `sp_encryption` を実行すると、現在のユーザが所有するベース・キーが表示されます。

いずれかのユーザが `sp_encryption helpuser`, *user_name* を実行すると、`owner.keyname` が所有するすべてのベース・キーがリストされます。 権限のないユーザがこのコマンドを実行し、ベース・キーを所有しない場合、Adaptive Server は、そのことを示す情報メッセージを表示します。

SSO、キー管理者、または DBO が `sp_encryption helpuser`, *user_name*, *key_copy* を実行すると、*user_name* に割り当てられているキー・コピーがリストされます。 権限のないユーザがこのコマンドを実行すると、このユーザに割り当てられているキー・コピーと、現在のデータベースでこのユーザが所有するキーのすべてのキー・コピーが、結果セットのカラム `Owner.Keyname`, `Assignee`, `Type of Password`, `Key Recovery` とともにリストされます。 出力は、`Assignee` でソートされます。

`sp_encryption helpuser user_name, key_copy` の `user_name` が NULL である場合、SSO、キー管理者、または DBO に対しては現在のデータベースのすべてのキー・コピーがリストされます。権限のないユーザが実行すると、現在のデータベースでこのユーザに割り当てられているすべてのキー・コピーと、このユーザが所有するキーのキー・コピーがリストされます。

- SSO、キー管理者、または DBO が `sp_encryption helpkey, keyname, key_copy` を実行すると、`keyname` の現在のデータベース内のキー・コピーがリストされます。権限のないユーザが実行すると、その `keyname` のユーザに割り当てられているキー・コピーがリストされ、このユーザがキー所有者である場合は、その `keyname` のキー・コピーもリストされます。
- SSO、キー管理者、または DBO は、`sp_encryption helpkey, system_encr_passwd, display_keys` を実行して、システム暗号化パスワードによって暗号化された、現在のデータベースのすべてのキーとキー・コピーに関する情報を取得できます。権限のないユーザは、現在のデータベースでユーザが所有する、またはユーザに割り当てられているベース暗号化キーまたはキー・コピーに関する情報を取得できません。ログイン関連付けのためにキーが作成された場合にのみ、キー・コピーはシステム暗号化パスワードによって暗号化されます。出力は、`owner.keyname` でソートされます。

パーミッション

`sp_encryption` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効

細密なパーミッションが有効化されている場合は、次のようになる。

`downgrade_kek_size` — manage security configuration 権限を持つユーザであることが必要。

`help/help_key system_encr_passwd, display_keys` — manage column encryption key 権限を持つユーザであることが必要。自分のキーは、すべてのユーザが表示できる。

`help/help_key system_encr_passwd` — manage column encryption key 権限を持つユーザであることが必要。

`help/help_key master key/ dual master key, display_keys` — manage master key 権限を持つユーザであることが必要。

`help/help_key keyname/wild card, display_cols` — データベース間チェックの場合は `use any database` 権限を持つユーザであることが必要。現在のデータベースの場合はすべてのユーザが表示できる。

`help/help_key service keyname, display_objs` — manage service key 権限を持つユーザであることが必要。

`help/help_key keyname/wild card, all_dbs` — キーの種類に応じた次の権限を持つユーザであることが必要。

- カラム暗号化キー — manage column encryption key
- マスタ・キー — manage master key
- サービス・キー — manage service key

データベース間チェックの場合、上記の3つのいずれかに加え、`use any database` パーミッションが必要。

`help/help_key keyname wildcard` — キーの種類に応じた次の権限を持つユーザであることが必要。

- カラム暗号化キー — manage column encryption key
- マスタ・キー — manage master key
- サービス・キー — manage service key

権限のないユーザの場合、限定された暗号化キー情報が表示されます。

`help/help_key keyname wildcard, key_copy` — キーの種類に応じた次の権限を持つユーザであることが必要。

- カラム暗号化キー — manage column encryption key
- マスタ・キー — manage master key

`helpcol` — データベース間チェックの場合は `use any database` 権限を持つユーザであることが必要。

`helpextpassword` — manage service key 権限を持つユーザであることが必要。

`helpuser username/wildcard, [key_copy/login_passwd_check]` — manage any encryption key 権限を持つユーザであることが必要。権限のないユーザは、自分のキーを表示できる。

`mkey_startup_file` — manage security configuration 権限を持つユーザであることが必要。

`system_encr_passwd` — manage column encryption key 権限を持つユーザであることが必要。

細密なパーミッションが無効	<p>細密なパーミッションが無効化されている場合は、次のようになる。</p> <p><code>downgrade_kek_size</code> - <code>sso_role</code> または <code>keycustdian_role</code> を持つユーザであることが必要。</p>
	<p><code>help/help_key system_encr_passwd, display_keys</code> - データベース所有者、<code>sso_role</code> を持つユーザ、または <code>keycustdian_role</code> を持つユーザであることが必要。自分のキーは、すべてのユーザが表示できる。</p>
	<p><code>help/help_key system_encr_passwd</code> - データベース所有者、<code>sso_role</code> を持つユーザ、または <code>keycustdian_role</code> を持つユーザであることが必要。</p>
	<p><code>help/help_key master key/ dual master key, display_keys</code> - データベース所有者、<code>sso_role</code> を持つユーザ、または <code>keycustdian_role</code> を持つユーザであることが必要。</p>
	<p><code>help/help_key keyname/wild card, display_cols</code> - データベース間チェックの場合は <code>sso_role</code> を持つユーザであることが必要。現在のデータベースの場合はすべてのユーザが表示できる。</p>
	<p><code>help/help_key service keyname, display_objs</code> - <code>sso_role</code> を持つユーザか、<code>keycustdian_role</code> を持つユーザであることが必要。</p>
	<p><code>help/help_key keyname/wild card, all_dbs</code> - <code>sso_role</code> を持つユーザであることが必要。</p>
	<p><code>help/help_key keyname wildcard</code> - データベース所有者、<code>sso_role</code> を持つユーザ、または <code>keycustdian_role</code> を持つユーザであることが必要。</p>
	<p><code>help/help_key keyname wildcard, key_copy</code> - データベース所有者、<code>sso_role</code> を持つユーザ、または <code>keycustdian_role</code> を持つユーザであることが必要。</p> <p>権限の有無により、次のように動作が異なる。</p> <ul style="list-style-type: none"> • 権限のないユーザの場合 - <code>key_copy</code> の情報のみを表示する。 • 権限のあるユーザの場合 - データベース内のすべてのユーザについて、暗号化キーおよび <code>key_copy</code> の情報を表示する。
	<p><code>helpcol</code> - <code>sso_role</code> を持つユーザであることが必要。</p>
	<p><code>helpxtpassword</code> - <code>sso_role</code> を持つユーザであることが必要。</p>
	<p><code>helpuser username/wildcard, [key_copy/login_passwd_check]</code> - データベース所有者、<code>sso_role</code> を持つユーザ、または <code>keycustdian_role</code> を持つユーザであることが必要。権限のないユーザは、自分のキーを表示できる。</p>
	<p><code>mkey_startup_file</code> - <code>sso_role</code> を持つユーザであることが必要。</p>
	<p><code>system_encr_passwd</code> - <code>sso_role</code> または <code>keycustdian_role</code> を持つユーザであることが必要。</p>
	<p><code>verify_downgrade</code> - <code>sso_role</code> または <code>keycustdian_role</code> を持つユーザであることが必要。</p>

sp_engine

説明	エンジンをオンラインまたはオフラインにできるようにします。スレッド・モードでエンジンをオンラインまたはオフラインにするには、 <code>alter thread pool</code> を使用します。
構文	<pre>sp_engine {"online" [offline can_offline] [, engine_id] ["shutdown", engine_id]}</pre>
パラメータ	<p>“online”</p> <p>エンジンをオンラインにします。sp_configure “max online engines” の値を、現在オンラインであるエンジンの数より大きくしてください。“online” は予約キーワードなので、引用符を使用してください。</p> <p>スレッド・モードの online では、syb_default_pool のスレッド・カウントが 1 ずつ増加します。</p> <p>offline</p> <p>エンジンをオフラインにします。engine_id パラメータを使用して、特定のエンジンをオフラインにするよう指定することもできます。</p> <p>スレッド・モードの offline では、syb_default_pool のスレッド・カウントが 1 ずつ減少します。</p> <p>can_offline</p> <p>エンジンをオフラインにできるかどうかの情報を返します。can_offline は、このエンジンの状態が online の場合に、このエンジンとの結びつきがある Adaptive Server タスクを (たとえば、Omni タスク中や java.net タスク中に) 返します。engine_id を指定しない場合、このコマンドは、engine_id が最も高い sysengines のエンジンのステータスを表示します。</p> <p>スレッド・モードでの can_offline は、エンジンの総数が syb_default_pool のスレッドの総数より少ない場合、および syb_default_pool のスレッドの総数が 2 より大きいまたは 2 の場合にのみ成功します。</p> <p>engine_id</p> <p>エンジンの ID です。engine_id パラメータは省略可能です。engine_id を指定しない場合、sp_engine は、sysengines 内で検出されたエンジンの値に対して、増分または減分した engine_id の値を使用します。たとえば、システムでエンジン 0、1、2、3 を使用していて、エンジンの ID が指定されない場合、sp_engine は、エンジン ID 3、次にエンジン ID 2 と順にオフラインにしていきます。</p> <p>このパラメータはスレッド・モードでは無視されます。</p>

“shutdown”

エンジンを強制的にオフラインにします。このエンジンと結び付きがあるタスクがある場合、それらは5分間の待機の後、強制終了されます。shutdownは予約キーワードなので、引用符を使用してください。

例

例1 エンジン1をオンラインにします。メッセージはプラットフォーム固有です(この例ではSun Solarisが使用されています)。

```
sp_engine "online", 1
02:00000:00000:2001/10/26 08:53:40.61 kernel Network and device connection
limit is 3042.
02:00000:00000:2001/10/26 08:53:40.61 kernel SSL Plus security modules
loaded successfully.
02:00000:00000:2001/10/26 08:53:40.67 kernel engine 2, os pid 8624 online
02:00000:00000:2001/10/26 08:53:40.67 kernel Enabling Sun Kernel
asynchronous disk I/O strategy
00:00000:00000:2001/10/26 08:53:40.70 kernel ncheck:Network fc0330c8
online
```

例2 このエンジンと結び付きがあるタスクを現在実行しているエンジンを、オフラインにする手順を説明しています。

```
select engine, status from sysengines

engine      status
-----
0           online
1           online
2           online
3           online
```

エンジン1をオフラインにする場合

```
sp_engine offline, 1
```

```
The following task(s) will affect the offline process:
spid:19 has outstanding ct-lib connections.
```

次に、前と同じクエリを実行します。エンジン1がオフライン状態になっていることが示されます。

```
select engine, status from sysengines

engine      status
-----
0           online
1           in offline
2           online
3           online
```

エンジン 1 と結び付きがあるタスクが終了するとすぐに、Adaptive Server は次のようなメッセージをエラー・ログに発行します。

```
02:00000:00000:2001/10/26 09:02:09.05 kernel engine 1, os pid
8623 offline
```

例 3 エンジン 1 をオフラインにできるかどうかを確認します。

```
sp_engine can_offline, 1
```

例 4 エンジン 1 をオフラインにします。

```
sp_engine offline, 1
```

最後に、Adaptive Server は次のようなメッセージを返します。

```
01:00000:00000:2001/11/09 16:11:11.85 kernel Engine 1 waiting for
affinitated process(es) before going offline
01:00000:00000:2001/11/09 16:11:11.85 kernel Process 917518 is preventing
engine 1 going offline
00:00000:00000:2001/11/09 16:16:01.90 kernel engine 1, os pid
21127 offline
```

例 5 エンジン 1 を停止します。

```
sp_engine shutdown, 1
```

使用法

- `sp_engine` はプロセス・モードでのみ動作するので、スレッド・モードで `sp_engine` を実行すると、Adaptive Server ではエラー・メッセージが発行されます。スレッド・モードでは `alter thread pool` を使用します。
- エンジン 0 をオフラインにしたり、停止にしたりすることはできません。
- エンジンのステータスと現在オンラインのエンジンを、次のクエリを使って判断できます。

```
select engine, status from sysengines
where status = "online"
```

- `online` と `shutdown` は、キーワードなので、引用符で囲んでください。
- `max online engines` の値が、その時点の `online` ステータスのエンジン数より大きく、CPU が十分に追加エンジンをサポートできる場合にだけ、エンジンをオンラインにできます。
- `sp_engine` は、オープン・トランザクションが存在しない場合に連鎖トランザクション・モードを使用するセッションで実行できます。
- `engine offline` は、そのエンジンと結び付きがあるサーバ・プロセスがある場合、失敗するか、すぐには有効にならない可能性があります。

sp_engine “offline” と sp_engine “shutdown” の使用

sp_engine “offline” を使用しても、すぐにはエンジンがオフラインにならず、エンジン・テーブルで休止状態になっているように見えます。これは、プロセスが他のエンジンにマイグレートできないエンジンに付加されたことが原因です。この場合、エンジンは新しい作業を受け付けず、最小限の CPU サイクルを消費します。engine offline の完了を妨げているプロセスが、終了するかまたはマイグレート可能になると、エンジンは休止状態から完全にオフラインとなり、エンジン・テーブルから削除されます。

sp_engine "shutdown" は、さらに積極的な意味を持つ offline コマンドです。sp_engine "shutdown" は、エンジンをオフラインにすることを妨げているプロセスを積極的に停止して、強制的にエンジンを停止します。

ただし、sp_engine "shutdown" を Client Library™ 接続または Java 接続があるエンジンに対して使用すると、次のエラー・メッセージが返されます。

```
Engine has outstanding ct-lib/java connections and
cannot be offlined.
```

この状態が生じた場合は、接続が解放されて、エンジンが停止できるようになるまで、コマンドを数分ごとに繰り返し実行します。

パーミッション

sp_engine のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage server 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

sp_errorlog

説明	エラー・ログのパスを動的に変更します。
構文	<pre>sp_errorlog "change log", "new_path" [{"jslog true" "jslog false"}] sp_errorlog "help", "change log"</pre>
パラメータ	<p><i>new_path</i> エラーログの新しいパス。<i>new_path</i> の最大長は 255 文字です。</p> <p><i>jslog true</i> デフォルトのオプション。Job Scheduler が実行されている場合、<i>change log</i> は Job Scheduler Agent ログを、新しい Adaptive Server エラー・ログが置かれるディレクトリに移動しようとします。どちらのログもエラー・メッセージがあれば、エラー・メッセージを示します。</p> <p><i>jslog false</i> Job Scheduler Agent ログの位置を変更しません。</p>
例	<p>例 1 Job Scheduler ログの位置を変更しないで、Adaptive Server エラー・ログを新しい位置に変更するには、次のコマンドを使用します。</p> <pre>sp_errorlog 'change log', '\$SYBASE/\$SYBASE_ASE/install/new.log', 'jslog false'</pre> <p>Adaptive Server エラー・ログの位置は <i>\$SYBASE/\$SYBASE_ASE/install/new.log</i> に変更されます。ただし、Job Scheduler Agent ログの位置は変わりません。</p> <p>例 2 次の例では、エラー・ログの位置を <i>\$SYBASE/\$SYBASE_ASE/install/new.log</i> に変更します。</p> <p>Job Scheduler Agent が実行されている場合は、エージェント・ログの位置も <i>\$SYBASE/\$SYBASE_ASE/install/new.log</i> に変更されます。</p> <p>Job Scheduler Agent が実行されていない場合、Adaptive Server はエージェント・ログの位置を変更しません。エージェント・ログの位置は変わらないことを示すメッセージが表示されます。</p> <pre>sp_errorlog 'change log', '\$SYBASE/\$SYBASE_ASE/install/new.log', 'jslog true'</pre> <p>例 3 次の例では、Adaptive Server エラー・ログの位置を <i>\$SYBASE/\$SYBASE_ASE/install/new.log</i> に変更します。</p> <p>Job Scheduler Agent が実行されている場合は、エージェント・ログも <i>\$SYBASE/\$SYBASE_ASE/install/new.log</i> に変更されます。</p>

Job Scheduler Agent が実行されていない場合、Adaptive Server は Job Scheduler Agent ログのパスを変更しません。エージェント・ログの位置は変わらないことを示すメッセージが表示されます。

```
sp_errorlog 'change log',
'$SYBASE/$SYBASE_ASE/install/new.log'
```

使用法

`sp_errorlog` は、新しい位置への切り替えが成功すると 0 を返します。エラーが発生した場合はゼロ以外の値を返します。

注意 サーバの再起動時にエラー・ログの新しい位置を取得するには、`runserver` ファイルで `-e` 引数を更新します。

パーミッション

`sp_errorlog` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`errorlog` 監査オプションを使用すると、エラー・ログ管理の変更を監査できます。

`errorlog` 監査オプションが有効になっている場合、エラー・ログが変更されると、`event=127` の監査レコードが生成されます。`sybsecurity` のイベントを確認するには、次の文を発行します。

```
select * from sybsecurity..sysaudits01 where event=127
```

`errorlog` 監査オプションを有効にするには、次のように入力します。

```
sp_audit 'errorlog', 'all', 'all', 'on'
```

参照

`runserver` ファイルについては、『設定ガイド』の「第 8 章 エラー・メッセージのロギングとイベントのロギング」を参照してください。

sp_estspace

説明	テーブルとそのインデックスに必要な領域と、インデックスの作成に必要な時間を見積もります。
構文	<code>sp_estspace table_name, no_of_rows, fill_factor, cols_to_max, textbin_len, iosec, page_size</code>
パラメータ	<p><i>table_name</i> テーブルの名前を指定します。これはすでに現在のデータベース内に存在している必要があります。</p> <p><i>no_of_rows</i> テーブルに含まれるローの数の概算です。</p> <p><i>fill_factor</i> インデックスのフィルファクタです。デフォルトは <code>null</code> で、この値が選ばれると、Adaptive Server が所有するデフォルトのフィルファクタが使用されます。</p> <p><i>cols_to_max</i> 平均長ではなく最大長で計算したい可変長カラムをカンマで区切って指定します。デフォルトでは、可変長カラムの平均長が使用されます。</p> <p><i>textbin_len</i> ロー 1 つあたりの、すべての <code>text</code> および <code>image</code> カラムの長さです。デフォルト値は 0 です。テーブルが <code>text</code> または <code>image</code> データを保管する場合のみ、値を指定する必要があります。<code>text</code> および <code>image</code> カラムは、他のデータとは別のデータ・ページに保管されます。実際のテーブル・ローには、<code>text</code> または <code>image</code> 値へのポインタが保管されます。<code>sp_estspace</code> は、ローの <code>text</code> または <code>image</code> ページのサイズについての情報を別の行で提供します。</p> <p><i>iosec</i> このマシンの 1 秒あたりのディスク I/O 数です。デフォルトは毎秒 30 I/O です。</p>

pagesize

特定のテーブルを、指定されたページ・サイズのサーバにマイグレートする場合、そのテーブル(とそのインデックスのすべて)に必要な空き領域を予測することができます。ページ・サイズ(2048、4096、8192、16384、または 2K、4K、8K、16K)を指定することも、NULL を指定して現在のページ・サイズを使用することもできます。単位に“K”を使用しない場合、*pagesize* のデフォルト単位はバイトです。ページ割り付けではさまざまなオブジェクトに同じサイズのページを割り付けるため、*page_size* 値はすべてのページ・タイプ(index、data、text など)に適用されます。

例

例 1 titles テーブルとそのインデックスに必要な領域と、インデックスの作成に必要な時間を計算します。ローの数は 10,000、フィルファクタは 50% であり、2 つの可変長カラムがカラムの最大サイズを使用して計算されます。ディスク I/O 速度は毎秒 25 I/O です。

```
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

例 2 au_pix テーブル内の既存の image データの平均長を使用して、1,000 のローが割り当てられているテーブルのサイズを計算します。また、このサイズを定数として提供することもできます。

```
declare @i int
select @i = avg(datalength(pic)) from au_pix
exec sp_estspace au_pix, 1000, null, null, @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

例 3 他のすべての値にデフォルトを使用して、50,000 のローが割り当てられている **titles** テーブルのサイズを計算します。

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

例 4 blurbs テーブルにクラスタード・インデックスを追加した後に実行します。

```
declare @i int
select @i = avg(datalength(copy)) from blurbs
exec sp_estspace blurbs, 6, null, null, @i
```

name	type	idx_level	Pages	Kbytes
blurbs	data	0	8	128
blurbs	text/image	0	6	96
blurbs_ind	clustered	0	1	16
blurbs_ind	clustered	1	1	16

```
Total_Mbytes
```

```
-----
0.25
```

name	type	total_pages	time_mins
blurbs_ind	clustered	10	0
blurbs	data	6	0

この例は 2K サーバに対して実行されており、blurbs テーブルは 16K サーバにマイグレートされた後は 0.25MB の領域を必要とすることを示しています。次に示すのは 16K サーバに対して実行されている同じクエリであり、0.25MB の領域を必要とすることを確認しています。

```
declare @i int
select @i = avg(datalength(copy)) from blurbs
exec sp_estspace blurbs, 6, null, null, @i
```

name	type	idx_level	Pages	Kbytes
blurbs	data	0	8	128
blurbs	text/image	0	6	96
blurbs_ind	clustered	0	1	16
blurbs_ind	clustered	1	1	16

```
Total_Mbytes
```

```
-----
0.25
```

name	type	total_pages	time_mins
blurbs_ind	clustered	10	0
blurbs	data	6	0

例 5 この例では、2K サーバ上の blurbs テーブルに 1,000 のローがある場合、1.99MB の領域が必要であると予測しています。

```
declare @i int
select @i = avg(datalength(copy)) from blurbs
exec sp_estspace blurbs, 1000, null, null, @i, null, "2k"
```

name	type	idx_level	Pages	Kbytes
blurbs	data	0	16	32
blurbs	text/image	0	1000	2000
blurbs_ind	clustered	0	1	2

blurbs_ind	clustered	1	1	2
Total_Mbytes				

1.99				
name	type	total_pages	time_mins	

blurbs_ind	clustered	18	0	
blurbs	data	1000	0	

使用法

- テーブルとそのインデックスに必要な領域を見積もるには、次の手順に従います。
 - a テーブルを作成します。
 - b テーブル上にすべてのインデックスを作成します。
 - c テーブル名、テーブルに対するローの見積もり数を指定して、**sp_estspace** を実行します。必要に応じてオプションの引数を指定します。

テーブルにデータを挿入する必要はありません。**sp_estspace** は、テーブルとインデックスのサイズの計算に、テーブル内のデータのサイズではなくシステム・テーブル内の情報を使用します。

- **auto identity** オプションがデータベースで設定されると、Adaptive Server は、プライマリ・キー (**primary**)、一意性制約 (**unique**)、または **IDENTITY** カラムを指定しないで作成された新しいテーブルのそれぞれに、10 桁の **IDENTITY** カラムを自動的に定義します。このカラムに必要な追加領域を見積もるには、次の手順に従います。
 - a **master** データベース内で **sp_dboption** を使用してそのデータベースに対応する **auto identity** オプションをオンにします。
 - b テーブルを作成します。
 - c 作成したテーブル上で **sp_estspace** を実行し、結果を記録します。
 - d テーブルを削除します。
 - e データベースの **auto identity** オプションをオフにします。
 - f テーブルを再作成します。
 - g テーブルで **sp_estspace** を再実行し、結果を記録します。
- テーブルまたはカラムについては、**sp_help tablename** を使用してください。

パーミッション `sp_estspace` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド `create index`、`create table`](#)
[システム・プロシージャ `sp_dboption`、`sp_help`](#)

sp_export_qpgroup

説明 指定したユーザのすべてのプランと抽象プラン・グループをユーザ・テーブルにエクスポートします。

構文 `sp_export_qpgroup usr, group, tab`

パラメータ

usr

エクスポートされる抽象プランを所有するユーザの名前です。

group

エクスポートされるプランを収納している抽象プラン・グループの名前です。

tab

プランのコピー先テーブルの名前です。コピー先のテーブルは現在のデータベースのテーブルを選択してください。 `dbname.tablename` の形式で、データベース名を指定できますが、所有者の名前を指定することはできません。識別子の最大長は 255 文字以下にしてください。

例 `ap_stdout` グループ内にあるユーザ “freidak” のすべてのプランを保管する、`moveplans` という名前のテーブルを作成します。

```
sp_export_qpgroup freidak, ap_stdout, "tempdb..moveplans"
```

使用法

- `sp_export_qpgroup` は、抽象プラン・グループからユーザ・テーブルにプランをコピーします。 `sp_import_qpgroup` と一緒に使用すると、サーバとデータベース間で抽象プラン・グループをコピーしたり、コピーされたプランにユーザ ID を割り当てることができます。
- `sp_export_qpgroup` を実行してからでなければ、指定するユーザ・テーブル名は実際には存在しません。このテーブルは、`sysqueryplans` と同じ構造を使用して作成されます。
- `sp_export_qpgroup` は、`select...into` を使用して、コピーされたプランを格納するテーブルを作成します。 `sp_export_qpgroup` を使用するには、また `tempdb` 内でテーブルを作成するには、 `sp_dboption` を使用して `select into/bulkcopy/plsort` を有効にしてください。

パーミッション

`sp_export_qpgroup` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効

細密なパーミッションが有効の場合、`manage abstract plans` 権限を持つユーザであることが必要。

細密なパーミッションが無効

細密なパーミッションが無効の場合、データベース所有者か `sa_role` が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_copy_all_qplans](#)、[sp_copy_qplan](#)、[sp_dboption](#)、[sp_import_qpgroup](#)

sp_extendsegment

説明	セグメントの範囲を別のデータベース・デバイスに拡張します。
構文	<code>sp_extendsegment segname, dbname, devname</code>
パラメータ	<p>segname sp_addsegment によって事前に定義されている既存のセグメントの名前です。</p> <p>dbname セグメントが拡張されるデータベースの名前です。 <i>dbname</i> には、現在のデータベースの名前を指定してください。</p> <p>devname 現在のデータベース・デバイス範囲に追加されるデータベース・デバイスの名前です。このデータベース・デバイス範囲はすでに <i>segname</i> 内に含まれています。</p>
例	<p><code>dev2</code> データベース・デバイスの <code>pubs2</code> データベースに対応する <code>indexes</code> セグメントの範囲を拡張します。</p> <pre>sp_extendsegment indexes, pubs2, dev2</pre>
使用法	<ul style="list-style-type: none"> • セグメントは、複数のデータベース・デバイス間で拡張できます。 • <code>sp_extendsegment</code> は、シングルユーザ・モードで <code>logsegment</code> システム・セグメントに対してのみ実行できます。 • <code>logsegment</code> セグメントが拡張されると、デバイス上の他のセグメントが削除され、そのデバイスはログ・セグメント用デバイスとして排他的に使用されます。 • <code>logsegment</code> セグメントを拡張すると、<code>Adaptive Server</code> は、そのセグメントのラストチャンス・スレッショルドを再計算します。 • セグメントをデータベース・デバイスと対応させるには、そのデバイスへの参照によってデータベースを作成または変更します。データベース・デバイスは、そのデバイスに対応するセグメントを複数持つことができます。 • セグメントを定義した後、そのセグメントを <code>create table</code> コマンドや <code>create index</code> コマンドで使用すると、セグメント上にテーブルまたはインデックスを配置することができます。特定のセグメントにテーブルまたはインデックスを作成すると、そのテーブルまたはインデックス用の後続データが、そのセグメントに配置されます。

パーミッション `sp_extendsegment` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage database</code> 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザーであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド `alter database`](#)、[create index](#)、[create table](#)
システム・プロシージャ [sp_addsegment](#)、[sp_dropsegment](#)、[sp_helpdb](#)、[sp_helpdevice](#)、[sp_helpsegment](#)、[sp_placeobject](#)

sp_extengine

説明	EJB サーバを起動したり停止したりします。また、EJB サーバのステータス情報も表示します。
構文	<code>sp_extengine 'ejb_server', '{ start stop status }'</code>
パラメータ	<code>ejb_server</code> EJB サーバの論理名です。 <code>start</code> EJB サーバを起動します。 <code>stop</code> EJB サーバをシャット・ダウンします。 <code>status</code> EJB サーバのステータス情報を表示します。
例	例 1 EJB サーバの SYB_EJB が動作していることをユーザに通知します。 <pre>sp_extengine 'SYB_EJB', 'status'</pre> Enterprise java bean server is up and running. 例 2 EJB サーバの SYB_EJB をシャット・ダウンします。 <pre>sp_extengine 'SYB_EJB', 'stop'</pre>
使用法	<ul style="list-style-type: none">• <code>sp_extengine</code> を使用するには、有効な Adaptive Server EJB サーバのサイト・ライセンスが必要です。
パーミッション	<code>sp_extengine</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 マニュアル 詳細については、『EJB Server ユーザーズ・ガイド』を参照してください。

sp_extrapwdchecks

説明	パスワード複雑性チェックのユーザ定義ロジックを指定できるカスタム・ストアド・プロシージャです。セキュリティのニーズに応じて、sp_extrapwdchecks を設定できます。master データベースに sp_extrapwdchecks をインストールします。
構文	sp_extrapwdchecks <i>caller_password</i> , <i>new_password</i> , <i>login_name</i>
パラメータ	<i>caller_password</i> 現在のパスワードを指定します。 <i>new_password</i> 新しいパスワードを設定することを指定します。 <i>login_name</i> 変更または追加されているパスワードと関連付けられたログイン名を指定します。
使用法	sp_extrapasswordchecks では、Adaptive Server に対して失敗を通知するために raiserror を使用する必要があります。sp_addmessage を使用して、Adaptive Server にこの失敗のエラー・メッセージを追加します。

注意 予期した動作を取得するために raiserror を使用しないでください。raiserror は、`@@error` グローバル変数を更新します。`@@error` も T-SQL 文 (print と if を含む) を実行するたびに更新されます。raiserror の後に T-SQL 文を指定すると、`@@error` が上書きされ、raiserror の後に TSQL 文が指定されていると sp_extrapwdchecks は失敗したパスワードのエラーを返しません。

sp_familylock

説明 並列に文を実行するファミリー (コーディネーティング・プロセスとそのワーカー・プロセス) が保持しているすべてのロックについての情報をレポートします。

構文 `sp_familylock [fpid1 [, fpid2]]`

パラメータ

fpid1

`master.dbo.sysprocesses` テーブルからのワーカー・プロセス用のファミリー識別子です。 `sp_who` または `sp_lock` を実行して、親プロセスの *spid* を取得します。

fpid2

他のロックの Adaptive Server プロセス ID 番号です。

例 プロセス ID 番号 5 の *fid* を割り当てられたファミリーの全メンバが保持しているロックに関する情報を表示します。

```
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

使用法

- パラメータなしの `sp_familylock` を実行すると、現時点でロックを保持している複数のファミリーに属するすべてのプロセスの情報がレポートされます。このレポートは `sp_lock` から出力されるものと同一です。ただし、`sp_familylock` を使用すると、プロセス ID ではなく、ファミリー ID に基づいたレポートを作成できます。ファミリー・デッドロックの検出に便利です。
- テーブルの ID 番号からテーブル名を検索するには、`object_name` システム関数を使用します。
- “locktype” カラムでは、ロックの種類が共有ロック (“Sh” プレフィックス)、排他ロック (“Ex” プレフィックス)、更新ロックであるかどうかを示し、そのロックが保持されている場所がテーブル上 (“table” または “intent”) またはページ上 (“page”) であるかどうかを示します。

“locktype” カラム内に “blk” サフィックスが表示されている場合は、このプロセスが、ロックを取得する必要がある他のプロセスをブロックしていることを示しています。このプロセスが完了すると、すぐに他のプロセス (1 つまたは複数) が動作を開始します。“demand” サフィックスの場合は、プロセスが排他ロックの取得を試みていることを示します。

- “class” カラムは、ロックがカーソルと対応しているかどうかを示します。次のうちの 1 つが表示されます。
 - “Non cursor lock” が表示されている場合は、ロックがカーソルと対応していません。
 - “Cursor Id number” が表示されている場合は、ロックが、その Adaptive Server プロセス ID のカーソル ID 番号と対応していることを示します。
 - カーソル名が表示されている場合は、ロックが、[sp_lock](#) を実行している現在のユーザが所有している *cursor_name* カーソルと対応していることを示します。
- “fid” カラムでは、ロックが属するファミリー (コーディネーティング・プロセスとそのワーカー・プロセス) が識別されます。“fid” の値は次のとおりです。
 - ゼロの値 : `spid` に表示されているタスクが逐次実行されることを示します。このタスクは並列実行には関与しません。
 - ゼロ以外の値 : ロックを保持しているタスク (`spid`) が、文を並列実行しているプロセス (“fid” のファミリー ID 番号によって識別されているプロセス) のファミリー・メンバであることを示します。この値が `spid` と同じ場合、そのタスクがファミリー内でクエリを並列実行しているコーディネーティング・プロセスであることを示します。
- “context” カラムは、ロックのコンテキストを識別しています。同一のファミリー内のワーカー・プロセスには、同じコンテキスト値が割り当てられます。“context” では、次のような有効値が表示されます。
 - “NULL” : このロックを保持しているタスクが、クエリを逐次実行するタスクなのか、それともトランザクション独立性レベル 1 で並列実行されているクエリなのかを示します。
 - “FAM_DUR” : ロックを保持しているタスクが、クエリが完了するまでロックを保持することを示します。

ロックのコンテキスト値が“FAM_DUR”になる可能性があるのは、ロックが並列クエリの一部として保持されているテーブル・ロックである場合や、ロックがトランザクション独立性レベル3でワーカー・プロセスによって保持されている場合です。または、ロックが並列クエリ内でワーカー・プロセスによって保持されていて、トランザクションの間中そのロックが保持される必要がある場合です。

パーミッション

`sp_familylock` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド `kill`、`select`

システム・プロシージャ `sp_lock`、`sp_who`

sp_find_qplan

説明 クエリ・テキストまたはプラン・テキストから指定されたパターンによって、抽象プランを検索します。

構文 `sp_find_qplan pattern [, group]`

パラメータ *pattern*
クエリまたは抽象プランのテキスト内で検索する文字列です。

group
抽象プラン・グループの名前です。

例 **例 1** クエリ内に “from titles” 文字列があるすべての抽象プランについてレポートします。

```
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 )
)
```

例 2 テーブル・スキャン演算子が含まれているすべてのプランを検出します。

```
sp_find_qplan "%t_scan%"
```

例 3 dev_plans グループのプラン内で、範囲パターン一致を使用して “table1”、“table2”、“table3”～ “table9” の文字列を探します。

```
sp_find_qplan "%table[0-9]%", dev_plans
```

使用法

- 特定の文字列を含む抽象プランを検出するには、**sp_find_qplan** を使用します。クエリ・テキスト、もしくは抽象プラン・テキストからの文字列を照合することができます。
- 一致するプランごとに、**sp_find_qplan** は、グループ ID、プラン ID、クエリ・テキスト、抽象プラン・テキストを出力します。
- グループ名を指定すると、**sp_find_qplan** は、指定したグループ内で文字列を検索します。グループ名を指定しない場合、**sp_find_qplan** は、すべてのグループに対してすべてのクエリとプランを検索します。
- クエリやプランの最初または最後の文字列を検索する場合を除いて、例に示しているように、ワイルドカード文字の “%” を指定してください。例 3 で示している構文のように、任意の Transact-SQL パターン一致構文を使用できます。
- **sysqueryplans** 内のクエリのテキストは、255 バイトのカラム値に分割されます。**sp_find_qplan** は、このカラム値で分割された境界をまたがる一致構文を見逃してしまうことがありますが、127 バイト未満の一致構文であれば、2 つのローにまたがる場合でもすべて検出します。

パーミッション

sp_find_qplan のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	<p>細密なパーミッションが有効の場合、manage abstract plans 権限または monitor qp performance 権限を持つユーザであることが必要。</p> <p>どのユーザも sp_find_qplan を実行して各自のレポート・プランを検索し表示できる。</p>
細密なパーミッションが無効	<p>細密なパーミッションが無効の場合、データベース所有者か sa_role が付与されたユーザであることが必要。</p> <p>どのユーザも sp_find_qplan を実行して各自のレポート・プランを検索し表示できる。</p>

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_help_qpgroup](#)、[sp_help_qplan](#)

sp_fixindex

説明

`sp_fixindex` は、システム・テーブルの破損したインデックスを修復します。テーブル上の指定した 1 つのインデックスまたはすべてのインデックスを再構築できます。ターゲット・テーブルに配置インデックスまたはクラスタード・インデックスが含まれている場合、`sp_fixindex` はデータ・レイヤを再構築します (システム・テーブルの配置インデックスまたはクラスタード・インデックスを処理している間、データ・レイヤの未使用領域を再利用します)。

構文

```
sp_fixindex database_name, table_name [, index_id | null]
           [, index_name | null] [, force_option]
```

パラメータ

dbname

データベース名です。

tablename

テーブル名です。

index_id

修復するインデックスの ID です。

index_name

処理が必要なインデックスを示します。NULL 値を使用すると、*index_id* に関連付けられたインデックスが再構築されます。*index_id* にも NULL 値を使用すると、システム・テーブル内のすべてのインデックスが再構築されます。

force_option

Adaptive Server に、`tempdb` 内のシステム・テーブル・インデックスを強制的に再構築させます。*force_option* なしの `sp_fixindex` では、*database_name* によって指定されたデータベースがシングルユーザ・モードに強制的に設定されます。`tempdb` の場合、シングルユーザ・モードは使用できません。*force_option* を使用すると、`tempdb` 内のシステム・カタログを再構築できますが、ユーザ・データベースには使用しないでください。

例

例 1 `pubs2` データベースの `sysprocedures` テーブル上のクラスタード・インデックスを修復します。

```
sp_fixindex pubs2, sysprocedures, 1
```

例 2 `testdb.sysprocedures` テーブル上のインデックス ID が 2 のインデックスを再構築します。

```
sp_fixindex 'testdb', 'sysprocedures', 2
```

例 3 testdb..sysprocedures システム・テーブル内のインデックス csysprocedures を再構築します。

```
sp_fixindex 'testdb', 'sysprocedures', null,
'csysprocedures'
```

例 4 testdb データベースの sysprocedures テーブル上の使用可能なすべてのインデックスを再構築します。テーブルにクラスタード・インデックスまたは配置インデックスが含まれている場合、sp_fixindex は、データ・ページに存在するガーベジを取り除くことによって未使用領域を再利用します（つまり、データ・ページを再構築します）。

```
sp_fixindex 'testdb', 'sysprocedures'
```

例 5 tempdb..sysprocedures 上のインデックス ID が 2 のインデックスを再構築します。

```
sp_fixindex 'tempdb', 'sysprocedures', 2, null, 1
```

例 6 テーブル tempdb..sysprocedures のインデックス csysprocedures を再構築します。

```
sp_fixindex 'tempdb', 'sysprocedures', null,
'sysprocedures', 1
```

例 7 tempdb データベース内の sysprocedures テーブル上のすべてのインデックスを再構築します。

```
sp_fixindex 'tempdb', 'sysprocedures', null, null, 1
```

使用法

データベースがシングルユーザ・モードになっており、システム・テーブルが更新できるように再設定されていることを確認してから、sp_fixindex を実行します。

sp_fixindex 実行後は、次の処理を行います。

- dbcc checktable コマンドを使用して、矛盾したインデックスが修復されていることを確認します。
- sp_configure を使用して、システム・テーブルへの更新を禁止します。
- シングルユーザ・モードを解除します。

ユーザ・テーブルに対して sp_fixindex を実行しないでください。

警告！ sp_fixindex は、sysindexes のクラスタード・インデックスに対しては使用できません。実行すると、sp_fixindex によって次のようなエラー・メッセージが返されます。

```
Cannot re-create index on this table.
```

パーミッション `sp_fixindex` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データベース所有者か <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 **マニュアル** `sp_fixindex` の詳細については、以下を参照してください。

- 『トラブルシューティング&エラー・メッセージ・ガイド』の「第2章 タスクについて」
- 『パフォーマンス&チューニング・ガイド：基本』の「第13章 インデックスとパフォーマンス」

sp_flushstats

説明	メモリ内記憶領域の統計を systabstats システム・テーブルと sysstatistics システム・テーブルにフラッシュします。
構文	sp_flushstats [<i>objname</i>]
パラメータ	<i>objname</i> テーブルの名前です。
例	titles テーブルの統計値をフラッシュします。 <pre>sp_flushstats titles</pre>
使用法	<ul style="list-style-type: none"> • objname パラメータでテーブルを指定しない場合、sp_flushstats はデータベース・レベルで機能します。 • systabstats テーブル内の一部の統計値は、インメモリ記憶ロケーションで更新されて、systabstats に定期的にフラッシュされ、systabstats でのオーバーヘッドと競合を減らします。 • SQL を使用して systabstats を問い合わせる場合、sp_flushstats を実行すると、メモリ内の統計値は確実に systabstats にフラッシュされます。 • optdiag コマンドは、必ずインメモリ統計をフラッシュしてから、出力を表示します。 • sysstatistics 内の統計値は、データ定義言語コマンドによってのみ変更されます。sp_flushstats を使用する必要はありません。 • メモリ内 datachange カウンタは、sysstatistics に永続的に保持されています。これらは、sp_flushstats の実行時にディスクにフラッシュされます。
パーミッション	sp_flushstats のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 monitor qp performance 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 sa_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

sp_forceonline_db

説明 以前リカバリを実行したときに疑わしいページとして **suspect** マークが付けられた、データベース内に格納されているすべてのページにアクセスできるようにします。

構文 `sp_forceonline_db dbname, {"sa_on" | "sa_off" | "all_users"}`

パラメータ `dbname`
オンラインにするデータベースの名前です。

sa_on
`sa_role` の役割を持つユーザだけに、指定したページへのアクセスを許可します。

sa_off
`sa_on` が以前に呼び出した `sp_forceonline_page` によって作成されたアクセス権限を取り消します。

all users
すべてのユーザに、指定ページへのアクセスを許可します。

例 **例 1** システム管理者に、`pubs2` データベース内にあるすべての疑わしいページへのアクセスを許可します。

```
sp_forceonline_db pubs2, "sa_on"
```

例 2 `pubs2` データベース内にあるすべての **suspect** ページへのアクセス権限を、システム管理者から取り消します。これで、`pubs2` 内の **suspect** ページに誰もアクセスできなくなります。

```
sp_forceonline_db pubs2, "sa_off"
```

例 3 すべてのユーザに、`pubs2` データベース内に格納されているすべてのページへのアクセスを許可します。

```
sp_forceonline_db pubs2, "all_users"
```

使用法

- 強制的にオンラインにされたページが必ずしも修復されるとは限りません。また、破壊されたページを強制的にオンラインにすることもできますが、**Adaptive Server** は、これらのページに対して一貫性の検査を実行しません。
- `all users` によって実行された `sp_forceonline_page` は、リバースできません。ページがすべてのユーザに対してオンラインになると、それらのページを再度オフラインにすることはできません。
- `sp_forceonline_db` は、トランザクション内で使用できません。

- 特定のオフライン・ページだけをオンラインにするには、[sp_forceonline_page](#) を使用します。

パーミッション

`sp_forceonline_db` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データベース所有者か <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_forceonline_page](#)、[sp_listsuspect_db](#)、[sp_listsuspect_page](#)、[sp_setsuspect_granularity](#)、[sp_setsuspect_threshold](#)

sp_forceonline_object

説明	以前リカバリを実行したときに疑わしいインデックスとして、 <code>suspect</code> マークが付けられたインデックスにアクセスできるようにします。
構文	<code>sp_forceonline_object dbname, objname, indid, {sa_on sa_off all_users} [, no_print]</code>
パラメータ	<p>dbname オンラインにするインデックスを格納しているデータベースの名前です。</p> <p>objname テーブルの名前を指定します。</p> <p>indid オンラインにする <code>suspect</code> (疑わしい) インデックスのインデックス ID です。</p> <p>sa_on <code>sa_role</code> の役割を持つユーザだけに、指定したインデックスへのアクセスを許可します。</p> <p>sa_off <code>sa_on</code> が以前に呼び出した <code>sp_forceonline_object</code> によって作成されたアクセス権限を取り消します。</p> <p>all_users すべてのユーザに、指定インデックスへのアクセスを許可します。</p> <p>no_print 指定したオブジェクトがオンラインになった後、他の <code>suspect</code> (疑わしい) オブジェクトのリストを出力することを省略します。</p>
例	<p>例 1 システム管理者が、<code>pubs2</code> データベース内の <code>titles</code> テーブル上で、<code>indid 3</code> インデックス ID が付いているインデックスにアクセスできるようにします。</p> <pre>sp_forceonline_object pubs2, titles, 3, sa_on</pre> <p>例 2 インデックスへのアクセス権限をシステム管理者から取り消します。これで、このインデックスに誰もアクセスできなくなります。</p> <pre>sp_forceonline_object pubs2, titles, 3, sa_off</pre> <p>例 3 すべてのユーザに、<code>pubs2</code> データベース内の <code>titles</code> テーブル上にあるインデックスへのアクセスを許可します。</p> <pre>sp_forceonline_object pubs2, titles, 3, all_users</pre>

使用法

- データオンリー・ロック・テーブルのインデックスに `suspect` ページがある場合は、リカバリ中のインデックス全体がオフラインになります。オフラインのインデックスは、クエリ・オプティマイザの対象になりません。リカバリ中に、すべてのページ・ロック・テーブルのインデックスが完全にオフラインになるわけではありません。これらのインデックスの個々のページだけがオフラインになります。`sp_forceonline_page` を使用して、これらのページをオンラインにします。
- `sp_listsuspect_object` を使用して、オフラインのデータベースのリストを表示します。
- `suspect` (疑わしい) インデックスを修復するには、`sa_on` アクセスによって、`sp_forceonline_object` を使用してください。次に、インデックスを削除してから再作成します。

注意 インデックスが `systabstats` または `sysstatistics` (データオンリー・ロック・システム・テーブルのみ) にある場合は、Sybase 製品の保守契約を結んでいるサポート・センタに問い合わせてください。

- `all_users` によって実行された `sp_forceonline_object` をリバースすることはできません。あるインデックスがすべてのユーザに対してオンラインになると、そのインデックスを再度オフラインにすることはできません。
- 破損したインデックスも強制的にオンラインにできるため、強制的にオンラインにされたインデックスが必ずしも修復されるとは限りません。Adaptive Server は、強制的にオンラインにされたインデックスでは一貫性チェックを実行しません。
- `sp_forceonline_object` をトランザクション内で使用することはできません。
- `sp_forceonline_page` は、リカバリ・フォールト・アイソレーション・モードが "page" であるデータベースにのみ有効です。`sp_setsuspect_granularity` を使用して、データベースのリカバリ・フォールト・アイソレーション・モードを表示してください。
- データベースのオフライン・ページとオフライン・インデックスを 1 つのコマンドですべてオンラインにするには、`sp_forceonline_db` を使用します。

パーミッション sp_forceonline_object のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データベース所有者か own database 権限を持つユーザであることが必要。
---------------	--

細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。
---------------	---

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照 マニュアル リカバリ・フォールト・アイソレーションの詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_listsuspect_object](#)、[sp_setsuspect_granularity](#)

sp_forceonline_page

説明	リカバリによって、以前に疑わしいページとして <code>suspect</code> のマークが付けられたページにアクセスできるようにします。
構文	<code>sp_forceonline_page dbname, pgid, {"sa_on" "sa_off" "all_users"}</code>
パラメータ	<p><code>dbname</code> オンラインにするページを格納しているデータベースの名前です。</p> <p><code>pgid</code> オンラインにするページのページ識別子です。</p> <p><code>sa_on</code> <code>sa_role</code> の役割を持つユーザだけに、指定したページへのアクセスを許可します。</p> <p><code>sa_off</code> <code>sa_on</code> が以前に呼び出した <code>sp_forceonline_page</code> によって作成されたアクセス権限を取り消します。</p> <p><code>all_users</code> すべてのユーザに、指定ページへのアクセスを許可します。</p>
例	<p>例 1 システム管理者に、<code>pubs2</code> データベース内にある 312 ページへのアクセスを許可します。</p> <pre>sp_forceonline_page pubs2, 312, "sa_on"</pre> <p>例 2 <code>pubs2</code> データベース内にある 312 ページへのアクセス権限を、システム管理者から取り消します。これで、誰も 312 ページにアクセスできなくなります。</p> <pre>sp_forceonline_page pubs2, 312, "sa_off"</pre> <p>例 3 すべてのユーザに、<code>pubs2</code> データベース内にある 312 ページへのアクセスを許可します。</p> <pre>sp_forceonline_page pubs2, 312, "all_users"</pre>
使用法	<ul style="list-style-type: none"> <code>all_users</code> によって実行された <code>sp_forceonline_page</code> をリバースすることはできません。ページがすべてのユーザに対してオンラインになると、それらのページを再度オフラインにすることはできません。 強制的にオンラインにされたページが必ずしも修復されるとは限りません。また、破壊されたページを強制的にオンラインにすることもできますが、Adaptive Server は、これらのページに対して一貫性チェックを実行しません。

- sp_forceonline_page をトランザクション内で使用することはできません。
- sp_forceonline_page は、リカバリ・フォールト・アイソレーション・モードが "page" であるデータベースにのみ有効です。
[sp_setsuspect_granularity](#) を使用して、データベースのリカバリ・フォールト・アイソレーション・モードを表示してください。
- 1つのコマンドですべてのデータベースのオフライン・ページをオンラインにするには、[sp_forceonline_db](#) を使用します。

パーミッション

sp_forceonline_page のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データベース所有者か own database 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_forceonline_db](#)、[sp_listsuspect_db](#)、[sp_listsuspect_page](#)、[sp_setsuspect_granularity](#)、[sp_setsuspect_threshold](#)

sp_foreignkey

説明	現在のデータベース内にあるテーブルまたはビューの外部キーを定義します。
構文	<code>sp_foreignkey <i>tablename</i>, <i>pktablename</i>, <i>col1</i> [, <i>col2</i>] ... [, <i>col8</i>]</code>
パラメータ	<p><i>tablename</i> 定義する外部キーが含まれているテーブルまたはビューの名前です。</p> <p><i>pktablename</i> 外部キーが適用されるプライマリ・キーを持っているテーブルまたはビューの名前です。プライマリ・キーは事前に定義しておいてください。</p> <p><i>col1</i> 外部キーを構成する最初のカラムの名前です。外部キーには 1 つ以上のカラムが必要です。最大 8 つのカラムを割り当てることができます。</p>
例	<p>例 1 <code>publishers</code> テーブルのプライマリ・キーは <code>pub_id</code> カラムです。 <code>titles</code> テーブルにも <code>pub_id</code> カラムがありますが、これは <code>publishers</code> の外部キーです。</p> <pre>sp_foreignkey titles, publishers, pub_id</pre> <p>例 2 <code>parts</code> テーブルのプライマリ・キーは、sp_primarykey によって、<code>partnumber</code> カラムと <code>subpartnumber</code> カラムとして定義されています。 <code>orders</code> テーブルには <code>part</code> カラムと <code>subpart</code> カラムがあります。これらは <code>parts</code> の外部キーを構成します。</p> <pre>sp_foreignkey orders, parts, part, subpart</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_foreignkey</code> は、<code>syskeys</code> テーブルにキーを追加します。キーは、データベース設計での暗黙的な論理関係を、明示的にするために作成されます。 • <code>sp_foreignkey</code> は参照整合性制約を実行しません。外部キーの関係を適用するには、<code>create table</code> コマンドまたは <code>alter table</code> コマンドの <code>foreign key</code> 句を使用してください。 • 外部キーを構成するカラムの数および順序は、プライマリ・キーを構成するカラムの数および順序と同じにしてください。プライマリ・キーと外部キーのデータ型 (および長さ) は一致していなければなりません。 <code>null</code> データ型が一致している必要はありません。

- インストール処理では、システム・テーブルの該当するカラムで `sp_foreignkey` が実行されます。
- 定義されているキーについてのレポートを表示するには、[sp_helpkey](#) を実行します。
- `sp_foreignkey` では、Java データ型を使用できません。

パーミッション

`sp_foreignkey` を実行するには、テーブルまたはビューの所有者である必要があります。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [alter table](#)、[create table](#)、[create trigger](#)

システム・プロシージャ [sp_commonkey](#)、[sp_dropkey](#)、[sp_helpjoins](#)、[sp_helpkey](#)、[sp_primarykey](#)

sp_freedll

説明	ESP (拡張スタアド・プロシージャ) の実行をサポートするために XP Server メモリ内に事前にロードされている DLL (ダイナミック・リンク・ライブラリ) をアンロードします。
構文	<code>sp_freedll dll_name</code>
パラメータ	<code>dll_name</code> XP Server メモリからアンロードされる DLL のファイル名です。
例	<code>sqlsrvdll.dll</code> DLL をアンロードします。 <pre>sp_freedll "sqlsrvdll.dll"</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_freedll</code> をトランザクション内で実行することはできません。 • <code>sp_freedll</code> は、システム ESP の DLL を解放できません。 • <code>sp_freedll</code> を使用して DLL を明示的にアンロードするもう 1 つの方法は、DLL を呼び出した ESP 要求の後で常にアンロードされる DLL が終了するように指定します。この方法を使用するには、<code>esp unload dll</code> 設定パラメータを 1 に設定するか、または <code>-u</code> オプションで <code>xserver</code> を起動します。 • <code>sp_freedll</code> を使用すると、XP Server と Adaptive Server を停止させずに DLL 内の ESP 関数を更新できます。 • <code>sp_freedll</code> によって、使用中の DLL をアンロードすると、<code>sp_freedll</code> が正常に実行されます。これによって、現在 DLL を使用している ESP が実行できなくなります。
パーミッション	<code>sp_freedll</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any ESP</code> 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザーであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addextendedproc](#)、[sp_dropextendedproc](#)、[sp_helpextendedproc](#)

sp_getmessage

説明	print 文と raiserror 文で使用するために、sysmessages と sysusermessages から、格納されているメッセージ文字列を検索します。
構文	sp_getmessage <i>message_num</i> , <i>result output</i> [, <i>language</i>]
パラメータ	<p><i>message_num</i> 検索されるメッセージの番号です。</p> <p><i>result output</i> リターン・メッセージのテキストを受け取る変数です。この変数の後にスペースと output キーワードが続きます。この変数に割り当てなければならないデータ型は、char、unichar、nchar、varchar、univarchar、nvarchar です。</p> <p><i>language</i> 検索されるメッセージの言語です。language には、syslanguages テーブル内の有効な言語名を指定してください。language を指定すると、示された <i>message_num</i> と language を持つメッセージが検索されます。language を指定しない場合は、変数 @@langid によって指示されているデフォルト・セッション言語のメッセージが検索されます。</p>
例	<p>例 1 sysusermessages から 20001 番のメッセージを検索します。</p> <pre>declare @myvar varchar(200) exec sp_getmessage 20001, @myvar output</pre> <p>例 2 sysusermessages から 20010 番のフランス語バージョンのメッセージを検索します。</p> <pre>declare @myvar varchar(200) exec sp_getmessage 20010, @myvar output, french</pre>
使用法	<ul style="list-style-type: none">sp_getmessage はどのアプリケーションでも使用できます。また、sysmessages と sysusermessages に保管されているメッセージは、どのユーザでも読み込むことができます。
パーミッション	sp_getmessage は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none">• <i>Roles</i> – 現在のアクティブな役割• <i>Keywords or options</i> – NULL• <i>Previous value</i> – NULL• <i>Current value</i> – NULL• <i>Other information</i> – すべての入力パラメータ• <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [print](#)、[raiserror](#)

システム・プロシージャ [sp_addmessage](#)、[sp_dropmessage](#)

sp_grantlogin

説明	Windows のみ — 統合セキュリティ・モードまたは混合モード (名前付きパイプ使用) がアクティブなとき、Windows のユーザとグループに、Adaptive Server の役割または default パーミッションを割り当てます。
構文	<pre>sp_grantlogin {login_name group_name} ["role_list" default]</pre>
パラメータ	<p><i>login_name</i> Windows NT ユーザのネットワーク・ログイン名です。</p> <p><i>group_name</i> Windows NT グループ名です。</p> <p><i>role_list</i> 付与される Adaptive Server 役割のリストです。役割のリストには、sa_role、sso_role、oper_role の役割名が 1 つ以上あります。sa_role、sso_role、oper_role. 複数の役割を指定する場合は、役割名をカンマではなくスペースで区切ってください。</p> <p>default <i>login_name</i> または <i>group_name</i> が、grant 文または sp_role プロシージャによって割り当てられたデフォルトのパーミッションを受け取るように指定します。</p>
例	<p>例 1 Adaptive Server の oper_role を、Windows NT ユーザ “jeanluc” に割り当てます。</p> <pre>sp_grantlogin jeanluc, oper_role</pre> <p>例 2 default 値を Windows NT ユーザ “valle” に割り当てます。ユーザ “valle” は、grant コマンドまたは sp_role プロシージャを介して割り当てられたパーミッションを受け取ります。</p> <pre>sp_grantlogin valle</pre> <p>例 3 Adaptive Server の sa_role と sso_role を、Windows NT の管理者グループのすべてのメンバに割り当てます。</p> <pre>sp_grantlogin Administrators, "sa_role sso_role"</pre>
使用法	<ul style="list-style-type: none">Windows NT のログイン名またはグループを作成してから、sp_grantlogin によって役割を割り当ててください。詳細については、Windows NT のマニュアルを参照してください。

- `sp_grantlogin` は、Adaptive Server が統合セキュリティ・モードまたは混合モード (接続が名前付きパイプの場合) で実行されているときだけ、アクティブになります。標準モード、または名前付きパイプ以外の接続による混合モードで Adaptive Server が実行されている場合は、代わりに `grant` と `sp_role` を使用します。
- `role_list` または `default` を指定しなければ、プロシージャは自動的に `default` 値を割り当てます。
- `default` 値は、Adaptive Server の役割を指示しません。 `default` 値によって、ユーザまたはグループが、`grant` コマンドまたは `sp_role` プロシージャを介して割り当てられたパーミッションを受け取るように指定されます。
- 既存の `login_name` または `group_name` とともに `sp_grantlogin` を使用すると、そのユーザまたはグループの既存の役割が上書きされます。

パーミッション

`sp_grantlogin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage roles</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド `grant`、`setuser`

システム・プロシージャ `sp_addlogin`、`sp_displaylogin`、`sp_droplogin`、`sp_locklogin`、`sp_logininfo`、`sp_modifylogin`、`sp_revokelogin`、`sp_role`

sp_ha_admin

説明	高可用性システムでの Sybase のフェールオーバ機能を使用して、Adaptive Server の管理作業を実行します。sp_ha_admin は、UNIX プラットフォームでは <i>installhavss</i> スクリプトによってインストールされます。Windows NT の場合は、 <i>insthasv</i> スクリプトによってインストールされます。
構文	sp_ha_admin [cleansessions help]
パラメータ	<p>cleansessions</p> <p>古いエントリを <i>sysessions</i> から削除します。古い <i>sysessions</i> エントリが残る原因は、通常の場合、リポート中に Adaptive Server が <i>sysessions</i> をクリーンアップできなかったか、またはクライアントが Adaptive Server との接続に失敗したかのどちらかです。</p> <p>help</p> <p>sp_ha_admin の構文を表示します。</p>
例	<p>例 1 クライアント接続が正しく終了しなかったことが原因で残された古いエントリを、<i>sysessions</i> から削除します。</p> <pre>sp_ha_admin cleansessions (return status = 0)</pre> <p>例 2 sp_ha_admin の構文を表示します。</p> <pre>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)</pre>
使用法	<ul style="list-style-type: none"> sp_ha_admin は、高可用性システムで Sybase のフェールオーバ機能用に設定された管理作業を Adaptive Server 上で実行します。sp_ha_admin は、<i>installmaster</i> スクリプトではインストールされません。代わりに <i>installhavss</i> スクリプトを使用して Sybase のフェールオーバをインストールし、Sybase のフェールオーバ用の設定を行います (Windows NT では <i>insthasv</i> を使用します)。 sp_ha_admin は、<i>sysessions</i> を正常にクリーンアップした場合は 0 を返し、エラーが発生した場合は 1 を返します。 sp_ha_admin は、<i>sysessions</i> からエントリを削除できない場合 (たとえば、<i>sysessions</i> をロックできない場合) には、エラーログにメッセージを書き込みます。

- `sysessions` 内にある現在のエントリをすべて表示するには、次のように入力します。

```
select * from sysessions
```

パーミッション

`sp_ha_admin` を実行するには、`ha_role` が付与されたユーザであることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

sp_help

説明 データベース・オブジェクト (**sysobjects** にリストされている任意のオブジェクト) に関する情報、システム定義データ型またはユーザ定義データ型に関する情報、ユーザ定義関数、および計算カラムと関数ベースのインデックスについてレポートします。カラムに **optimistic_index_lock** が表示されます。

構文 `sp_help [objname]`

パラメータ *objname*

systypes 内にある任意のオブジェクトの名前です。または、**sysobjects** 内の任意のユーザ定義データ型またはシステム・データ型の名前です。データベース名は指定できません。*objname* にはテーブル、ビュー、事前計算済み結果セット、ストアド・プロシージャ、ログ、ルール、デフォルト、トリガ、参照整合性制約、暗号化キー、述部、検査制約も指定できます。ただし、**optimistic_index_lock** を有効にすると、*objname* はテーブルを参照します。オブジェクト所有者がコマンドを実行しているユーザでない場合、またデータベース所有者でもない場合は、所有者名を使用してください。

例 **例 1** **sysobjects** 内のオブジェクトのリストを表示し、各オブジェクトの名前、所有者、オブジェクト・タイプも表示します。**systypes** 内にあるユーザ定義の各データ型のリストも表示して、データ型の名前、記憶領域の種類、長さ、**null** データ型の種類、デフォルト名、ルール名を示します。**null** データ型の種類は、0 (**null** 値の禁止) か 1 (**null** 値の許容) の 2 つです。

`sp_help`

例 2 分割された **publishers** テーブルについての情報を表示します。また、**sp_help** も指定されたテーブルとそのインデックスに割り当てられた属性をリストします。その内容は、属性のクラス、名前、整数値、文字値、コメントです。

`sp_help publishers`

Name	Owner	Object_Type	Create_date
publishers	dbo	user table	Oct 7 2005 11:14AM

Column_name	Type	Length	Prec	Scale	Nulls	Default_name	Rule_name
pub_id	char	4	NULL	NULL	0	NULL	pub_idrule

pub_name	varchar	40	NULL	NULL	1	NULL
	NULL			NULL	0	
city	varchar	20	NULL	NULL	1	NULL
	NULL			NULL	0	
state	char	2	NULL	NULL	1	NULL
	NULL			NULL	0	

Object does not have any indexes.

keytype	object	related_objs	object_keys	related_keys
primary	publishers	-- none --	pub_id,*,*,*,*,*,*,*	*,*,*,*,*,*,*,*

name	type	partition_type	partitions	partition_keys
publishers	base table	roundrobin	3	NULL

partition_name	partition_id	pages	segment	create_date
publishers_608002166	608002166	1	default	Oct 13 2005 11:18AM
publishers_1116527980	1116527980	1	default	Oct 13 2005 11:18AM
publishers_1132528037	1132528037	1	default	Oct 13 2005 11:19AM

Partition_Conditions

NULL

Avg_pages	Max_pages	Min_pages	Ratio(Max/Avg)	Ratio(Min/Avg)
1	1	1	1.0000000	1.0000000

Lock scheme Allpages

The attribute 'exp_row_size' is not applicable to tables with allpages lock scheme.

exp_row	reservepagegap	fillfactor	max_rows_per_page	identity_gap
0	0	0	0	0

concurrency_opt_threshold	optimistic_index_lock	dealloc_first_txtpg
0	0	0

例 3 分割された titles テーブルに関する情報を表示します。

sp_help titles

Name	Owner	Object_Type	Create_date
titles	db	user table	Oct 7 2005 11:14AM

(1 row affected)

Column_name	Type	Length	Prec	Scale	Nulls	Default_name	Rule_name	Access_Rule_name
title_id	tid	6	NULL	NULL	0	NULL	title_idrule	NULL
title	varchar	80	NULL	NULL	0	NULL	NULL	NULL
type	char	12	NULL	NULL	0	typedflt	NULL	NULL
pub_id	char	4	NULL	NULL	1	NULL	NULL	NULL
price	money	8	NULL	NULL	1	NULL	NULL	NULL
advance	money	8	NULL	NULL	1	NULL	NULL	NULL
total_sales	int	4	NULL	NULL	1	NULL	NULL	NULL
notes	varchar	200	NULL	NULL	1	NULL	NULL	NULL
pubdate	datetime	8	NULL	NULL	0	datedflt	NULL	NULL
contract	bit	1	NULL	NULL	0	NULL	NULL	NULL
index_name	index_description				index_keys			

Object has the following indexes

index_name	index_keys	index_description	index_max_rows_per_page
index_fillfactor	index_reservepagegap	index_created	index_local
title_idx	total_sales	clustered	0
	0	0 Oct 13 2005 5:20PM	Local Index

index_ptn_name	index_ptn_seg
p1	default
p2	default
p3	default
title_idx_98505151	default

keytype	object	related_object	object_keys
related_keys			
foreign	roysched	titles	title_id, *, *, *, *, *, *, * title_id, *, *, *, *, *, *, *
foreign	salesdetail	titles	title_id, *, *, *, *, *, *, * title_id, *, *, *, *, *, *, *
foreign	titleauthor	titles	title_id, *, *, *, *, *, *, * title_id, *, *, *, *, *, *, *

```

*, *, *, *
foreign titles publishers pub_id, *, *, *, *, *, *, *
pub_id, *, *, *, *, *, *, *
primary titles -- none -- title_id, *, *, *, *, *, *, *
*, *, *, *, *, *, *, *

```

```

name type partition_type partitions partition_keys
-----

```

```

titles base table range 4 pubdate

```

```

partition_name partition_id pages segment create_date
-----
q1 937051343 1 default Oct 13 2005 5:20PM
q2 953051400 1 default Oct 13 2005 5:20PM
q3 969051457 1 default Oct 13 2005 5:20PM
q4 985051514 1 default Oct 13 2005 5:20PM

```

Partition_Conditions

```

-----
VALUES <= ("3/31/2006")
VALUES <= ("6/30/2006")
VALUES <= ("9/30/2006")
VALUES <= ("12/31/2006")
VALUES <= ("3' 31' 2006")

```

```

Avg_pages Max_pages Min_pages Ratio(Max/Avg) Ratio(Min/Avg)
-----
1 1 1 1.000000 1.000000

```

Lock scheme Allpages

The attribute 'exp_row_size' is not applicable to tables with allpages lock scheme.

```

exp_row reservepagegap fillfactor max_rows_per_page identity_gap
-----
0 0 0 0 0

```

```

concurrency_opt_threshold optimistic_index_lock dealloc_first_txtpg
-----
0 0 0

```

例 4 ユーザ “mary” が所有している marytrig トリガについての情報を表示します。ピリオドは特殊文字の1つであるため、引用符が必要です。

```
sp_help "mary.marytrig"
```

```

Name Owner Object_type
-----
marytrig mary trigger

```

```

Data_located_on_segment  When_created
-----
not applicable           Mar 20 2002  2:03PM

```

例5 システム・データ型の `money` に関する情報を表示します。

```
sp_help money
```

```

Type_name  Storage_type  Length  Prec  Scale  Nulls  Defaul_name
-----
Rule_name  Access_Rule_name  Identity
-----
money      money             8  NULL  NULL  1      NULL
NULL      NULL             NULL   0

```

例6 ユーザ定義データ型の `identye` に関する情報を表示します。レポートが表示する内容は、ユーザ定義データ型の作成に使用された基本データ型、ユーザ定義データ型が `null` を許容するかどうか、このデータ型にバインドされたすべてのルールとデフォルトの名前、このデータ型に `IDENTITY` プロパティが割り当てられているかどうかについての情報です。

```
sp_help identye
```

```

Type_name  Storage_type  Length  Prec  Scale  Nulls  Defaul_name
-----
Rule_name  Access_Rule_name  Identity
-----
identye    numeric       4  NULL  NULL  1      NULL
NULL      NULL          NULL   1

```

新しいカラムを表示します。オプティミスティック・インデックス・ロックが有効かどうかを示されます。1はロックが有効、0は無効を示します。

```
sp_help "mytable"
```

```

-----
exp_row_size  reserve  pagegap  fillfactor  max_rows_per_page
-----
1            0        0         0            0
concurrency_opt_threshold  optimistic_index_lock
-----
0            1

```

例7 仮想計算カラムを表示します。

```

alter table authors add fullname as au_fname + ' ' + au_lname
sp_help authors

```

Object has the following computed columns

```
Column_Name Property
-----
```

```
fullname    virtual
```

```
Text
-----
```

```
AS au_fname + ' ' + au_lname
```

例 8 仮想の計算カラムを、実体化された計算カラムに対して示します。

```
alter table authors modify fullname materialized
sp_help authors
```

Object has the following computed columns

```
Column_Name Property
-----
```

```
fullname    materialized
```

```
Text
-----
```

```
AS au_fname + ' ' + au_lname
MATERIALIZED
```

例 9 sp_help *table_name* の結果セットには、Decrypt_Default_name カラムが含まれ、これは、カラムの復号化デフォルト名を示します。たとえば、次のコマンドを実行したとします。

```
create table encr_table(coll int encrypt decrypt_default 1)
```

encr_table に対して sp_help を実行すると、次の内容が表示されます。

```
Column_name Type Length Prec Scale Nulls Default_name Rule_name Access_Rule_name
Computed_Column_object Identity Encrypted Decrypt_Default_name
-----
```

```
-----
c1          int          4 NULL NULL    0 NULL          NULL          NULL
NULL                                     0              1 encr_table_coll_1036527695
```

例 10 述部オブジェクトの Name、Owner、Object_type、Object_status、Create_date を表示します。

```
grant select on tabl where coll = 5 as pred1 to robert
sp_help pred1
```

```
Name Owner Object_type Object_status Create_date
-----
```



```
pred1 dbo predicate -- none -- Feb 9 2010 12:49PM
```

例 11 次のように事前計算済み結果セットを作成します。

```
create table numtrips (source int, destination int, count_trip int)
create precomputed result set frequent_trips unique (source, destination)
as
select * from numtrips where count_trip > 100
```

この場合、

`sp_help numtrips` は次のような結果を返します。

```
Name          Owner          Object_type
Object_status
Create_date
-----
numtrips      dbo            user table
precomputed result set defined
May 11 2012  6:46AM
. . .
```

`sp_help frequent_trips` は次のような結果を返します。

```
Name          Owner
Object_type
Object_status
Create_date
-----
frequent_trips      dbo
precomputed result set
immediate, enabled, enabled for QRW
May 11 2012  6:46AM
. . .
```

例 12 次のように、`sp_help` は `Object` ステータス・フィールドに `execute as owner` または `execute as caller` を表示します。

```
create proc p1 with execute as owner as
select 1
go
sp_help p1
Name Owner Object_type Object_status Create_date
```

```

-----
pl          dbo          stored procedure
execute as owner

Jun  8 2012 10:05AM
(1 row affected)

Column_name  Type Length Prec Scale Nulls Not_compressed  Default_name
Rule _name  Access_Rule_name  Computed_Column_object  Identity
-----
(return status = 0) Rule_name

```

使用法

- 仮想ハッシュ・テーブルの場合、sp_help は次の内容を報告します。
 - 次のメッセージによって、テーブルが仮想的にハッシュされていることを報告します。


```
Object is Virtually Hashed
```
 - 次の構文を使用したメッセージによって、hash_key_factors をレポートします。


```
column_1:hash_factor_1,
column_2:hash_factor_2...,
max_hash_key=max_hash_value
```

次に例を示します。

attribute_class	attribute	int_value	comments
hash clustered tables	hash key factors		NULL
id:10.0, id2:1.0, max_hash_key=1000.0		NULL	

- `sp_help` は、現在のデータベース内にあるオブジェクトだけを検索します。
- `sp_help` は、次のような Adaptive Server のルールに従ってオブジェクトを検索します。
 - ユーザが所有者名を指定しなくても、所有者名が指定されているオブジェクトをユーザが所有していれば、`sp_help` はそのオブジェクトについてレポートします。
 - ユーザが所有者名を指定せず、その名前のオブジェクトを所有していなくても、データベース所有者が所有していれば、`sp_help` はデータベース所有者のオブジェクトについてレポートします。
 - ユーザもデータベース所有者も、指定された名前のオブジェクトを所有していない場合は、別の所有者がデータベース内にその名前のオブジェクトを所有していたとしても、`sp_help` はエラー状態をレポートします。自分とデータベース所有者以外のデータベース・ユーザが所有しているオブジェクトは、その所有者名で修飾してください。例 4 を参照してください。
 - ユーザとデータベース所有者の両方が、指定した名前のオブジェクトを所有しているときに、ユーザがデータベース所有者のオブジェクトにアクセスする場合は、その名前を `dbo.objectname` の形式で指定します。
- `sp_help` は、事前計算済み結果セット・オブジェクトに関する情報を `Object_type` カラムに表示します。Adaptive Server 内部では、事前計算済み結果セット・オブジェクトがユーザ・テーブルとして扱われます。`objectname` に事前計算済み結果セット・オブジェクトを指定して `sp_help` を発行すると、ユーザ・テーブルに対して `sp_help` を実行した場合と同じように、カラム、パーティション、キー、インデックスなどの関連する詳細がすべてレポートされます。

さらに、`Object_Status` カラムは事前計算済み結果セットについて次の情報を返します。

- ユーザ・テーブルの場合 — ユーザ・テーブルで事前計算済み結果セットが定義されていれば、`Object_Status` カラムに `precomputed result set defined` が表示されます。
- 事前計算済み結果セット・オブジェクトの場合 — `Object_Status` カラムに次の情報が表示されます。
 - 1 リフレッシュ・モード — `immediate` または `manual`
 - 2 事前計算済み結果セットの状態 — `enabled` または

disabled

3 クエリ再書き込みの状態 — enable for QRW または disabled for QRW

前述の例 11 を参照してください。

- tempdb から sp_help を実行すると、このコマンドはテンポラリ・テーブル上で機能します。
- IDENTITY プロパティ付きのカラムには、“Identity” 値 1 が割り当てられます。他のカラムには、“Identity” 値 0 が割り当てられます。例 2 では、IDENTITY カラムはありません。
- sp_help は、テーブル上のすべてのインデックスをリストします。これには、create table 文または alter table 文で一意性制約またはプライマリ・キーの制約を定義して作成されたインデックスも含まれます。また、インデックスに対応している属性もリストされます。ただし、sp_help は、テーブル用に定義された整合性制約についての情報は示しません。整合性制約の情報については、[sp_helpconstraint](#) を使用してください。
- sp_help は、次の新しい設定内容を表示します。
 - ロック・スキーム：create table によって設定され、alter table によって変更されます。
 - 予想されるロー・サイズ：create table によって設定され、sp_chgattribute によって変更されます。
 - ページ・ギャップの予約：create table によって設定され、sp_chgattribute によって変更されます。
 - ロー・ロック・プロモーションの設定：sp_setpglockpromote によって設定または変更され、sp_dropprowlockpromote によって削除されます。
- sp_help には、以下のレポートが含まれています。
 - [sp_helpindex](#) — インデックスと記憶領域管理プロパティの作成に使用されるキーの順序が示されます。
 - [sp_helppartition](#) — テーブルのパーティション情報が示されます。
 - [sp_helpcomputedcolumn](#) — テーブルの計算カラム情報が示されます。
- コンポーネント統合サービスが使用可能であれば、sp_help は、リモート・オブジェクトの記憶ロケーションに関する情報を表示します。

- `sp_help` は、暗号化キーについての情報を表示します。キー名が `sp_help` に対するパラメータとして指定されている場合、コマンドの実行結果には、キーの名前、所有者、オブジェクトの種類、および作成日がリストされます。
- `sp_help tablename` は、カラムが暗号化されているかどうかと、カラムの復号化デフォルト名 (存在する場合) を示します。
- `sp_help predicate_name` は、述部付きの権限についての情報を表示します。

パーミッション

`sp_help` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル SQLJ ルーチンの詳細については、『Adaptive Server Enterprise における Java』を参照してください。

コマンド [alter table](#)、[create table](#)

システム・プロシージャ [sp_chgattribute](#)、[sp_droprowlockpromote](#)、[sp_helppartition](#)、[sp_helpcomputedcolumn](#)、[sp_helpconstraint](#)、[sp_helpindex](#)、[sp_setpglockpromote](#)

sp_help_resource_limit

説明 リソース制限についてレポートします。

構文 `sp_help_resource_limit [name [, appname [, limittime
[, limitday [, scope [, action[, verbose]]]]]]]`

パラメータ *name*
制限が適用される Adaptive Server ログインです。特定のログインに適用されている制限の情報については、ログイン *name* を指定します。ログインに関係のない制限の情報については、`null` を指定します。

注意 ユーザがシステム管理者でない場合は、自分のログインを指定するか、または `NULL` のログインを指定すると、適用されているリソース制限についての情報が表示されます。

appname
制限が適用されるアプリケーションの名前です。特定のアプリケーションに適用されている制限についての情報を調べるには、ログイン・パケットで、クライアント・プログラムが Adaptive Server に渡すアプリケーション名を指定します。特定のアプリケーションに関係のない制限の情報については、`null` を指定します。

limittime
制限が実施される時間範囲です。任意の時刻に有効な制限についての情報を調べるには、次の形式を使用して、“00:00”～“23:59”の値を指定します。

`"HH:MM"`

時刻に関係のない制限の情報については、`null` を指定します。

limitday

制限が実施されている曜日です。特定の曜日に有効な制限の情報については、**master** データベースのシステム・テーブル **syslanguages** に格納されているデフォルト・サーバ言語で、曜日をフルスペルで指定します。実施される曜日に関係のない制限の情報については、**null** を指定します。

scope

制限が適用されるスコープです。次のいずれかを指定します。

スコープのコード	help でリストするすべての制限のスコープ
1	クエリ
2	クエリ・バッチ (クライアントがサーバに送信する 1 つ以上の SQL 文)
4	トランザクション
6	クエリ・バッチとトランザクションの両方
NULL	指定した <i>name</i> 、 <i>appname</i> 、 <i>limittime</i> 、 <i>limitday</i> 、 <i>action</i> 。各制限の <i>scope</i> は無関係。

action

制限値を超えたときに実行される動作です。次のいずれかを指定します。

動作コード	help でリストするすべての制限のアクション
1	警告を出す。
2	クエリ・バッチをアボートさせる。
3	トランザクションをアボートさせる。
4	セッションを強制終了させる。
NULL	指定した <i>name</i> 、 <i>appname</i> 、 <i>limittime</i> 、 <i>limitday</i> 、 <i>scope</i> を制御する。各制限の <i>action</i> とは無関係。

verbose

値 1 または 0 (ゼロ) で、出力が冗長モードで表示されます。

例

例 1 システム・テーブル **sysresourcelimits** に保管された、すべてのリソース制限をリストします。

```
sp_help_resource_limit
```

例 2 ユーザ “joe_user” に適用されるすべての制限をリストします。

```
sp_help_resource_limit joe_user
```

例 3 アプリケーション **my_app** に適用されるすべての制限をリストします。

```
sp_help_resource_limit NULL, my_app
```

例 4 9:00 a.m. に実施されるすべての制限をリストします。

```
sp_help_resource_limit NULL, NULL, "09:00"
```

例 5 9:00 a.m. に実施されるすべての制限をリストする、もう 1 つの方法です。

```
sp_help_resource_limit @limitttype = "09:00"
```

例 6 月曜日に実施されるすべての制限をリストします。

```
sp_help_resource_limit NULL, NULL, NULL, Monday
```

例 7 月曜日の 9:00 a.m. に、“joe_user” に対して実施されるすべての制限をリストします。

```
sp_help_resource_limit joe_user, NULL, "09:00", Monday
```

例 8 すべての制限を冗長モードでリストします。

```
sp_help_resource_limit null,null,null,null,null,null,1
```

例 9 すべてのリソースの制限を冗長モードでリストします。

```
sp_help_resource_limit @verbose=1
```

使用法

- `sp_help_resource_limit` は、すべてのリソースの制限、任意のログインまたはアプリケーションに適用されている制限、任意の時刻または曜日に適用されている制限、あるいは任意のスコープまたはアクションが設定された制限についてレポートします。

パーミッション

`sp_help_resource_limit` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage resource limit</code> 権限を持つユーザであることが必要。どのユーザも <code>sp_help_resource_limit</code> を実行して各自のリソース制限を表示できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。どのユーザも <code>sp_help_resource_limit</code> を実行して各自のリソース制限を表示できる。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 **マニュアル** リソース制限の詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_add_resource_limit](#)、[sp_drop_resource_limit](#)、[sp_modify_resource_limit](#)

sp_help_qpgroup

説明 抽象プラン・グループの情報をレポートします。

構文 sp_help_qpgroup [*group* [, *mode*]]

パラメータ *group*
抽象プラン・グループの名前です。

mode
出力するレポートのタイプです。次のうちのいずれかです。

モード	返される情報
full	グループ内のローの数とプランの数、複数のローを使用するプランの数、最も長いプランのローとプラン ID の数、ハッシュ・キーとハッシュ・キーの衝突情報。これはデフォルトのレポート・モードです。
stats	ハッシュ・キー情報以外の “full” レポートからの情報すべて。
hash	グループ内のローの数と抽象プランの数、ハッシュ・キーの数、ハッシュ・キーの衝突情報。
list	グループ内のローの数と抽象プランの数、および各クエリ／プランのペアの情報 (ハッシュ・キー、プラン ID、クエリの最初の数文字、プランの最初の数文字)。
queries	グループ内のローの数と抽象プランの数、および各クエリの情報 (ハッシュ・キー、プラン ID、クエリの最初の数文字)。
plans	グループ内のローの数と抽象プランの数、および各プランの情報 (ハッシュ・キー、プラン ID、プランの最初の数文字)。
counts	グループ内のローの数と抽象プランの数、および各プランの情報 (ローの数、文字数、ハッシュ・キー、プラン ID、クエリの最初の文字数)。

例 **例 1** データベース内にあるすべての抽象プラン・グループの概要情報をレポートします。

```
sp_help_qpgroup
```

```

Group                GID          Plans
-----
ap_stdin              1              0
ap_stdout             2              0
dev_test              3             209

```

例 2 test_plans グループについてレポートします。

```
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.

- 使用法**
- 抽象プランのグループ名で使用され、`mode` パラメータが指定されていない場合、`sp_help_qpgroup` のデフォルト・モードは、`full` になります。
 - ハッシュ・キーの衝突は、特定のユーザが使用する複数のプランが同じハッシュ・キー値を持っていることを示しています。ハッシュ・キーの衝突があるとき、一致するハッシュ・キーを持つ各クエリのクエリ・テキストは、一致するクエリを特定するために、ユーザのクエリ・テキストと比較されなければなりません。そのため、わずかにパフォーマンスが低下します。

パーミッション `sp_help_qpgroup` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。自分の抽象プラン・グループに対しては、すべてのユーザが <code>sp_help_qpgroup</code> を実行できる。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_help_qplan](#)

sp_help_qplan

説明 抽象プランについての情報をレポートします。

構文 `sp_help_qplan id[, mode]`

パラメータ *id*
抽象プランの ID です。

mode
出力するレポートのタイプです。次のうちのいずれかです。

モード	返される情報
full	プラン ID、グループ ID、ハッシュ・キー、full クエリとプランのテキスト。
brief	full と同じである。ただし、クエリとプランの全体ではなく、クエリとプランの約 80 文字しか出力しない。これがデフォルト。
list	ハッシュ・キー、ID、およびクエリとプランの最初の 20 文字。

例 **例 1** 簡単な抽象プラン・レポートを作成します。

```
sp_help_qplan 800005881
gid          hashkey      id
-----
          5   2054169974   937054374

query
-----
select type, avg(price) from titles group by type

query_plan
-----
( plan
  ( store Worktab1
    ( i_scan type_price titles )
  )
  ( t_scan ( ...
```

例 2 抽象プラン・レポートの全体を表示します。

```
sp_help_qplan 784005824, full
```

使用法

- mode のパラメータ値を指定しなければ、デフォルトは brief になります。

パーミッション	<code>sp_help_qplan</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザであることが必要。自分の抽象プランに対しては、すべてのユーザが <code>sp_help_qplan</code> を実行できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。自分の抽象プランに対しては、すべてのユーザが <code>sp_help_qplan</code> を実行できる。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_find_qplan](#)、[sp_help_qpgroup](#)

sp_helpappttrace

説明 Adaptive Server がトレースしているセッションを調べます。
sp_helpappttrace は、Adaptive Server がトレースしているすべてのセッションのサーバ・プロセス ID (spid)、spid をトレースしているセッションの spid、およびトレース・ファイルの名前を返します。

構文 sp_helpappttrace

例

```
sp_helpappttrace

   traced_spid  tracer_spid      trace_file
   -----
11              exited         /tmp/myfile1
13              14              /tpcc/sybase.15_0/myfile2
```

使用法 sp_helpappttrace は次のカラムを返します。

- **traced_spid** — ユーザがトレースしているセッションの spid。
- **tracer_spid** — traced_spid がトレースしているセッションの spid。
tracer_spid セッションが終了している場合は、“exited” と出力されます。
- **trace_file** — トレース・ファイルのフル・パス。

トレースの再バインド

あるセッションが別のセッションをトレースしているときに、そのトレースを無効にしないでセッションが終了した場合は、新しいセッションを以前のトレースと再バインドできます。つまり、sa または sso は開始したトレースをすべて完了する必要がなく、1つのトレース・セッションを開始して終了した後で、そのトレース・セッションに再バインドできます。

パーミッション sp_helpappttrace のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage server 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role または sso_role が付与されたユーザであることが必要。

sp_helppartition

- 説明** テーブルまたはインデックスのパーティション関連の情報をリストします。
- 構文** `sp_helppartition [tablename [, { null | indexname | 'all' }], partitionname]]`
- パラメータ**
- tablename***
現在のデータベース内にあるテーブルの名前です。
- null**
ベース・テーブル・パーティションに関する情報をリストするように指定します。
- indexname***
現在のテーブルのインデックスの名前です。このインデックスに関する情報を表示します。
- 'all'**
すべてのインデックス・パーティション情報をリストするよう指定します。
- partitionname***
ベース・テーブルまたはインデックスのパーティションの名前です。
- 例** **例 1** `titles` テーブルのデータ・パーティションについて、概要と詳細情報を返します。

```
sp_helppartition titles
go
```

```
name          type          partition_type  partitions  partition_keys
-----
titles        base table range                5  total_sales
```

(1 row affected)

```
partition_name  partition_id  pages  row_count  segment  create_date
-----
smallsales      1440005130    1      5  titleseg1  Sep 26 2005 5:44PM
smallsales2     1456005187    1      0  titleseg2  Sep 26 2005 5:44PM
smallsales3     1472005244    1      2  titleseg3  Sep 26 2005 5:44PM
mediumsales4    1488005301    1      8  titleseg4  Sep 26 2005 5:44PM
bigsales5       1504005358    1      3  titleseg5  Sep 26 2005 5:44PM
```

Partition_Conditions

```
-----
VALUES <= (1000)
VALUES <= (2000)
```

```
VALUES <= (3000)
VALUES <= (10000)
VALUES <= (25000)
```

Avg_pages	Max_pages	Min_pages	Ratio(Max/Avg)	Ratio(Min/Avg)
1	1	1	1.000000	1.000000

(return status = 0)

例 2 titles テーブルについての概要パーティション情報と、smallsales データ・パーティションについての詳細情報を返します。

```
sp_helppartition titles, null, smallsales
```

name	type	partition_type	partitions	partition_keys
titles	base table	range	5	total_sales

(1 row affected)

partition_name	partition_id	pages	row_count	segment	create_date
smallsales	1440005130	1	5	titleseg1	Sep 26 2005 5:44PM

Partition_Conditions

```
VALUES <= (1000)
(return status = 0)
```

例 3 最初に my_titles テーブルにノンクラスタード・インデックス ncidx_local を作成し、次に my_titles についての概要パーティション情報と、ncidx_local のパーティション ncip4 についての詳細情報を返します。

```
create nonclustered index ncidx_local on my_titles(title_id) local index
(ncip1, ncip2, ncip3, ncip4, ncip5)
```

```
go
sp_helppartition my_titles, ncidx_local, ncip4
go
```

name	type	partition_type	partitions	partition_keys
ncidx_local	local index	range	5	total_sales

(1 row affected)

partition_name	partition_id	pages	row_count	segment	create_date
ncip4	1584005643	1	8	default	Sep 26 2005 6:06PM


```
Partition_Conditions
```

```
-----
VALUES <= (10000)
(return status = 0)
```

使用法

- `sp_helppartition` は、テーブル・レベル、インデックス・レベル、パーティション・レベルで、パーティション関連の情報をリストします。テーブル・レベルまたはインデックス・レベルのパーティション情報には、インデックスのタイプ(ローカル・インデックスかグローバル・インデックスか)、パーティションのタイプ、パーティションの数、パーティション・キー(該当する場合)が含まれます。各パーティションについての情報に含まれるのは、パーティション名、ID、ページ数、セグメント名、作成日、分割条件(該当する場合)です。

概要情報は、パーティション 1 つあたりのページ数、ページの最小数と最大数、各ページの平均ページ数と最大または最小ページ数の間の比率を表示します。

- テーブル名を指定しない場合、`sp_helppartition` では、現在のデータベース内のすべてのユーザ・テーブルに関して、所有者、テーブル名、パーティション数、パーティション・タイプがリストされます。
- `sp_helppartition` で、インデックス名または `null` の代わりに 'all' を指定した場合、指定したテーブルおよびベース・テーブルの各インデックスについて、テーブル・レベルおよびインデックス・レベルのパーティション情報がリストされます。
- `sp_helppartition` で、特定のインデックスを指定した場合、そのインデックスについてのインデックス・レベルの情報がリストされます。
 - `sp_helppartition` で、パーティション名を指定しなかった場合、インデックスのすべてのパーティションについてのパーティション・レベルの情報と、パーティションの概要情報が表示されます。
 - `sp_helppartition` で、パーティション名を指定した場合、そのパーティションについてのパーティション・レベルの情報のみが表示されます。
- `sp_helppartition` で、テーブル名のみを指定した場合、ベース・テーブルについてのテーブル・レベルのインデックス・パーティション情報と、ベース・テーブルのすべてのパーティションについてのパーティション・レベルの情報が表示されます。

- `sp_helppartition` で、インデックス名の代わりに `null` を指定し、パーティション名を指定した場合、ベース・テーブルについてのテーブル・レベルのパーティション情報と、指定したパーティションについてのパーティション・レベルの情報が表示されます。概要情報は表示されません。
- パーティションは、`create table`、`alter table`、および `select into` を使用して作成されます。分割の詳細については、これらのコマンドを参照してください。
- パーティションが格納されているセグメント上の使用済みページ数と空きページ数を表示するには、`sp_helpsegment` を使用します。

結果の正確さ

- “pages” カラムにレポートされた値は、実際の値と異なる可能性があります。カウントが正確かどうかを見極めるには、`sp_statistics` と `sp_helppartition` を実行してデータ・ページ・カウントを比較します。`sp_statistics` が報告するカウントは常に正確です。

`sp_statistics` がレポートするページ・カウントと、`sp_helppartition` がレポートするパーティション・ページの合計数との誤差が 5% を超えた場合は、次のコマンドのいずれかを実行して、パーティションの統計値を更新します。

- `dbcc checkdb`
- `dbcc checktable`
- `update all statistics`
- `update table statistics`

更新後、`sp_helppartition` を再実行して、正確なレポートを取得します。

パーミッション

`sp_helppartition` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 [カタログ・システム・プロシージャ sp_statistics](#)
 コマンド [alter table](#)、[create table](#)、[select into](#)
 システム・プロシージャ [sp_helpsegment](#)

sp_helpcache

説明 データ・キャッシュにバインドされたオブジェクトについての情報、または指定されたキャッシュ・サイズに必要なオーバヘッド量を表示します。

構文 `sp_helpcache {cache_name | "cache_size[P | K | M | G]",
'instance instance_name'}`

パラメータ

cache_name

既存のデータ・キャッシュの名前です。

cache_size

キャッシュのサイズを指定します。サイズの指定に使用する単位は、P (ページ)、K (キロバイト)、M (メガバイト)、G (ギガバイト) です。デフォルトは K です。

instance_name

調べるキャッシュのインスタンス名です。

例 **例 1** `pub_cache` にバインドされた項目についての情報を表示します。

```
sp_helpcache pub_cache
```

例 2 80MB のデータ・キャッシュを作成するために必要なオーバヘッド量を示します。

```
sp_helpcache "80M"
```

例 3 すべてのキャッシュとそれらにバインドされたすべての項目についての情報を表示します。

```
sp_helpcache
```

例 4 クラスタ環境の場合 - サイズが 10M のインスタンス “blade1” に対してキャッシュ C2 のオーバヘッドを表示します。

```
sp_helpcache 'C2', '10M', 'instance blade1'
```

使用法

- サーバ上にあるすべてのデータ・キャッシュのサイズ、ステータス、I/O サイズを調べるには、[sp_cacheconfig](#) を使用します。
- [sp_cacheconfig](#) によってデータ・キャッシュを設定すると、指定するすべてのメモリが、そのデータ・キャッシュ内で使用できます。キャッシュを管理するためのオーバヘッド量は、デフォルト・データ・キャッシュのサイズから引かれます。[sp_helpcache](#) は、指定されたサイズのキャッシュに必要なメモリ量を表示します。
- クラスタ環境の場合 - *instance_name* を指定しない場合、[sp_helpcache](#) はすべてのキャッシュの情報を表示します。

- オブジェクトをキャッシュにバインドするには、[sp_bindcache](#) を使用します。特定のオブジェクトをキャッシュからバインド解除するには、[sp_unbindcache](#) を使用します。特定のキャッシュにバインドされているすべてのオブジェクトをバインド解除するには、[sp_unbindcache_all](#) を使用します。
- プロシージャ [sp_cacheconfig](#) は、データ・キャッシュを設定します。プロシージャ [sp_poolconfig](#) は、データ・キャッシュ内のメモリ・プールを設定します。
- [sp_helpcache](#) は、74GB までならオーバーヘッドを正確に計算できます。
- システム [tempdb](#) 上で [sp_bindcache](#) を引き続き実行できますが、システム [tempdb](#) のバインドは、動的ではありません。サーバを再起動すると変更内容が反映されます。システム [tempdb](#) をデフォルトのデータ・キャッシュに明示的にバインドしていないかぎり、[sp_helpcache](#) は保留中を示すステータス “P” をレポートします。明示的にバインドしている場合は、有効のステータス “V” をレポートします。これは、デフォルトでシステム [tempdb](#) はすでにデフォルトのデータ・キャッシュにバインドされているからです。

パーミッション

[sp_helpcache](#) は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

[sysaudits](#) テーブルの [event](#) カラムと [extrainfo](#) カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_bindcache](#)、[sp_cacheconfig](#)、[sp_poolconfig](#)、[sp_unbindcache](#)、[sp_unbindcache_all](#)

sp_helpcomputedcolumn

説明 指定したテーブルの計算カラムに関する情報をレポートします。

構文 `sp_helpcomputedcolumn {tablename}`

パラメータ *tablename*
計算カラムを含むテーブルの名前です。

例 この例では、`mytitles` テーブルの計算カラムについてレポートします。

```
sp_helpcomputedcolumn mytitles
Column_Name Property
-----
sum_sales      materialized

Text
-----
AS price * total_sales materialized

(return status = 0)
```

使用法

パーミッション `sp_helpcomputedcolumn` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

sp_helpconfig

- 説明** 設定パラメータに関するヘルプ情報をレポートします。
- 構文** `sp_helpconfig "configname" [, "size"]`
- パラメータ**
- configname**
クエリ (照会) される設定パラメータ、またはユニークなパラメータ以外のパラメータのフラグメントです。
- size**
メモリのサイズです。サイズの指定に使用する単位は、**B** (バイト)、**K** (キロバイト)、**M** (メガバイト)、**G** (ギガバイト)、**P** (ページ) です。**size** は、サイズの種類の指定なしに使用されると、このパラメータによって設定されているエンティティ (たとえば、ロック、オープン・インデックスなど) の数を指定します。**configname** がユニークなパラメータ名でない場合、**size** は無視されます。
- 例** **例 1** “allow” で始まる設定オプションすべてに関して、レポートを返します。

```
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
```

例 2 オブジェクト記述子 421 用のメタデータ・キャッシュを作成するために必要なメモリ量に関するレポートを返します。

```
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.
```

例 3 1MB のデータベース・キャッシュを満たすデータベース記述子の数に関するレポートを返します。

```
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.

例 4 512K のメモリを使用するロックの数に関するレポートを返します。

```
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.

例 5 allow updates to system tables 設定パラメータのステータスに関するレポートを返します。

```
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

使用法

- `sp_helpconfig` は、設定パラメータに関するヘルプ情報、たとえば、パラメータがある値に設定された場合に必要メモリ・サイズなどをレポートします。`sp_helpconfig` は、現在の設定、その設定に使用されるメモリ量、デフォルト値、最小設定値および最大設定値も表示します。

注意 「最大値」の設定は、実際に設定可能な値ではなく、パラメータのデータ型が受け入れることができる最大数のことを指します。

多くの場合、設定パラメータの最大許容値は非常に高い値になっています。サーバで設定できる最大値は、通常、設定パラメータによる制限ではなく、利用可能なメモリとその他のリソースによって制限されます。

- `cluster options` は、厳密にクラスタ全体の設定オプションをすべて表示します。
- `system_view` を `cluster` に設定すると、`sp_helpconfig` は、クラスタ内のすべてのインスタンスの設定情報を表示します。
- `system_view` を `instance` に設定すると、`sp_helpconfig` は、現在のインスタンスの設定情報を表示します。
- ユニークなパラメータ以外のパラメータのセグメントを `configname` に使用すると、`sp_helpconfig` は、設定された値と現在の値が一致しているパラメータのリストを返します。例 1 を参照してください。
- `sp_helpconfig` は、静的オプション、動的オプション、および読み込み専用オプションを受け入れます。
- `sp_helpconfig 'restricted decrypt permission'` は次の表示を返します。

```
sp_helpconfig 'restricted decrypt permission'

0 - restricted decrypt permission disabled (default).
1 - restricted decrypt permission enabled

Minimum Value Maximum Value Default Value Current Value
Memory Used   Unit      Type
-----
0              0 switch   1         0         0
              0 dynamic
```

メタデータ・キャッシュの設定の計画

- サーバ用のメタデータ・キャッシュ設定の計画を立てるときは、`sp_helpconfig` を使用します。

たとえば、2,000 のユーザ・インデックスを格納しているデータベースを、別のサーバに移す計画を立てているとします。データベースのユーザ・インデックスを収容できるように、別のサーバ用に設定する必要があるメモリ量を求めるには、次のコマンドを入力します。

```
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.

もう 1 つの例として、インデックス・キャッシュに使用できる 1MB のメモリがあり、そのメモリをサポートできるインデックス記述子の数を知る必要がある場合は、次のコマンドを入力します。

```
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.

この出力に基づいて判断すると、1MB のメモリがある場合、最大 2461 個のインデックス記述子を格納できるインデックス記述子キャッシュを作成できます。このキャッシュを作成するには、`number of open indexes` 設定パラメータを次のように設定します。

```
sp_configure "number of open indexes", 2461
```

sybdiagdb での *sp_helpconfig* の使用 (Sybase サポート・センタ専用)

注意 Sybase 製品の保守契約を結んでいるサポート・センタは、デバッグ用に、使用されているシステムに *sybdiagdb* データベースを作成することができます。診断用の設定データが保管されるこのデータベースは、Sybase 製品の保守契約を結んでいるサポート・センタ専用のデータベースです。

sybdiagdb データベースで使用するために、*sp_helpconfig* には、次のような *configname* の各オプションが追加されています。これらの各オプションは、Sybase 製品の保守契約を結んでいるサポート・センタが使用します。

- *number of ccbs* — デバッグの支援に使用できる、設定可能なアクション・ポイント制御ブロックの数です。
- *caps per ccb* — 1つの設定可能なアクション・ポイント内でどの時点でも設定できる、設定可能アクション・ポイントの最大数です。
- *average cap size* — 通常の設定可能アクション・ポイントに対応する情報を格納するために必要なメモリのバイト数の見積もりです。

次に例を示します。

```
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

パーミッション

sybase_ts_role を必要とする次のオプションを除き、sp_helpconfig はすべてのユーザが実行できます。

- number of ccbs
- caps per ccb
- average cap size

パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

「[sybdiagdb での sp_helpconfig の使用 \(Sybase サポート・センタ専用\)](#)」(441 ページ) で指定されているオプションは、Sybase 製品の保守契約を結んでいるサポート・センタのみが使用できます。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_configure](#)、[sp_countmetadata](#)、[sp_monitorconfig](#)

sp_helpconstraint

説明 指定したテーブルで使用される整合性制約に関する情報をレポートします。

構文 `sp_helpconstraint [objname] [, detail]`

パラメータ *objname*

`create table` 文または `alter table` 文によって定義された、1 つまたは複数の整合性制約を課せられたテーブルの名前です。

detail

制約のユーザ・メッセージまたはエラー・メッセージに関する情報を返します。

例 **例 1** pubs3 データベース内の `store_employees` テーブルに関する制約情報を表示します。`store_employees` テーブルには、`stores` テーブルに対する外部キー (`stor_id`) が割り当てられています。また、(`mgr_id` が `emp_id` を参照する) 自己参照が割り当てられています。

```
sp_helpconstraint store_employees

name                defn
-----
store_empl_stor_i_27200400  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
```

例 2 制約のタイプやエラー・メッセージなど、pubs3..salesdetail 制約に関する詳細な情報を表示します。

```

sp_helpconstraint titles, detail

name                                type
   defn
   msg
-----
-----
datedflt                            default value
   create default datedflt as getdate()

typedflt                            default value
   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.

例 3 pubs3 データベース内にあるすべてのテーブルのリストを表示します。

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)

使用法

- `sp_helpconstraint` では、外部キーとリファレンス・キーが 30 文字にトランケートされます。
- `sp_helpconstraint` は、整合性制約の名前と定義、およびテーブルが使用している参照の数を出力します。`detail` オプションは、制約のユーザ・メッセージまたはエラー・メッセージについての情報を返します。
- `sp_helpconstraint` は、通常のインライン・デフォルトを表示する方法と同様に、共有可能なインライン・デフォルトを表示します。
- パラメータなしで `sp_helpconstraint` を実行すると、現在のデータベース内の、参照があるすべてのテーブルがリストされ、それぞれのテーブルでの参照の総数が表示されます。`sp_helpconstraint` は、それぞれのテーブルの参照の数をもとに、降順でテーブルをリストします。

- `sp_helpconstraint` は、(`create table` 文または `alter table` 文によって定義された) テーブルに関する整合性制約の情報だけをレポートします。`create index` 文を使用して作成されたインデックス、ルール、トリガに関する情報はレポートされません。テーブルのルール、トリガ、インデックスに関する情報を参照するには、`sp_help` を使用してください。
- ユーザ定義メッセージがない制約の場合、Adaptive Server は制約に対応したシステム・エラー・メッセージをレポートします。`sysmessages` を問い合わせると、そのエラー・メッセージの実際のテキストを表示できます。
- `sp_helpconstraint` は、現在のデータベース内にあるテーブルに対してのみ使用できます。
- 設定された補助スキャン記述子の数をクエリが超えると、Adaptive Server はエラー・メッセージを返します。必要なスキャン記述子の数を決定するために、`sp_helpconstraint` を使用できます。`number of aux scan descriptors` 設定パラメータの詳細については、『システム管理ガイド』を参照してください。
- システム・セキュリティ担当者は、`sp_helpconstraint` を実行するほとんどのユーザに対し、制約定義のソース・テキストを非表示にすることができます。`syscomments` テーブルの `text` カラムに対する `select` パーミッションをオブジェクト所有者またはシステム管理者だけに制限するには、`sp_configure` を使用して `select on syscomments.text column` パラメータを 0 に設定します。この制限は、評価済み設定で Adaptive Server を実行する場合に必要となります。評価済み設定の詳細については、『システム管理ガイド』を参照してください。

パーミッション

`sp_helpconstraint` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [alter table](#)、[create table](#)
 システム・プロシージャ [sp_configure](#)、[sp_help](#)、[sp_helpdb](#)、[sp_monitorconfig](#)

sp_helpdb

説明 特定のデータベースまたはすべてのデータベースについての情報をレポートします。

構文 sp_helpdb [dbname [, order]]

パラメータ *dbname*
その情報がレポートされるデータベースの名前です。このパラメータは省略できますが、省略すると、sp_helpdb はすべてのデータベースに関する情報をレポートします。*dbname* では、ワイルドカード文字を使用して、指定したパターンと一致するすべてのデータベースを返すことができます。

order
デフォルトでは、出力は *lstart* の順序になります。これは、データベースが作成または変更された順序です。*dbname* とともに *device_name* を使用すると、sp_helpdb の出力が *device_name* の順序で表示されます。

例 **例 1** Adaptive Server のすべてのデータベースに関する情報を表示します。

```
sp_helpdb
```

name	db_size	owner	bid	created	status
master	24.0 MB	sa	1	Jan 07, 2004	mixed log and data
model	8.0 MB	sa	3	Jan 07, 2004	mixed log and data
pubs2	8.0 MB	sa	4	Jan 21, 2004	trunc log on chkpt, mixed log and data
sybssystemdb	8.0 MB	sa	31513	Jan 07, 2004	mixed log and data
sybssystemprocs	112.0 MB	sa	31514	Jan 07, 2004	trunc log on chkpt, mixed log and data
tempdb	8.0 MB	sa	2	Feb 24, 2004	select into/bulkcopy/ pllsort, trunc log on chkpt, mixed log and data

```
(1 row affected)
```

```
(return status = 0)
```

例 2 (pubs2 内部から発行された場合) pubs2 データベースに関する情報と、セグメント情報を表示します。

```
1> use pubs2
2> go
1> sp_helpdb pubs2
2> go
```

```
name db_size owner dbid created status
```

```

-----
pubs2  20.0 MB  sa      4      Apr 13, 2005  trunc log on chkpt, mixed log
                                           and data

(1 row affected)
pubs2
device_fragments  size  usage          created          free kbytes
-----
master            10.0 MB  data and log  Apr 13 2005 10:29AM  2304
pubs_2_dev       10.0 MB  data and log  Apr 13 2005 10:33AM  9888

device           segment
-----
master           default
master           logsegment
master           system
pubs_2_dev       default
pubs_2_dev       logsegment
pubs_2_dev       system
pubs_2_dev       titleseg1
pubs_2_dev       titleseg2
pubs_2_dev       titleseg3
pubs_2_dev       titleseg4
pubs_2_dev       titleseg5
return status = 0)

```

例 3 (pubs2 内部から発行されていない場合) pubs2 データベースに関する情報を表示します。

```

sp_helppdb pubs2

name  db_size  owner  dbid    created      status
-----
pubs2  20.0 MB  sa     4      Jan 21, 2004  trunc log on chkpt, single
user,
                                           mixed log and data

(1 row affected)

device_fragments  size  usage          created          free kbytes
-----
master            10.0 MB  data and log  Apr 13 2005 10:29AM  2304
pubs_2_dev       10.0 MB  data and log  Apr 13 2005 10:33AM  9888
(return status = 0)

```

例 4 order パラメータに device_name を指定して sp_helppdb のデフォルトのソート順を上書きし、mydb のデバイス・フラグメントをアルファベット順に表示します。

```

sp_helppdb mydb, device_name

```

```

name                db_size  owner  dbid  created        status
-----
mydb                4.5 MB   sa     5     Feb 27, 2003  no options set

(1 row affected)
device_fragments  size      usage      created        free kbytes
-----
A                 1.5 MB   data only  Feb 27 2003   7:50AM   1530
B                 1.0 MB   log only   Feb 27 2003   7:50AM   not applicable
C                 2.0 MB   data only  Feb 27 2003   7:50AM   846

```

例 5 pubtune データベース用に設定されているロー・ロック・プロモーションの属性を表示します。

```

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

```

例 6 データベースが、**status** カラム下にあるユーザ作成のテンポラリ・データベースかどうかを表示します。

```

sp_helpdb "mytempdb3"

name      db_size  owner  dbid  created        status
-----
mytempdb 32.0 MB  sa     7     Dec 2, 2001  select into/bulkcopy/pllsort, trunc
log on chkpt, user created temp db

```

使用方法

- **sp_helpdb** は、**dbname** が特定されている場合は、指定されたデータベースに関する情報をレポートします。**dbname** に値が提供されていない場合、**sp_helpdb** は、**master.dbo.sysdatabases** にリストされているすべてのデータベースに関する情報をレポートします。
- **sp_helpdb** は、データベースがオフライン、圧縮タイプ、ラージ・オブジェクトの圧縮レベル、ロー内ラージ・オブジェクトの長さ、ロー・ロック・プロモーション・スレッショルド(データベースに定義されている場合)かどうかなど、すべてのデータベース固有のプロパティおよび設定をレポートします。
- データベースの非同期ログ・サービスを有効にすると、**sp_helpdb** の出力の **attribute** カラムに “**async log srv**” と表示されます。

非同期ログ・サービスの詳細については、[sp_dboption](#)、および『パフォーマンス&チューニング・シリーズ：オブティマイザと抽象プラン』の「第3章 高度な最適化ツール」を参照してください。

- 専用のログ・データベースでのログ・セグメントのディスク区分の場合、[sp_helpdb](#) は、ディスク区分単位のレポートでの空き領域フィールドに “not applicable” と発行します。また、[sp_helpdb](#) は、[free pages](#) というカラムも含めます。これは、ログ・セグメントにある空きページの数を示す値です。
- (クラスタ環境のみ) 指定したデータベースがリモート・インスタンスによって所有されているローカル・テンポラリ・データベースの場合、[sp_helpdb](#) はデバイス関連の情報を表示しません。
- [dbname](#) 内でワイルドカード文字を使用すると、指定したパターンに一致するすべてのデータベースを返すことができます。ワイルドカード文字使用の詳細については、『リファレンス・マニュアル：ビルディング・ブロック』の「第4章 式、識別子、およびワイルドカード文字」を参照してください。
- [sp_helpdb dbname](#) を [dbname](#) から実行すると、レポート内に空き領域とセグメントの情報が追加されます。
- [sp_helpdb](#) は、データベースの属性についての情報を表示します。属性が定義されていると、属性のクラス、名前、整数値、文字値、コメントを表示します。例3は、[pubs2](#) データベースに対するキャッシュ・バインド属性を示しています。
- [for load](#) オプションによって作成されたデータベースには、[sp_helpdb](#) からの出力に “don't recover” (リカバリ禁止) のステータスが表示されます。
- コンポーネント統合サービスが使用可能な場合、[sp_helpdb](#) は、指定されたデータベースまたはすべてのデータベースのデフォルトの記憶ロケーションをリストします。デフォルトの記憶ロケーションがない場合は、“NULL” を表示します。

パーミッション

[sp_helpdb](#) は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの 実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 [マニュアル『パフォーマンス&チューニング・ガイド：オプティマイザと抽象プラン』の「第3章 高度な最適化ツール」](#)、[『リファレンス・マニュアル：ビルディング・ブロック』の「第4章 式、識別子、およびワイルドカード文字」](#)

コマンド [alter database](#)、[create database](#)

システム・プロシージャ [sp_configure](#)、[sp_dboption](#)、[sp_rename](#)

sp_helpdevice

説明 特定のデバイスまたはすべての Adaptive Server のデータベース・デバイスとダンプ・デバイスについての情報をレポートします。

構文 sp_helpdevice [devname]

パラメータ *devname*
その情報がレポートされるデバイスの名前です。このパラメータを省略すると、sp_helpdevice はすべてのデバイスについてレポートします。

例 **例 1** Adaptive Server 上のすべてのデバイスに関する情報を表示します。

```
1sp_helpdevice
2> go

device_name physical_name      description
            status cntrltype vdevno      vpn_low      vpn_high
-----
-----
dev1        d:¥sybdata¥RV150.dev1 special, dsync off, directio on, physical
disk, 150.00 MB, Free:0.00 MB
            2          0          2          0          76799

dev2        d:¥sybdata¥RV150.dev2 special, dsync on, directio off, physical
disk, 150.00 MB, Free:130.00 MB
            16386       0          3          0          76799

master      d:¥sybdata¥RV150.mas special, dsync on, directio off, default
disk, physical disk, 30.00 MB, Free:0.50 MB
            3          0          0          0          15359

sysprocsdev d:¥sybdata¥RV150.ssp special, dsync on, directio off, physical
disk, 120.00 MB, Free:0.00 MB
            16386       0          1          0          61439

tapedump1  ¥¥.¥TAPE0          disk, dump device
            16          2          0          0          20000

tapedump2  ¥¥.¥TAPE1          tape,          625 MB, dump device
            16          3          0          0          20000

(6 rows affected, return status = 0)
```

例 2 diskdump という名前のダンプ・デバイスに関する情報をレポートします。

```
sp_helpdevice diskdump
```

使用法

- `sp_helpdevice` は、割り付けられていない領域の量をデバイス単位で表示します。出力の `description` カラムにブレースホルダ `Free` で示されます。

注意 特にページ・サイズが大きいサーバでは、デバイス上の小さい領域が未使用のままになる場合があります。たとえば、16K のサーバで 250MB のデバイスのうち、残りの 2MB は割り付けできません。`sp_helpdevice` はこの 2MB を `Free` とレポートします。これは、16K のサーバにおけるアロケーション・ユニットのサイズが 4Mb であり、アロケーション・ユニットの倍数にあたる領域しか割り付けできないためです。

- `sp_helpdevice` は、`devname` が特定されている場合は、指定したデバイスについての情報を表示します。引数を指定しなかった場合は、`master.dbo.sysdevices` 内のすべてのデバイスに関する情報を表示します。
- `sysdevices` テーブルには、ダンプ・デバイスとデータベース・デバイスが格納されています。

データベース・デバイスを、デフォルト・デバイスとして指定することができます。つまり、データベース・デバイスは、データベースの記憶領域として使用できます。このデータベース・デバイスは、ユーザが `create database` または `alter database` を発行するときに、データベース・デバイス名を指定しなかったり、`default` キーワードを指定した場合に、デフォルトとして使用できます。データベース・デバイスをデフォルト・データベース・デバイスにするには、システム・プロシージャ `sp_diskdefault` を実行します。

- `disk init` コマンドによって、データベース・デバイスをシステムに追加します。ダンプ・デバイスは、`sp_addumpdevice` で追加します。
- 単一のデバイスに対して `sp_helpdevice` を発行すると、そのデバイスに割り付けられたフラグメントのリストが表示されます。
- `description` カラムには、デバイス・タイプに関する情報が表示されます。次に、デバイス・タイプを示します。
 - ブロック・デバイス
 - ファイル・システム・デバイス
 - ロー・デバイス

status カラム内の番号は、“description” カラム内のステータス記述に対応します。

cntrtype カラムは、デバイスのコントローラ番号を指定します。ディスクまたはファイル・ダンプ・デバイスの場合、cntrtype は 2 であり、テープ・ダンプ・デバイスの場合には 3 ~ 8 です。データベース・デバイスの場合、cntrtype は通常 0 です (インストール環境に特殊なタイプのディスク・コントローラがない場合)。

vdevno カラムは、ダンプ・デバイスは 0、マスタ・データベース・デバイスは 0、その他のデータベース・デバイスは 1 以上です。

vpn_low カラムと vpn_high カラムは、それぞれ Adaptive Server 内のすべてのデバイス間でユニークな仮想ページ番号を表します。

パーミッション

sp_helpdevice は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照

コマンド [disk init](#)、[dump database](#)、[dump transaction](#)、[load database](#)、[load transaction](#)

システム・プロシージャ [sp_addumpdevice](#)、[sp_deviceattr](#)、[sp_diskdefault](#)、[sp_dropdevice](#)、[sp_logdevice](#)

sp_helpextendedproc

説明 現在のデータベース内の ESP (拡張ストアド・プロシージャ) と、それに対応した DLL ファイルを表示します。

構文 sp_helpextendedproc [esp_name]

パラメータ *esp_name*
拡張ストアド・プロシージャの名前です。現在のデータベースのプロシージャでなければなりません。

例 **例 1** xp_cmdshell ESP とその関数が格納されている DLL ファイルの名前をリストします。

```
use sybsemprocs
go
sp_helpextendedproc xp_cmdshell

ESP Name      DLL Name
-----
xp_cmdshell   sybsyesp
```

例 2 現在のデータベース内のすべての ESP と、その関数が格納されている DLL ファイルの名前をリストします。

```
sp_helpextendedproc

ESP Name      DLL Name
-----
xp_freedl     sybsyesp
xp_cmdshell   sybsyesp
```

使用法

- *esp_name* を省略すると、sp_helpextendedproc は、データベース内のすべての拡張ストアド・プロシージャをリストします。
- *esp_name* では大文字と小文字を区別します。ESP の作成に使用された *esp_name* と同じ名前を、大文字と小文字を区別して指定してください。

パーミッション sp_helpextendedproc は、すべてのユーザが実行できます。

パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

コマンド [create procedure](#)、[drop procedure](#)

拡張システム・プロシージャ [xp_cmdshell](#)

システム・プロシージャ [sp_addextendedproc](#)、[sp_dropextendedproc](#)

sp_helpexternlogin

説明	(コンポーネント統合サービスのみ) 外部ログイン名についての情報をレポートします。
構文	sp_helpexternlogin [server[, loginame[, rolename]]]
パラメータ	<p><i>server</i></p> <p>sp_addserver によって、ローカル・サーバに追加されているリモート・サーバの名前です。</p> <p><i>loginame</i></p> <p>ローカル・サーバのログイン・アカウントです。</p> <p><i>rolename</i></p> <p>Adaptive Server のユーザに割り当てられた役割です。</p>
例	<p>例 1 すべてのリモート・サーバ、ローカル・ログイン名、役割名、外部ログインを表示します。</p> <pre>sp_helpexternlogin</pre> <p>例 2 SSB という名前のサーバのローカル・ログイン、役割名、外部ログインを表示します。</p> <pre>sp_helpexternlogin SSB</pre> <p>例 3 “milo” という名前のユーザの、リモート・サーバ、ローカル・ログイン名、外部ログインを表示します。</p> <pre>sp_helpexternlogin NULL, milo</pre> <p>例 4 ローカル・ユーザ名が “trixi” であるリモート・サーバ SSB の外部ログインを表示します。</p> <pre>sp_helpexternlogin SSB, trixi</pre> <p>例 5 sa_role を与えられたローカル・ユーザに、リモート・サーバ SSB の外部ログインを表示します。</p> <pre>sp_helpexternlogin SSB, NULL, sa_role</pre>
使用法	<ul style="list-style-type: none"> sp_helpexternlogin は、すべてのリモート・サーバ、ユーザのローカル・ログイン名、役割名、ユーザの外部ログイン名を表示します。 リモート・サーバを追加するには、sp_addserver を使用します。ローカル・ログインを追加するには、sp_addlogin を使用します。
パーミッション	sp_helpexternlogin は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addexternlogin](#)、[sp_addlogin](#)、[sp_addserver](#)、[sp_dropexternlogin](#)、[sp_helpserver](#)

sp_helpgroup

説明 特定のグループまたは現在のデータベース内のすべてのグループに関する情報をレポートします。

構文 sp_helpgroup [grpname]

パラメータ *grpname*
 sp_addgroup によって作成された、データベース内のグループの名前です。

例 1 現在のデータベース内のすべてのグループに関する情報を表示します。

```
sp_helpgroup

Group_name          Group_id
-----
hackers             16384
public              0
```

例 2 “hackers” グループに関する情報を表示します。

sp_helpgroup hackers

```
Group_name          Group_id          Users_in_group    Userid
-----
hackers             16384            ann                4
hackers             16384            judy               3
```

- 使用法**
- デフォルト・グループ “public” についてのレポートを取得するには、“public” という名前を一重引用符または二重引用符で囲みます (“public” は予約語です)。
 - 指定されたグループにメンバがない場合、sp_helpgroup は次のようにヘッダを表示しますが、ユーザはリストしません。

```
Group_name          Group_id          Users_in_group    Userid
-----
```

パーミッション sp_helpgroup は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [grant](#)、[revoke](#)

システム・プロシージャ [sp_addgroup](#)、[sp_changegroup](#)、[sp_dropgroup](#)、[sp_helprotect](#)、[sp_helpuser](#)

sp_helpindex

説明 テーブル上に作成されたインデックスについての情報をレポートします。計算カラム・インデックスと関数ベース・インデックスについての情報をレポートします。

構文 `sp_helpindex objname`

パラメータ *objname*
現在のデータベース内にあるテーブルの名前です。

例 **例 1** sysobjects テーブルのインデックスのタイプを表示します。

```
sp_helpindex sysobjects

index_name index_keys index_description index_max_rows_per_page
          index_fillfactor index_reservepagegap index_created
          index_local

sysobjects      id          clustered, unique          0
                0          0          Apr 12 2005  2:38PM
          Global Index
ncsysobjects    name, uid nonclustered, unique
                0          0          Apr 12 2005  2:38PM
          Global Index

(2 rows affected)
index_pt_name          index_ptn_seg
-----
sysobjects_1          system
ncsysobjects_1       system
```

例 2 pubs2 データベースの titles テーブルのインデックスについての情報を表示します。titles テーブルは分割されていますが、インデックス titleind は分割されていません。titleind は、ノンクラスタード (単一パーティション) のグローバル・インデックスです。

```
sp_helpindex titles

index_name index_keys index_description index_max_rows_per_page
          index_fillfactor index_reservepagegap index_created
          index_local

titleind      title          nonclustered          0
          Global Index

(1 row affected)
index_pt_name          index_ptn_seg
-----
titleind_1232004389  default
```


例 3 mysalesdetail テーブルについてのインデックス情報を表示します。mysalesdetail は、ord_num カラムのハッシュによって分割されています。ord_num に、3つのパーティションからなるローカルのクラスタード・インデックスも作成されています。

```
sp_helpindex mysalesdetail

index_name index_keys index_description index_max_rows_per_page
      index_fillfactor index_reservepagegap index_created      index_local
-----
-----
-----
clust_idx  ord_num      clustered                          0
          0                          0 Apr 12 2005 2:38PM  Local Index

(1 row affected)
index_pt_name      index_ptn_seg
-----
clust_idx_1344004788  default
clust_idx_1360004845  default
clust_idx_1376004902  default
```

例 4 関数ベースのインデックスを表示します。

```
create index sum_sales on mytitles (price * total_sales)
sp_helpindex mytitles

Object has the following indexes

index_name index_keys index_description index_max_rows_per_page
      index_fillfactor index_reservepagegap index_created      index_local
-----
-----
-----
sum_sales  sybfi2_1  nonclustered                          0
          0                          0 Oct 12 2005 3:34PM  Global Index

(1 row affected)
index_ptn_name      index_ptn_seg
-----
sum_sales_1724867646  default

(1 row affected)

Object has the following functional index keys

Internal_Index_Key_Name
-----
sybfi2_1

(1 row affected)

Expression
-----
price * total_sales

(return status = 0)
```

使用法

- `sp_helpindex` は、`create table` 文または `alter table` 文によって一意性制約またはプライマリ・キー制約を定義して作成されたインデックスを含め、テーブル上のすべてのインデックスをリストします。
- `sp_helpindex` は、テーブルのインデックスに割り当てられた属性 (たとえば、キャッシュ・バインド) を表示します。
- `sp_helpindex` は、次のものを表示します。
 - 各インデックスのパーティション情報。
 - インデックスがローカルかグローバルか、クラスタードかノックラスタードか。
 - インデックスの `max_rows_per_page` 設定値。
 - データオンリー・ロック・テーブル上のクラスタード・インデックスの情報。
データオンリー・ロック・テーブル内にあるクラスタード・インデックスのインデックス ID (`indid`) は、1 と等しくありません。
 - キーのカラム順は、昇順か、または降順かを示します。
 - 記憶領域管理プロパティの値。
 - 後ろに順番が示されるキー・カラム名。降順のみが表示されます。たとえば、カラム `a ASC`, `b DESC`, `c ASC` にインデックスがある場合、“`index_keys`” は “`a, b DESC, c`” と表示します。

パーミッション

`sp_helpindex` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [create index](#)、[drop index](#)、[update statistics](#)

システム・プロシージャ [sp_help](#)、[sp_helpkey](#)、[sp_helppartition](#)

sp_helpjava

説明	データベースにインストールされている Java クラスと関連する JAR についての情報を表示します。
構文	<code>sp_helpjava ["class"[, <i>java_class_name</i>[, "detail" "depends"]] "jar", <i>jar_name</i>[, "depends"]]</code>
パラメータ	<p>"class" "jar" クラスまたは JAR のどちらかの情報を表示するよう指定します。"class" と "jar" は両方ともキーワードであるため、引用符で囲む必要があります。</p> <p><i>java_class_name</i> クラスの情報を表示したいときに指定するクラスの名前です。システムのクラス、またはデータベースにインストールされているユーザ定義のクラスを指定してください。</p> <p>detail そのクラスに関する詳細情報を表示するように指定します。</p> <p>depends 指定のクラス、または JAR 内のクラスに従属しているすべてのデータベース・オブジェクトをリストします。このリストには、SQLJ の関数、SQLJ のストアド・プロシージャ、ビュー、Transact-SQL のストアド・プロシージャ、テーブルも表示されます。</p> <p><i>jar_name</i> JAR の情報を表示したいときに指定する JAR の名前です。JAR をデータベースにインストールするときは、<code>installjava</code> を使用してください。</p>
例	<p>例 1 データベースにインストールされているすべてのクラスと、関連する JAR ファイルを表示します。</p> <pre>sp_helpjava</pre> <p>例 2 すべてのクラス名を表示します。</p> <pre>sp_helpjava "class"</pre> <p>例 3 Address クラスに関する詳細情報を表示します。</p>

```
sp_helpjava "class", Address, detail
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
```

使用法 **depends** パラメータがクラス (1 つまたは複数) の依存関係をリストするのは、クラスが、**SQLJ** ルーチンに使用される **create** 文の **external name** 句の中でリストされているときです。または、クラスが、データベース内でカラムのデータ型として使用されているときです。

パーミッション **sp_helpjava** は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 マニュアル データベースにおける Java の詳細については、『Adaptive Server Enterprise における Java』を参照してください。

コマンド remove java

ユーティリティ extractjava、installjava

sp_helpjoins

説明 2つのテーブルまたはビューから、ジョイン候補になると思われるカラムをリストします。

構文 sp_helpjoins *lefttab*, *righttab*

パラメータ *lefttab*
最初のテーブルまたはビューです。

righttab
2番目のテーブルまたはビューです。パラメータの順序は重要ではありません。

例 **例 1** sales と salesdetail テーブル内の、ジョイン候補になると思われるカラムのリストを表示します。

```
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
```

例 2 sysobjects と syscolumns のシステム・テーブル内の、ジョイン候補になると思われるカラムのリストを表示します。

```
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
```

- 使用法
- `sp_helpjoins` によって表示されるカラムのペアは、2つのソースのどちらかから取り出されたものです。`sp_helpjoins` は、現在のデータベース内の `syskeys` テーブルを調べて、2つのテーブルに `sp_foreignkey` によって定義されている外部キーがあるかどうかを確認します。次に、2つのテーブルに `sp_commonkey` によって定義されている共通キーがないかどうかを確認します。外部キーも共通キーも検出されない場合、`sp_helpjoins` は同じユーザ定義データ型を持っているキーを調べます。ユーザ定義データ型も見つからない場合は、同じ名前とデータ型を持っているカラムを調べます。
 - `sp_helpjoins` はジョインを作成しません。

パーミッション `sp_helpjoins` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_commonkey](#)、[sp_foreignkey](#)、[sp_helpkey](#)、[sp_primarykey](#)

sp_helpkey

説明 特定のテーブルまたはビューのプライマリ・キー、外部キー、共通キーに関する情報をレポートします。または現在のデータベース内にあるすべてのキーに関する情報をレポートします。

構文 sp_helpkey [tabname]

パラメータ *tabname*
現在のデータベース内にあるテーブルまたはビューの名前です。名前を指定しなければ、このプロシージャは現在のデータベースに定義されているすべてのキーについてレポートします。

例 現在のデータベースに定義されているキーについての情報を表示します。“object_keys”カラムと“related_keys”カラムは、そのキーを構成しているカラムの名前を示します。

```
sp_helpkey
```

```
keytype object      related_object object_keys      related_keys
-----
primary authors    -- none --      au_id,*,*,*,*,*,* *,*,*,*,*,*,*,*
foreign titleauthor authors      au_id,*,*,*,*,*,* au_id,*,*,*,*,*,*
                                     *,*
```

- 使用法**
- sp_helpkey は、*tabname* テーブルを参照するすべてのプライマリ・キー、外部キー、共通キーの定義についての情報を表示します。なお、*tabname* パラメータが省略されている場合は、データベース内のすべてのキーに関する情報を表示します。これらのキーの定義には、sp_primarykey、sp_foreignkey、sp_commonkey システム・プロシージャを使用します。
 - sp_helpkey は、create table 文によって定義された一意性制約またはプライマリ・キー制約についての情報は表示しません。テーブルにどのような制約が定義されているかを調べるには、sp_helpconstraint を使用します。
 - データベース設計での暗黙的な論理関係を明示的にするためにキーを作成して、アプリケーションがこの情報を使用できるようにします。
 - オブジェクト名を指定すると、sp_helpkey は Adaptive Server の次のようなルールに従って、オブジェクトを検索します。
 - ユーザが所有者名を指定しなかったときでも、指定した所有者名のオブジェクトをユーザが所有していれば、sp_helpkey はそのオブジェクトについてレポートします。

- ユーザが所有者名を指定しなかったときに、ユーザがその名前のオブジェクトを所有していない場合 (ただしデータベース所有者はその名前のオブジェクトを所有している場合)、`sp_helpkey` は、データベース所有者が所有するオブジェクトについてレポートします。
- ユーザもデータベース所有者も、指定された名前のオブジェクトを所有していない場合は、別の所有者がデータベース内にその名前のオブジェクトを所有していたとしても、`sp_helpkey` はエラー状況をレポートします。
- ユーザとデータベース所有者の両方が、指定した名前のオブジェクトを所有しているときに、ユーザがデータベース所有者のオブジェクトにアクセスする場合は、その名前を `dbo.objectname` の形式で指定します。
- このコマンドを実行したユーザとデータベース所有者以外のデータベース・ユーザが所有しているオブジェクトは、“`mary.myproc`” など、その所有者名で修飾してください。

パーミッション

`sp_helpkey` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [create trigger](#)

システム・プロシージャ [sp_commonkey](#)、[sp_foreignkey](#)、[sp_primarykey](#)

sp_helplanguage

説明 特定の代替言語またはすべての言語についての情報をレポートします。

構文 sp_helplanguage [*language*]

パラメータ *language*
情報を表示する代替言語の名前です。

例 **例 1** 代替言語 “french” についての情報を表示します。

```
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,jui,juil,aou,sep,oct,nov,dec
lundi,mardi,mercredi,jeudi,vendredi,samedi,dimanche
```

例 2 インストールされているすべての代替言語についての情報を表示します。

```
sp_helplanguage
```

使用法

- sp_helplanguage は、言語が特定されている場合は指定した言語についてレポートし、言語が指定されていない場合は master.dbo.syslanguages 内のすべての言語についてレポートします。

パーミッション sp_helplanguage は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addlanguage](#)、[sp_droplanguage](#)、[sp_setlangalias](#)

sp_helplog

説明 トランザクション・ログの最初のページが入っているデバイスの名前をレポートします。

構文 sp_helplog

パラメータ なし

例 デバイスの名前として “master” をレポートします。

```
sp_helplog
```

```
In database 'master', the log starts on device 'master'.
```

使用法

- sp_helplog は、現在のデータベース内に格納されている、トランザクション・ログの最初のページが入っているデバイスの名前を表示します。

パーミッション sp_helplog は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> Roles — 現在のアクティブな役割 Keywords or options — NULL Previous value — NULL Current value — NULL Other information — すべての入力パラメータ Proxy information — set proxy が有効な場合は元のログイン名

参照 コマンド [alter database](#)、[create database](#)
システム・プロシージャ [sp_helpdevice](#)、[sp_logdevice](#)

sp_helpobjectdef

説明	(コンポーネント統合サービスのみ) リモート・オブジェクト定義の所有者、オブジェクト、タイプについての情報をレポートします。
構文	sp_helpobjectdef [objname]
パラメータ	<p><i>objname</i></p> <p>sysattributes テーブル内で定義されているとおりのオブジェクトの名前です。この <i>objname</i> は、次のいずれかのフォーマットで指定できます。</p> <ul style="list-style-type: none"> • <i>dbname.owner.object</i> • <i>dbname..object</i> • <i>owner.object</i> • <i>object</i> <p><i>dbname</i> と <i>owner</i> の指定は省略可能です。<i>object</i> は必須です。<i>owner</i> を指定しなければ、<i>owner</i> がデフォルトで現在のユーザ名になります。<i>dbname</i> を指定する場合は、現在のデータベース名を指定してください。さらに <i>owner</i> を指定するか、プレースホルダ <i>dbname..object</i> によってマークを付ける必要があります。複数部分からなる <i>objname</i> は、引用符で囲みます。</p>
例	<p>例 1 現在のデータベースにあるすべてのリモート・オブジェクトの定義を表示します。</p> <pre>sp_helpobjectdef</pre> <p>例 2 データベース所有者が所有しているテーブル <i>tb1</i> のリモート・オブジェクトの定義を表示します。</p> <pre>sp_helpobjectdef "dbo.tb1"</pre>
使用法	<ul style="list-style-type: none"> • <i>objname</i> を指定しなければ、sp_helpobjectdef はすべてのリモート・オブジェクト定義を表示します。 • <i>objname</i> パラメータ内でサーバ名を指定することはできません。
パーミッション	sp_helpobjectdef は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド create table、create existing table、drop table

システム・プロシージャ [sp_addobjectdef](#)、[sp_dropobjectdef](#)、[sp_helpserver](#)

sp_helpremotelogin

説明	特定のリモート・サーバのログイン、またはすべてのリモート・サーバのログインについての情報をレポートします。
構文	sp_helpremotelogin [<i>remoteserver</i> [, <i>remotename</i>]]
パラメータ	<p><i>remoteserver</i> リモート・ログイン情報をレポートさせるサーバの名前です。</p> <p><i>remotename</i> リモート・サーバ上の特定のリモート・ユーザの名前です。</p>
例	<p>例 1 リモート・サーバ GATEWAY のすべてのリモート・ユーザに関する情報を表示します。</p> <pre>sp_helpremotelogin GATEWAY</pre> <p>例 2 ローカル・サーバに認識されているすべてのリモート・サーバのあらゆるリモート・ユーザに関する情報を表示します。</p> <pre>sp_helpremotelogin</pre>
使用法	<ul style="list-style-type: none"> sp_helpremotelogin は、<i>remoteserver</i> を指定していれば、指定したサーバのリモート・ログインについてレポートし、パラメータを指定していない場合にはすべてのサーバについてレポートします。
パーミッション	sp_helpremotelogin は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。
監査	sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addremotelogin](#)、[sp_dropremotelogin](#)、[sp_helpserver](#)

sp_helpprotect

説明	データベース・オブジェクト、ユーザ、グループ、または役割のパーミッションについてレポートします。
構文	<pre>sp_helpprotect [name[, username[, "grant" [,"none" "granted" "enabled" role_name[, permission_name]]]]]</pre>
パラメータ	<p>name テーブル、ビュー、ストアド・プロシージャ、SQLJ ストアド・プロシージャ、SQLJ 関数、ユーザ定義関数の名前か、または現在のデータベース内のユーザ、役割、グループの名前のどちらかです。名前を指定しない場合、sp_helpprotect はデータベース内のパーミッションすべてをレポートします。</p> <p>username 現在のデータベース内のユーザ、グループ、または役割の名前です。</p> <p>grant grant オプションによって name で username に付与されている権限を表示します。username が null の場合、sp_helpprotect は、name の grant オプションによって付与されているすべての権限をリストします。</p> <p>none 付与されたパーミッションを調べるときに、ユーザに付与された役割を無視します。</p> <p>granted 付与されたパーミッションを調べるときに、ユーザに付与された役割すべてに関する情報を含めます。</p> <p>enabled 付与されたパーミッションを調べるときに、ユーザがアクティブにしたすべての役割に関する情報を含めます。</p> <p>role_name role_name によって付与された権限をリストします。</p> <p>permission_name sp_helpprotect で、所定のデータベースについて許可されている特定のパーミッションに関する情報 (付与者の名前、付与されたユーザの名前、テーブル/カラム名、付与可能性) を表示できます。 このパラメータの値には、sysprotects.action カラムの任意の値を指定できます。</p>
例	例 1 次の grant 文と revoke 文を続けて実行した後で sp_helpprotect titles を実行すると、次のように表示されます。


```
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
go
sp_helprotect titles
```

grantor	grantee	type	action	object	column	predicate	grantable
dbo	judy	Grant	Select	titles	All	0	FALSE
dbo	judy	Grant	Update	titles	advance	0	FALSE
dbo	judy	Grant	Update	titles	notes	0	FALSE
dbo	judy	Grant	Update	titles	pub_id	0	FALSE
dbo	judy	Grant	Update	titles	pubdate	0	FALSE
dbo	judy	Grant	Update	titles	title	0	FALSE
dbo	judy	Grant	Update	titles	title_id	0	FALSE
dbo	judy	Grant	Update	titles	total_sales	0	FALSE
dbo	judy	Grant	Update	titles	type	0	FALSE
dbo	judy	Grant	Select	titles	all	0	TRUE

例 2 次の `grant` 文を実行した後で `sp_helprotect` を実行すると、次のように表示されます。

```
grant select, update on titles(price, advance)
to mary
with grant option
go
sp_helprotect titles
```

grantor	grantee	type	action	object	column	predicate	grantable
dbo	mary	Grant	Select	titles	advance	0	TRUE
dbo	mary	Grant	Select	titles	price	0	TRUE
dbo	mary	Grant	Update	titles	advance	0	TRUE
dbo	mary	Grant	Update	titles	price	0	TRUE

例 3 “judy” がデータベース内で割り当てられているすべてのパーミッションを表示します。

```
sp_helprotect judy
```

例 4 “csmith” が `sysusers` テーブルで割り当てられているパーミッションを表示します。また、他のユーザにパーミッションを付与することを許可する `with grant option` を “csmith” が割り当てられているかどうか也表示します。

```
sp_helprotect sysusers, csmith, "grant"
```

grantor	grantee	type	action	object	column	predicate	grantable
dbo	doctor	Grant	Delete	sysusers	All	0	FALSE
dbo	doctor	Grant	Insert	sysusers	All	0	FALSE
dbo	doctor	Grant	References	sysusers	All	0	FALSE

例 5 データベース内で `doctor_role` に割り当てられているパーミッションについての情報を表示します。

```
sp_helprotect doctor
```

grantor	grantee	type	action	object	column	predicate	grantable
dbo	doctor	Grant	Delete	sysusers	All	0	FALSE
dbo	doctor	Grant	Insert	sysusers	All	0	FALSE
dbo	doctor	Grant	References	sysusers	All	0	FALSE

例 6 “csmith” に付与されているすべての役割についての情報を表示します。

```
sp_helprotect csmith, null, null, "granted"
```

grantor	grantee	type	action	object	column	predicate	grantable
dbo	csmith	Grant	Update	sysusers	All	0	FALSE
dbo	doctor	Grant	Delete	sysusers	All	0	FALSE
dbo	doctor	Grant	Insert	sysusers	All	0	FALSE
dbo	doctor	Grant	References	sysusers	All	0	FALSE

```
(1 row affected)
```

```
(return status = 0)
```

例 7 “rpillai” に付与されているすべてのアクティブな役割についての情報を表示します。

```
sp_helprotect rpillai, null, null, "enabled"
```

grantor	grantee	type	action	object	column	predicate	grantable
dbo	public	Grant	Select	sysattributes	All	0	FALSE

```
(1 row affected)
```

```
(return status = 0)
```

例 8 SQLJ 関数に誰でもアクセスできることを通知します。

```
sp_helprotect function_sqlj
Implicit grant to public for SQLJ functions.
```

例 9 sysprotects.action のアクション “Decrypt” を使用します。

```
sp_helprotect @permission_name = "Decrypt"

grantor  grantee  type    action  object    column  predicate  grantable
-----  -
sal      hr_login Grant   Decrypt employee  ssn      0          TRUE
sal      hr_role  Grant   Decrypt employee  ssn      0          FALSE
```

例 10 述部付き権限の名前を出力に表示します。

```
grant select, update, on tabl where col1 = 8 as pred1 to robert
grant select, delete on tabl where col1 = 9 to robert, joffrey
grant select, delete, update on tabl where col2 = 10 as pred2 to role1,
group1
sp_helprotect tabl
```

grantor	grantee	type	action	object	column	predicate	grantable
dbo	joffrey	Grant	Delete	tabl	All	tabl_720002565	FALSE
dbo	joffrey	Grant	Select	tabl	All	tabl_720002565	FALSE
dbo	group1	Grant	Delete	tabl	All	pred2	FALSE
dbo	group1	Grant	Select	tabl	All	pred2	FALSE
dbo	group1	Grant	Update	tabl	All	pred2	FALSE
dbo	role1	Grant	Delete	tabl	All	pred2	FALSE
dbo	role1	Grant	Select	tabl	All	pred2	FALSE
dbo	role1	Grant	Update	tabl	All	pred2	FALSE
dbo	robert	Grant	Delete	tabl	All	tabl_720002565	FALSE
dbo	robert	Grant	Select	tabl	All	pred1	FALSE
dbo	robert	Grant	Select	tabl	All	tabl_720002565	FALSE
dbo	robert	Grant	Update	tabl	All	pred1	FALSE

使用法

- sp_helprotect を実行すると、データベース・オブジェクトに対するパーミッションがレポートされます。username パラメータを指定した場合は、データベース・オブジェクトに対するそのユーザのパーミッションだけがレポートされます。name がオブジェクトでない場合、sp_helprotect は、それがユーザ、グループ、役割、またはパーミッション名であるかどうかを確認します。ユーザ、グループ、または役割である場合は、sp_helprotect がそのパーミッションをユーザ、グループ、または役割ごとにリストします。

- `sp_helprotect` は、現在のデータベース内でだけオブジェクトとユーザを検索します。
- `granted`、`enabled`、`none`、または `role_name` などのオプション値を指定しなければ、Adaptive Server は、指定された現在のユーザがアクティブにしたすべての役割についての情報を返します。
- 指定されたユーザが現在のユーザでない場合、Adaptive Server は、指定されたユーザに付与されているすべての役割についての情報を返します。
- 表示された情報には、指定されたユーザがメンバとなっているグループに対して付与されたパーミッションが常に含まれます。
- パーミッションを付与するとき、システム管理者はオブジェクト所有者として扱われます。システム管理者が他のユーザのオブジェクトにパーミッションを付与すると、`sp_helprotect` の出力に所有者の名前が付与者として表示されます。

`sp_helprotect` および暗号化カラム

`sp_helprotect` では、暗号化カラム、暗号化キー、およびユーザについての情報が次のようにレポートされます。

- テーブルとカラム — `decrypt` パーミッションが付与されているユーザとカラムをレポートする。
- 暗号化キー — `select` パーミッションが付与されているユーザをレポートする。
- ユーザ — `create encryption key` パーミッションが付与されているユーザを示す。

`sp_helprotect` および述部付き権限

`sp_helprotect` は、述部付き権限がある場合、その名前を追加カラムに表示することによって述部付き権限の情報をレポートします。

パーミッション

`sp_helprotect` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド [grant](#)、[revoke](#)

システム・プロシージャ [sp_activeroles](#)、[sp_displayroles](#)

sp_helpsegment

説明 特定のセグメントまたは現在のデータベース内のすべてのセグメントについての情報をレポートします。

構文 sp_helpsegment [segname]

パラメータ *segname*
セグメントの関連情報が必要なときに指定するセグメントの名前です。このパラメータを省略すると、現在のデータベース内のすべてのセグメントに関連する情報が表示されます。

例 1 現在のデータベースのすべてのセグメントに関する情報をレポートします。

```
sp_helpsegment

segment name                status
-----
0 system                    0
1 default                   1
2 logsegment                0
3 seg1                      0
4 seg2                      0
5 seg3                      0
6 seg4                      0
```

例 2 `order_seg` という名前のセグメントについての情報をレポートします。この情報には、このセグメントに対応するデータベース・テーブルとインデックス (テーブルおよびインデックス・レベルで指定された、現在このセグメントを保持しているテーブルとインデックス)、および現在このセグメントにあるオブジェクト (このセグメントに実際に配置されているパーティション) が含まれます。加えて、この例では、このセグメントの総ページ数、空きページ、使用ページ、予約ページをレポートします。

```
sp_helpsegment seg1

segment name                status
-----
3 seg1                      0

device                size                free_pages
-----
pubs_dev1             2.0MB                240

Objects on segment 'seg1' :
```

```

table_name      index_name      indid  partition_name
-----
fictionsales   fictionsales    0     q1
pb_fictionsales pb_fictionsales 0     lov

```

Objects currently bound to segment 'seg1' :

```

table_name  index_name  indid
-----
new_titles  new_titles    0

```

```

total_size  total_pages  free-pages  used_pages  reserved pages
-----
2.0MB       256          240         16          0

```

例 3 default セグメントについての情報をレポートします。キーワードの **default** は引用符で囲みます。この出力は、長さの都合上、一部を抜粋したものです。

```
sp_helpsegment "default"
```

```

segment  name      status
-----
1  default  1

```

```

device      size      free_pages
-----
master      14.0MB   303
pubs_dev1   2.0MB    240
pubs_dev2   2.0MB    232
pubs_dev3   2.0MB    232
pubs_dev4   2.0MB    240

```

Objects on segment 'default' :

```

table_name      index_name      indid  partition_name
-----
au_pix          au_pix          0     au_pix_864003078
au_pix          tau_pix         0     tau_pix_864003078
...
titles          title_idx       0     p1
titles          title_idx       0     p2
titles          title_idx       0     p3
titles          title_idx       0     title_idx_985051514

```

Objects currently bound to segment 'default' :

table_name	index_name	indid		
au_pix	au_pix	0		
...				
titleauthor	titleidind	3		
titles	title_idx	1		
total_size	total_pages	free_pages	used_pages	reserved_pages
22.0MB	2816	1247	1569	0

例 4 トランザクション・ログが保管されているセグメントについての情報をレポートします。

```
1> sp_helpsegment "logsegment"
2> go
```

segment name	status
2 logsegment	0

device	device size
master	14.0MB
pubs_dev1	2.0MB
pubs_dev2	2.0MB
pubs_dev3	2.0MB
pubs_dev4	2.0MB

free_pages
1239

Objects on segment 'logsegment':

table_name	index_name	indid	partition_name
syslogs	syslogs	0	syslogs_8

Objects currently bound to segment 'logsegment':

table_name	index_name	indid
syslogs	syslogs	0

total_size	total_pages	free_pages	used_pages	reserved_pages
22.0MB	2816	1239	13	0

```
(return status = 0)
```

使用法

- `sp_helpsegment` は、`segname` が特定されていれば、指定したセグメントについての情報を表示し、引数が特定されていない場合は、現在のデータベース内のすべてのセグメントについての情報を表示します。
- データベースを最初に作成すると、Adaptive Server は自動的に `system`、`default`、`logsegment` のセグメントを生成します。現在のデータベースにセグメントを追加するには、`sp_addsegment` を使用します。
- 専用のログ・データベースからのログ・セグメントを `segname` パラメータに指定すると、`sp_helpsegment` は、ログ・セグメントにある空きページの数レポートします。
- `system`、`default`、`logsegment` のセグメントには、それぞれ 0、1、2 の番号が付けられます。
- “status” カラムは、領域のデフォルト・プールであるセグメントを示します。特定のセグメントにオブジェクトを作成するには、`sp_placeobject` を使用します。または、`create table` コマンドか `create index` コマンドの `on segment_name` 句を使用して、特定のセグメントにオブジェクトを配置します。
- “indid” カラムは、テーブルにクラスタード・インデックスがない場合は 0、ある場合は 1 です。

パーミッション

`sp_helpsegment` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [create index](#)、[create table](#)

システム・プロシージャ [sp_addsegment](#)、[sp_dropsegment](#)、[sp_extendsegment](#)、[sp_helpdb](#)、[sp_helpdevice](#)、[sp_placeobject](#)

sp_helpserver

説明 特定のリモート・サーバ、またはすべてのリモート・サーバについての情報をレポートします。

構文 `sp_helpserver [server]`

パラメータ `server`
リモート・サーバの関連情報が必要なときに指定するリモート・サーバの名前です。

例 **例 1** リモート・サーバ GATEWAY についての情報を表示します。

```
sp_helpserver GATEWAY
```

例 2 ローカル Backup Server についての情報を表示します。

```
sp_helpserver SYB_BACKUP
name          network_name    security_mechanism  server_principal
              class
              status
id cost
-----
-----
-----
-----
-----
SYB_BACKUP    SYB_BACKUP          NULL                NULL
              NULL
timeouts, no net password encryption, writable, enable login redirection 1
              NULL
```

例 3 ローカル・サーバで認識されているすべてのリモート・サーバについての情報を表示します。

```
sp_helpserver
```

使用法

- `sp_helpserver` は、`master.dbo.sysservers` 内のすべてのサーバに関する情報をレポートします。また、`server` が指定されている場合は、そのリモート・サーバについての情報をレポートします。
- コンポーネント統合サービス (CIS) をインストールすると、`sp_helpserver` は、セキュリティ・メカニズム、サーバ・プリンシパル名、および各サーバのサーバ・クラスをリストします。

パーミッション `sp_helpserver` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addserver](#)、[sp_dropserver](#)、[sp_helpremotelogin](#)、[sp_serveroption](#)

sp_helpsort

説明 Adaptive Server のデフォルトのソート順と文字セットを表示します。

構文 sp_helpsort

パラメータ なし

例 **例1** クラス1(シングルバイト)文字セットの場合、sp_helpsort は、サーバのデフォルトのソート順の名前、サーバの文字セット、サーバのプライマリ・ソート値のテーブルを表示します。7ビットの端末では、次のように表示されます。

```
sp_helpsort
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 { | } ~
```

例2 8ビットの端末では、次のように表示されます。

```
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 { | } ~
! ¢ £ ? ¥ | § ¨ ¯ ° ² ³ ´ µ ¶ · ¸ ¹ º ¼ ½ ¾ ? A
A A A A A A C E E E E I I I I D N O O O O O .: O U U U U Y P s a
```

a a a a a c e e e e i i i i n o o o o o 3 o u u u u y p y

例 3 クラス 2 (マルチバイト) 文字セットの場合は、文字はリストされ
ないで、文字セットの説明が表示されます。次に例を示します。

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.
```

例 4 大文字と小文字を区別しない文字セットの場合、大文字と小文字
を区別しない利用できるソート順の名前およびソート順 ID がリスト
されます。

Name	ID
nocase_eucgb	52
nocase_cp936	52
nocase_gb18030	52
nocase_eucjis	52
nocase_sjis	52
nocase_deckanji	52

使用法 バイナリ・ソート順がデフォルトです。

パーミッション sp_helpsort は、すべてのユーザが実行できます。パーミッション・
チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベ ント	監査オプショ ン	監査対象のコマ ンドまたはアク セス	extrainfo の情報
38	exec_procedure	プロシージャの 実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイ ン名

sp_helptext

説明

コンパイル済みオブジェクトのソース・テキストと、ユーザ定義関数、計算カラム、関数ベース・インデックス定義などのテキストを表示します。

構文

```
sp_helptext objname[,grouping_num][, numlines[, printopts]]
```

パラメータ

objname

ソース・テキストが表示されるコンパイル済みオブジェクトの名前です。コンパイル済みオブジェクトが現在のデータベース内に格納されている必要があります。

grouping_num

objname がプロシージャのグループを表す場合、個別のプロシージャを特定する整数です。このパラメータを指定すると、**sp_helptext** はグループ内の指定されたプロシージャのソース・テキストを表示します。

また、このパラメータは、*printopts* 引数が指定された場合、SQL テキストを生成する開始行番号も指定します。

注意 ビュー、デフォルト、その他のプロシージャ以外のオブジェクトは、グループ化されることはありません。プロシージャのグループには、*number* だけを使用します。

numlines

SQL テキストを生成する行の数を指定します。引数 *printopts* を `showsql` とともに使用すると、*numlines* は表示する SQL テキストの行数を指定します。*printopts* を `context` とともに使用すると、*numlines* は開始行番号を囲むコンテキスト・ブロック幅として処理されます。

printopts

出力フォーマットのさまざまなカンマ区切りプロパティをサポートします。カンマ区切り文字列として、次の印刷オプションのうち、1つ以上を任意の順序で指定できます。

- **showsql** — コンパイル済みオブジェクトのフォーマットされた SQL 出力を生成します。**showsql** を *printopts* リストに含めない場合、このプロパティは呼び出されません。
- **linenumbers** — SQL 出力の各行の行番号を生成します。
- **comments** — コメント・フィールド (`/*<nnn>*/`) として行番号を生成します。これにより、それ以上の編集を行わないで、生成された SQL で必要に応じてコンパイル済みオブジェクトを再作成できるようになります。
- **context** — 指定した開始行番号の周囲にコンテキスト・ブロック出力を生成します。*numlines* パラメータを呼び出さない (`null` を指定する) と、対象の行番号の前後に 5 行のデフォルトのコンテキスト・ブロックが生成されます。
- **noparams** — 自動的に生成されたパラメータ情報を非表示にします。コンパイル済みオブジェクトについて SQL 出力の関連する部分のみを生成するには、この印刷オプションを使用します。
- **ddlgen** — DDL スクリプトとして SQL テキストを生成し、**use database** コマンド、および **drop object** コマンドを使用して出力にプレフィックスを付けます。これにより、プロシージャ、トリガ、ビュー、デフォルト、ルールなど、ほとんどのコンパイル済みオブジェクトを再作成するために必要な SQL をほぼ正確に再生成できます。

印刷オプション **ddlgen** と **context** は互いに排他的な指定子です。同時に使用すると、エラーが発生します。SQL テキストのコンテキスト・ブロックを表示するときに行番号を取得するには、**context** 指定子および **linenumbers** 指定子を使用します。

例 **例 1** `pub_idrule` のソース・テキストを表示します。このルールは `pubs2` データベース内にあるため、このコマンドは `pubs2` から実行します。

```
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]"
```

例 2 `sp_helptext` のソース・テキストを表示します。システム・プロシージャは `sybsystemprocs` データベースに格納されているため、このコマンドは `sybsystemprocs` から実行します。

```
sp_helptext sp_helptext
```

例 3 `number` 引数を指定していない場合、`myproc` グループ動作のソース・テキストを表示します。プロシージャの `number` は、テキストの隣に表示されます。

```
sp_helptext myproc

# Lines of Text
-----
2
number
text
-----
1
create procedure myproc; as select 1
2
create procedure myproc;2 as select 2
(2 rows affected)
```

例 4 `myproc` のソース・テキストを表示します。`myproc` グループのプロシージャを指定しますが、グループ番号は表示しません。

```
sp_helptext myproc, 2

# Lines of Text
-----
1
text
-----
create procedure myproc;2 as select 2
```

例 5 `sp_help` のテキストを生成します。

```
sp_helptext sp_help,NULL,NULLM 'showsql'
```

例 6 行番号を生成し、`sp_help` のテキストを生成します。

```
sp_helptext sp_help, NULL,NULL, 'showsql,linenumbers'
```

例 7 コメント・ブロックに出力を生成し、25 行目以降の 7 行のコンテキスト・ブロックに `sp_help` のテキストを生成します。

```
sp_helptext sp_help,25,7,'showsql,comments,context'
```

例 8 `sp_droptabledef` のテキストを生成し、スタンドアロン DDL スクリプトとしてプロシージャの再作成に使用できる出力を生成します。

```
sp_helptext sp_droptabledef,NULL,NULL,'showsql,ddlgen'
-----
use sybssystemprocs

-----
IF EXISTS (SELECT 1 FROM sysobjects
WHERE name = 'sp_droptabledef'
AND type = 'P'
DROP PROCEDURE sp_droptabledef
-----
/*Sccsud="%Z%generic/sproc/src/%M%I%G%*"*/
/*
**Omni only
*/
create procedure sp_droptabledef
    @tablename varchar(92) /*tablename*/
as begin
    declare @status int
    exec @status = sp_dropobjectdef @tablename
    return(@status)
end
-----
(return status = 0)
```

例 9 区切り識別子を使用して作成されたビューで `sp_helptext` を使用します。ビューを定義する SQL を抽出する場合は、`quoted_identifier` を ON に設定する必要はありません。区切り識別子を使用してオブジェクトを作成する場合は、ON に設定する必要があります。

```
set quoted_identifier ON
-----
create table "t one"
    (c1 int,
    "c two" varchar(10),
    "c three" int)
-----
create table "t two"
    ("t2 one" int,
    "t2 two" varchar(10),
    t2_three int)
```

```

-----
create view "v one" as
  select * from "t one"
  UNION
  select "t2 one","t2 two",t2_three
  from "t two"
-----

```

例 10 Adaptive Server は述部のテキストを表示します。sp_helptext では、述部のユーザ定義名 (ある場合) または内部名を指定できます。次に例を示します。

```

sp_helptext pred1

# Lines of Text
-----
          1
text

-----
grant select on tabl where coll = 5 as pred1 to robert

```

使用法

- sp_helptext は、syscomments のソース・テキストを表示するときに後続スペースをトランケートします。
- sp_helptext は、コンパイル済みオブジェクトによって占有されている syscomments 内のローの数 (それぞれの長さは 255 文字) を表示し、その後にコンパイル済みオブジェクトのソース・テキストを表示します。
- ソース・テキストは char(255) を使用して表示されるため、表示されるテキストには後続スペースがあります。syscomments に格納されているテキストには、これらの後続スペースは含まれていないことがあります。syscomments には、渡されたテキストがそのまま格納されるので、使用したアプリケーションやツールによっては、これらの後続スペースが付加されていない場合もあります。したがって、sp_helptext は、格納されているテキストのコピーを取得する目的では使用しないでください。代わりに、別のツール (defncopy など) を使用します。
- sp_helptext は、現在のデータベースの syscomments テーブル内でソース・テキストを検索します。
- sp_hidetext を使用して、ソース・テキストを暗号化できます。

- `sp_helptext` がプロシージャのグループに対して実行されると、ソース・テキストの他に、`syscomments` から `number` カラムを出力します。
- システム・セキュリティ担当者は、`sp_helptext` を実行するほとんどのユーザに対し、コンパイル済みオブジェクトのソース・テキストが表示されないようにできます。`syscomments` テーブルの `text` カラムに対する `select` パーミッションをオブジェクト所有者またはシステム管理者だけに制限するには、`sp_configure` を使用して `select on syscomments.text column` パラメータを 0 に設定します。この制限は、評価済み設定で `Adaptive Server` を実行する場合に必要となります。評価済み設定の詳細については、『システム管理ガイド』を参照してください。
- `sp_helptext` を `ddlgen` モードで使用する場合でも、`showsql` 印刷オプションが必要です。
- 取得するテキストを含むオブジェクトは、プロシージャが実行されるデータベース内に存在する必要があります。
- テキストが隠されている場合も `syscomments` にない場合も、エラー・メッセージが表示されます。ただし、コンテキスト・ブロック出力を要求し、テキストが見つからないか隠されている場合は、見つからないテキストを示すメッセージが出力されますが、エラーは発生しません。
- 生成済みスクリプトが実行される時、コンパイル済みオブジェクトがすでに存在しない他のオブジェクト(テンポラリ・テーブルなど)の参照を含んでいる場合、`ddlgen` 印刷オプションを使用して生成されたテキストは、そのコンパイル済みオブジェクトを正しく作成できない可能性があります。
- コンパイル済みオブジェクトが `select * 文` を含んでいる場合、通常は、この文が参照するテーブルのカラム・リスト全体が反映されます。
- 引用符で囲んだ識別子を使用して作成したコンパイル済みオブジェクトの `SQL` テキストを生成できます。ただし、コンパイル済みオブジェクトが `select * 文` を含んでいる場合、`Adaptive Server` がテキストを `syscomments` に書き込むと、角カッコで囲んだ識別子を使用した拡張カラム・リストが表示されます。次に例を示します。

```
[this column], [column name with space]
```

コンパイル済みオブジェクト自体のテキストを生成する場合に `set quoted_identifier ON` に設定したり、区切り識別子を使用したりする必要はありません。

パーミッション `sp_helptext` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、オブジェクト所有者、データベース所有者、または <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、オブジェクト所有者、データベース所有者、または <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_checksource](#)、[sp_configure](#)、[sp_hidetext](#)

sp_helpthread

説明 現在のスレッド・プール設定を表示します。

構文 sp_helpthread [*pool_name*]

パラメータ *pool_name*
表示するプールの名前です。*pool_name* が null の場合、sp_helpthread はすべてのプールに関する設定情報を表示します。

例 1 すべてのプールについての情報を表示します。

```
sp_helpthread
name          type      size  idle_timeout
-----
pubs_pool     Engine    2      100
syb_blocking_pool  RTC      4      0
              A pool dedicated to executing blocking calls
syb_default_pool Engine    1      100
              The default pool to run query sessions
syb_system_pool  RTC      4      0
              The I/O and system task pool
```

例 2 pubs_pool ユーザについての情報を表示します。

```
sp_helpthread pubs_pool
name          type      size  idle_timeout  description
-----
pubs_pool     Engine    2      100          NULL

thread_id    osthread_id  state  affinity  instance_id
-----
          12    1248065856  IDLE    NULL      0
          13    1237576000  IDLE    NULL      0
```

使用法

- sp_helpthread は、monThread モニタリング・テーブルからレポートの情報を収集します。

- sp_helpthread は、スレッド・モードでのみ出力を生成します。

パーミッション sp_helpthread は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

sp_helpthreshold

説明	現在のデータベース内のすべてのスレッシュホールド、または特定のセグメントのすべてのスレッシュホールドに対応するセグメント、空き領域値、ステータス、ストアド・プロシージャをレポートします。
構文	<code>sp_helpthreshold [segname]</code>
パラメータ	segname 現在のデータベース内にあるセグメントの名前です。
例	例 1 ログ・セグメントのすべてのスレッシュホールドを表示します。 <pre>sp_helpthreshold logsegment</pre> 例 2 現在のデータベース内にあるすべてのセグメントのスレッシュホールドをすべて表示します。 <pre>sp_helpthreshold</pre> 例 3 デフォルト・セグメントのすべてのスレッシュホールドを表示します。予約語の “default” が引用符で囲まれていることに注意してください。 <pre>sp_helpthreshold "default"</pre>
使用法	<ul style="list-style-type: none">• <code>sp_helpthreshold</code> は、現在のデータベースにあるすべてのセグメントのスレッシュホールド情報を表示します。セグメント名を指定すると、<code>sp_helpthreshold</code> は、指定されたセグメントのすべてのスレッシュホールドを表示します。• <code>status</code> カラムは、ラストチャンス・スレッシュホールドの場合は 1 を表示します。それ以外のスレッシュホールドの場合は、すべて 0 になります。トランザクション・ログを独立したセグメントに保管していないデータベースには、ラストチャンス・スレッシュホールドは割り当てられていません。
パーミッション	<code>sp_helpthreshold</code> は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addthreshold](#)、[sp_droptreshold](#)、[sp_helpsegment](#)、[sp_modifythreshold](#)、[sp_thresholdaction](#)

sp_helpuser

説明 現在のデータベース内の特定のユーザ、グループ、エイリアスに関する情報や、すべてのユーザに関する情報をレポートします。ユーザにより所有されているオブジェクトとユーザ定義データ型も示しています。

構文 sp_helpuser [*name_in_db* [, *display_object*]]

パラメータ

name_in_db

null または現在のデータベースで有効なユーザの名前です。

display_object

現在のデータベースの *name_in_db* により所有されているすべてのオブジェクトとユーザ定義データ型をリストします。*name_in_db* が null の場合、呼び出し元ユーザが所有するオブジェクトとユーザ定義のデータ型がリストされます。オブジェクトの出力には、*object_type* と *object_name* でソートされた *object_name*、*object_type*、および *create_date* が含まれます。ユーザ定義データ型の出力にはユーザ・タイプ名が含まれます。

例 **例 1** 現在のデータベース内のすべてのユーザに関する情報を表示します。

```
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
```

例 2 データベース所有者 (ユーザ名 “dbo”) についての情報を表示します。

```
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
```

例 3 そのユーザ `bill` が所有するオブジェクトを表示します。

```

sp_helpuser bill, display_object
Object_name          Object_type          Create_date
-----
proc_update_titles  stored procedures    Apr 28 2007 04:47PM
author              user table           Apr 27 2007 04:47PM
publisher           user table           Apr 27 2007 05:47PM
titles              user table           Apr 27 2007 06:47PM
vw_author_in_ca     view                 Apr 27 2007 05:47PM

```

例 4 データベース所有者 (DBO) が所有するオブジェクトを表示します。

```

sp_helpuser 'dbo', display_object
Object_name          Object_type          Create_date
-----
enter_key            encryption key       Sep 7 2007 03:37PM
sysalternatives     system table         Jul 17 2007 09:25AM
sysattributes       system table         Jul 17 2007 09:25AM
syscolumns          system table         Jul 17 2007 09:25AM
.....
sysquerymetrics     view                 Jul 17 2007 09:25AM

```

使用法

- `sp_helpuser` は、現在のデータベースのすべてのユーザに関する情報をレポートします。`name_in_db` を指定すると、`sp_helpuser` は、指定したユーザの情報だけをレポートします。
- 指定ユーザが現在のデータベースの `sysusers` テーブル内にリストされていない場合、`sp_helpuser` は、指定ユーザが他のユーザに割り当てられたエイリアスになっていないか、またはグループ名になっていないかを確認します。

パーミッション

`sp_helpuser` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_adduser](#)、[sp_dropuser](#)、[sp_helpgroup](#)

sp_hidetext

説明	指定したコンパイル済みオブジェクトのソース・テキストと、計算カラム、述部、および関数ベース・インデックス・キーのテキストを隠します。また、sp_hidetext は、ユーザ定義関数のテキストを暗号化します。
構文	sp_hidetext [objname[, tabname[, username]]]
パラメータ	<p><i>objname</i> ソース・テキストを隠すコンパイル済みオブジェクトを指定します。</p> <p><i>tabname</i> ソース・テキストを隠すテーブル名またはビュー名を指定します。</p> <p><i>username</i> ソース・テキストを隠すコンパイル済みオブジェクトを所有するユーザ名を指定します。</p>
例	<p>例 1 現在のデータベース内にあるすべてのコンパイル済みオブジェクトのソース・テキストを隠します。</p> <pre>sp_hidetext</pre> <p>例 2 Mary が所有しているユーザ定義のストアド・プロシージャ sp_sort_table のソース・テキストを隠します。</p> <pre>sp_hidetext @objname = "sp_sort_table", @username = "Mary"</pre> <p>例 3 ストアド・プロシージャ pr_phone_list のソース・テキストを隠します。</p> <pre>sp_hidetext "pr_phone_list"</pre> <p>例 4 my_tab テーブルで定義されているすべての検査制約、デフォルト、トリガのソース・テキストを隠します。</p> <pre>sp_hidetext @tabname = "my_tab"</pre> <p>例 5 my_tab テーブルで定義されている my_vu ビューとすべての検査制約、デフォルト、トリガのソース・テキストを隠します。</p> <pre>sp_hidetext "my_vu", "my_tab"</pre> <p>例 6 Tom が所有しているすべてのコンパイル済みオブジェクトのソース・テキストを隠します。</p> <pre>sp_hidetext @username = "Tom"</pre>

使用法

- sp_hidetext は、指定されたコンパイル済みオブジェクトのソース・テキストを隠します。

警告！必ずソース・テキストのバックアップをとってから、sp_hidetext を実行してください。sp_hidetext の実行結果をリバースすることはできません。

- パラメータを何も指定しなければ、sp_hidetext は、現在のデータベース内にあるすべてのコンパイル済みオブジェクトのソース・テキストを隠します。
- sp_hidetext の後で、プラットフォーム間の dump と load を使用した場合、すべての隠しオブジェクトを手動で削除して再作成する必要があります。

パーミッション

sp_hidetext のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage database 権限を持つユーザであることが必要。 すべてのユーザは、sp_hidetext を実行して自分のコンパイル済みオブジェクトのソース・テキストを隠すことができる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データ型の所有者であるか sa_role が付与されたユーザであることが必要。 すべてのユーザは、sp_hidetext を実行して自分のコンパイル済みオブジェクトのソース・テキストを隠すことができる。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

参照

コマンド [dump database](#)、[dump transaction](#)、[load database](#)、[load transaction](#)

マニュアル ソース・テキストを隠すことに関する詳細情報については、『Transact-SQL ユーザーズ・ガイド』を参照してください。

システム・プロシージャ [sp_checksourc](#)

sp_import_qpgroup

説明	抽象プランをユーザ・テーブルから抽象プラン・グループにインポートします。
構文	<code>sp_import_qpgroup tab, usr, group</code>
パラメータ	<p>tab プランのコピー元テーブルの名前です。<code>dbname..tablename</code> の形式で、データベース名を指定できますが、所有者の名前を指定することはできません。合計の長さは、最大 255 文字です。</p> <p>usr 抽象プランがインポートされるときに、抽象プランに割り当てられる ID を所有するユーザの名前です。</p> <p>group インポートされるプランが含まれている抽象プラン・グループの名前です。</p>
例	プランを <code>moveplans</code> テーブルから <code>new_plans</code> グループにコピーします。次に、データベース所有者のユーザ ID を、コピーされたプランに割り当てます。 <pre>sp_import_qpgroup moveplans, dbo, new_plans</pre>
使用法	<ul style="list-style-type: none">• <code>sp_import_qpgroup</code> は、プランを、ユーザ・テーブルから <code>sysqueryplans</code> 内の抽象プラン・グループにコピーします。<code>sp_export_qpgroup</code> と一緒に使用すると、サーバとデータベースの間で抽象プラン・グループをコピーできます。または、あるユーザに属しているプランをコピーして、他のユーザの ID を、コピーされたプランに割り当てることができます。• <code>sp_import_qpgroup</code> は、プロシージャが実行されるときに抽象プラン・グループが存在しない場合、抽象プラン・グループを作成します。• <code>sp_import_qpgroup</code> が実行されるときに抽象プラン・グループが存在していると、指定されたユーザのプランは一切含むことができません。<code>sp_import_qpgroup</code> は、グループ内にすでにクエリが存在するかどうかを確認するためにクエリ・テキストを検査することはありません。すでにユーザ用のプランがいくつか存在するグループにユーザのプランをインポートしたい場合は、次の手順に従います。<ul style="list-style-type: none">• <code>sp_import_qpgroup</code> を使用して、新しいプラン・グループにプランをインポートします。

- [sp_copy_all_qplans](#) を使用して、新しく作成されたグループからコピー先グループにプランをコピーします。
[sp_copy_all_qplans](#) は、重複するプランが作成されていないことを確認するためにクエリを検査します。
- インポート用に作成したグループが必要でなくなれば、[sp_copy_all_qplans](#) によってそのグループ内のプランを削除してから、[sp_drop_qpgroup](#) によってそのグループを削除します。
- 抽象プランをバルク・コピーするために空のテーブルを作成するには、次の構文を使用してください。

```
select * into load_table
from sysqueryplans
where 1 = 2
```

パーミッション

[sp_import_qpgroup](#) のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 manage abstract plans 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データ型の所有者であるか sa_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの **event** カラムと **extrainfo** カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

コマンド [create plan](#)

システム・プロシージャ [sp_copy_all_qplans](#)、[sp_copy_qplan](#)、[sp_drop_all_qplans](#)、[sp_drop_qpgroup](#)、[sp_export_qpgroup](#)、[sp_help_qpgroup](#)

sp_indsuspect

説明	ソート順変更後のリカバリ時に、 <code>suspect</code> (疑わしい) としてマーク付けされたインデックスがないか、ユーザ・テーブルを検査します。
構文	<code>sp_indsuspect [tab_name]</code>
パラメータ	<code>tab_name</code> 検査されるユーザ・テーブルの名前です。
例	<code>suspect</code> としてマーク付けされたインデックスがないかどうか、 <code>newaccts</code> テーブルを調べます。 <code>sp_indsuspect newaccts</code>
使用法	<ul style="list-style-type: none"> パラメータを指定しないで <code>sp_indsuspect</code> を実行すると、ソート順変更の結果として再構築されなければならないインデックスを持つ、現在のデータベース内にあるテーブルがすべてリストされます。<code>tab_name</code> パラメータを指定すると、<code>sp_indsuspect</code> は指定されたテーブルを検査し、ソート順変更後のリカバリ時に <code>suspect</code> としてマーク付けされたインデックスがないかどうかを確認します。 疑わしいインデックスをすべてリストするには、<code>sp_indsuspect</code> を使用します。テーブル所有者またはシステム管理者は、<code>dbcc reindex</code> を使用すると、リストされているインデックスの整合性を検査し、必要に応じてインデックスを再構築できます。
パーミッション	<code>sp_indsuspect</code> は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド dbcc](#)

sp_jreconfig

説明

Java PCA/JVM を管理します。引数およびディレクティブの有効化と無効化、設定値の変更、設定値のレポートを行います。

注意 `pca_jvm_module_path`、`pca_jvm_work_dir`、`pca_jvm_dbg_agent_port`、`pca_jvm_java_dbg_agent_suspend`、`pca_jvm_java_options`、および `pca_jvm_netio` の各引数は、問題なく変更できます。Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、`sp_jreconfig` を使用してその他の引数またはディレクティブを変更しないでください。

構文

```
sp_jreconfig {
    add array_arg, new_string |
    array_clear array_arg |
    array_enable array_arg |
    array_disable array_arg |
    delete array_arg, string_value |
    disable { directive | argument | array_arg, string_value } |
    enable { directive | argument | array_arg, string_value } |
    list { list_type [, formatted ] | units | units, units_type[, formatted ] } |
    reload_config |
    report { directive[, formatted ] | directive, args[, formatted ]
           | argument[, formatted ] } |
    update { argument, old_value, new_value }
```

パラメータ

add

新しい引数を引数配列に追加します。`units_type` が array である引数とのみ、`add` を使用できます。

array_arg

`units_type` が array である引数の名前です。

new_string

新しい配列要素の文字列値です。

array_clear

引数配列のすべての要素を削除します。

array_enable

引数配列のすべての要素を有効にします。各配列要素を有効に設定します。

array_disable

引数配列のすべての要素を無効にしますが、削除しません。各要素を無効に設定します。

delete

引数配列から既存の要素を削除します。*units_type* が array である引数とのみ、**delete** を使用できます。

disable

指定したディレクティブまたは引数を無効にします。

string_value

削除、有効化、または無効化の対象となる、指定した引数配列の配列要素を指定します。

directive

有効なディレクティブの名前です。

argument

有効な引数の名前です。

enable

ディレクティブまたは引数を有効にします。

list

sp_jreconfig list, **directives**、**sp_jreconfig list**, **enabled** のように使用して、関連する引数のグループをリストします。また、**sp_jreconfig list**, **units**, **string** のように使用して、特定の型のすべての引数をリストします。*units_types* のすべての現在値を確認するには、**sp_jreconfig list**, **units** を使用します。

formatted

表示されるリストを読みやすいようにフォーマットします。長い値がトランケートされることがあります。

注意 フォーマットされたレポートでは、読みやすくするためにワイド・カラムがトランケートされることがあります。また、カラム見出しが上書きされ、実際のテーブル名に一致しない場合があります。出力を解析する場合やデータのトランケーションが適切でない可能性がある場合は、レポートをフォーマットしないでください。

list_type

リストのタイプを指定します。有効な値は次のとおりです。

- **directives** — ディレクティブのリスト
- **enabled** — 有効な引数のリスト
- **disabled** — 無効な引数のリスト
- **argnames** — 引数名、各引数の *units_type*、および各引数が属するディレクティブのリスト

units

list とともに使用すると、現在使用している *units_type* のリストを生成します。

units_type

引数の型です。各引数には、型を指定する *units_type* があります。値は次のとおりです。

- **switch**
- **string**
- **number**
- **array**

reload_config

sybpcidb テーブルの設定をメモリに再ロードします。『Adaptive Server Enterprise における Java』の「第2章 Java 環境の管理」の「*sybpcidb* へのデフォルト設定値の復元」を参照してください。

report

指定した引数に基づいてレポートを作成します。通常は、引数の現在値と引数があるかどうかを確認するため、引数のレポートの生成に使用されます。ディレクティブまたはその引数のレポートの生成にも使用できます。

directive

有効なディレクティブです。

args

指定したディレクティブの引数名のリストを生成するために、レポートで使用されるキーワードです。次に例を示します。

```
sp_jreconfig report, "PCA_JVM", "args"
```

update

units_type が `string`、`number`、または `array` である引数の、文字列または数値を変更します。*units_type* が `switch` である引数は変更できません。

old_value

更新される既存の引数または配列要素を示す文字列または数値です。

new_value

新しい引数または配列要素を定義する文字列または数値です。

例

例 1 `PCA_JVM_OPT` ディレクティブのレポートをフォーマットして生成します。

```
sp_jreconfig "report", "PCA_JVM_OPT", "formatted"
```

例 2 `PCA_JVM_OPT` ディレクティブの引数について、レポートを生成します。

```
sp_jreconfig "report", "PCA_JVM_OPT", "args"
```

例 3 引数 `pca_jvm_netio` について、レポートを生成します。

```
sp_jreconfig "report", "pca_jvm_netio"
```

例 4 `"pca_jvm"` に一致するすべての引数について、レポートを生成します。引数の部分名を指定すると、一致するすべての引数についてレポートが生成されます。

```
sp_jreconfig "report", "pca_jvm_"
```

例 5 リストの生成 — すべてのディレクティブとその状態 (有効/無効) のリストを表示します。

```
sp_jreconfig "list", "directives"
```

例 6 リストの生成 — すべての引数、その単位のタイプ、およびディレクティブのリストを表示します。

```
sp_jreconfig "list", "argnames", "formatted"
```

例 7 リストの生成 — 現在有効になっているすべての引数のリストを表示します。

```
sp_jreconfig "list", "enabled"
```

例 8 リストの生成 — すべての配列引数のリストをフォーマットして表示します。

```
sp_jreconfig "list", "units", "array", "formatted"
```

例 9 リストの生成 — 引数の単位のタイプのリストを表示します。このコマンドのレポートは、デフォルトでフォーマットされます。“-formatted” オプションを使用すると、エラーが発生します。

```
sp_jreconfig "list", "units"
```

例 10 ディレクティブおよび引数の有効化 — PCA_JVM_WORK_DIR ディレクティブを有効にします。ディレクティブをユニークに識別するのに十分な情報が含まれている場合は、ディレクティブの部分名を使用できます。

```
sp_jreconfig "enable", "PCA_JVM_WORK_DIR"  
sp_jreconfig "enable", "WORK_DIR"
```

例 11 ディレクティブおよび引数の有効化 — pca_jvm_netio 引数を有効にします。

```
sp_jreconfig "enable", "pca_jvm_netio"
```

例 12 ディレクティブおよび引数の無効化 — WORK_DIR ディレクティブを無効にします。この例では、ディレクティブの部分名を使用します。それには、ディレクティブをユニークに識別するのに十分な情報が含まれている必要があります。

```
sp_jreconfig "disable", "WORK_DIR"
```

注意 ディレクティブを無効にすると、その引数も無効になったように動作しますが、引数のベースの状態は変更されません。

例 13 ディレクティブおよび引数の無効化 — pca_jvm_netio 引数を無効にします。

```
sp_jreconfig "disable", "pca_jvm_netio"
```

例 14 ディレクティブおよび引数の無効化 — PCA_JVM_WORK_DIR の配列要素を無効にします。パスは必須ですが、パーミッション・マスクは必須ではありません。『Adaptive Server Enterprise における Java』の「第 8 章 Java を使用したファイルおよびネットワークへのアクセス」を参照してください。

```
sp_jreconfig "disable", "pca_jvm_work_dir",  
"/some/path"
```

例 15 文字列、数字、および配列の引数の更新 — 文字列引数を更新します。この例では、pca_jvm_log_filename 引数のファイル・ロケーションを更新します。

```
sp_jreconfig "update", "pca_jvm_log_filename", "/old/path/filename.log",
"/new/path/filename.log"
```

注意 update オプションは、ディレクティブまたはスイッチ引数では使用できません。これらは変更できないためです。

例 16 文字列、数字、および配列の引数の更新 — 数字引数を更新します。ストアド・プロシージャでは、数値を文字列として処理するため引用符で囲む必要があります。Adaptive Server では、これらを数値として格納します。

```
sp_jreconfig "update", "pca_jvm_min_port", "1026",
"2056"
```

例 17 文字列、数字、および配列の引数の更新 — PCA_JVM_WORK_DIR ディレクティブでは、work_dir 値はパスとオプションのパーミッション・マスクで構成されます。パーミッション・マスクはオプションですが、work_dir を指定するため元の文字列パスを含める必要があります。パーミッション・マスクはオプションです。指定しない場合、デフォルトのマスクである 8 進形式の 0666 が使用されます。例 a ではパーミッション・マスクを設定しないで、デフォルトのマスクを使用します。例 b および例 c では、パーミッション・マスク 0644 を設定します。

```
[a] sp_jreconfig "update", "pca_jvm_work_dir",
"/old/path", "/new/working/directory"
```

```
[b] sp_jreconfig "update", "pca_jvm_work_dir",
"/old/path", "/new/working/directory(u=rw,go=r)"
```

```
[c] sp_jreconfig "update", "pca_jvm_work_dir",
"/old/path", "/new/working/directory(u+w,ugo+r)"
```

例 18 配列要素の追加 — PCA_JVM_WORK_DIR ディレクティブの pca_jvm_work_dir 引数配列に新しい要素を追加します。例 a ではデフォルトのマスクを使用します。例 b および例 c ではパーミッション・マスク 0644 を設定します (マスクは左から右に評価されます)。

```
[a] sp_jreconfig "add", "pca_jvm_work_dir",
"/new/working/directory"
```

```
[b] sp_jreconfig "add", "pca_jvm_work_dir",
"/new/working/directory(u=rw,go=r)"
```

```
[c] sp_jreconfig "add", "pca_jvm_work_dir",
"/new/working/directory(u+w,ugo+r)"
```

例 19 配列要素の削除 — `pca_jvm_work_dir` の配列要素を削除します。

```
sp_jreconfig "delete", "pca_jvm_work_dir",
            "/new/working/directory"
```

注意 `PCA_JVM_WORK_DIR` ディレクティブの `pca_jvm_work_dir` の要素を削除するとき、指定した文字列がレコードをユニークに識別する場合は、部分文字列を指定できます。パーミッション・マスクは必須ではありません。元の `work_dir` 要素が特定のパーミッション・マスクを使用して定義されている場合でも、パスを指定するだけで十分です。

例 20 配列のすべての要素の有効化または無効化 — `pca_jvm_work_dir` 配列のすべての要素を有効にします。

```
sp_jreconfig "array_enable", "pca_jvm_work_dir"
```

例 21 配列のすべての要素の有効化または無効化 — `pca_jvm_work_dir` 配列のすべての要素を無効にします。

```
sp_jreconfig "array_disable", "pca_jvm_work_dir"
```

例 22 配列のすべてのレコードのクリア — `pca_jvm_work_dir` 配列のすべてのレコードを削除し、空の配列を作成します。

```
sp_jreconfig "array_clear", "pca_jvm_work_dir"
```

例 23 デフォルトの設定値の再ロード — `sybpcidb` に格納されている設定値をメモリにロードします。

```
sp_jreconfig "reload_config"
```

使用法

ディレクティブと引数の有効化と無効化

ディレクティブの有効と無効は切り替えられます。

- **ディレクティブが有効になっている場合** — Adaptive Server は各引数の設定値 (有効または無効) を使用します。この値は `sybpcidb` に保存されています。
- **ディレクティブが無効になっている場合** — Adaptive Server は、各引数の設定値 (有効または無効) を無視し、各引数のベース値が `sybpcidb` に保持されているにもかかわらず、ディレクティブのすべての引数を無効として扱います。

引数は、有効/無効を個別に切り替えることができます。引数の型は次のとおりです。

- *switch* — これらの引数は、機能のオンまたはオフを切り替えられます。たとえば、ロギング用の引数が有効になっている場合は、ログ・ファイルが生成され、無効になっている場合は、ログ・ファイルが生成されません。
- *string* — これらは文字列および数値用の引数です。文字列または数字の引数を有効にすると、Adaptive Server は設定値を使用します。文字列または数字の引数を無効にすると、Adaptive Server は設定値を無視し、デフォルト値を使用します。設定値とデフォルト値は同一である場合も、異なる場合もあります。
- *array* — 配列引数は、関連する文字列引数の集まりであり、それぞれ個別に有効または無効にすることができます。個別の文字列引数(または要素)を無効にすると、その値は無視され、要素が削除された場合と同様に動作します。有効にすると、引数値はコレクションに含まれ、アクティブになります。

配列引数は、必要に応じて有効または無効にすることができます。値を削除して後で再入力する必要はありません。

表 1-13: sp_jreconfig の設定ディレクティブ

ディレクティブ	説明
PCA_JVM_MODULE_PATH	JVM 共有オブジェクト・ライブラリへのパス。Sybase から提供されたもの以外の JRE を使用する場合は、PCA/JVM がアクセスできるロケーションを示すように、この引数を設定する必要があります。絶対パスまたは \$\$SYBASE を拡張する相対パスを指定できます。絶対パスを使用する場合は、“/” (UNIX の場合) または “\” (Windows の場合) で始まるパスを使用します。それ以外の場合、Adaptive Server は相対パスとして \$\$SYBASE の下位にあるものと見なします。
PCA_JVM_OPT	Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、デフォルト値は変更しないでください。
PCA_JVM_DIR_OPTIONS	JVM によって使用される ROOT および TEMP ディレクトリのディレクトリ定義。この値は、システムに精通したユーザであるか Sybase 製品の保守契約を結んでいるサポート・センタから指示がある場合を除いて、変更しないでください。 警告！ このディレクティブは慎重に使用してください。PCA_JVM_DIR_OPTIONS ディレクティブの <code>pca_jvm_tmp_dir</code> は、必ずシステムのテンポラリ・ディレクトリを示す必要があります。このロケーションを変更すると、重大なセキュリティ上のリスクが発生する可能性があります。JVM では、このディレクトリで、読み込みおよび書き込みのためにファイルを開いたり、ファイルを作成したりできます。

ディレクティブ	説明
PCA_JVM_WORK_DIR	信頼された JVM ディレクトリを設定します。この引数は、Java プログラム・クラスが特定のファイル I/O 操作を実行できる、ファイル・システム上の個別のロケーションの集まりで構成されます。各ディレクトリには、各ディレクトリで許可されるファイル I/O 操作を定義する、オプションのパーミッション・マスクを指定できます。
PCA_JVM_MIN_JNI_VERSION	JNI バージョンとの最小下位互換性。
PCA_JVM_LOGGING	JRE/VM ロギング・オプション。
PCA_JVM_EXT_CLASS_LOADER	グローバルおよびデータベース拡張クラス・ローダ。
PCA_JVM_JAVA_OPTIONS	Java 起動オプション (通常および拡張)。
PCA_JVM_JAVA_DBG_AGENT_PORT	Java VM デバッグ・エージェントのポート番号 (Java デバッガによる Java アプリケーションのデバッグに使用)。詳細については、『Adaptive Server Enterprise における Java』を参照してください。
PCA_JVM_SYS_DEVICE_PATH	プラットフォーム固有のシステム・デバイス・ディレクトリ (Solaris の場合に必須)。

表 1-14: PCA_JVM_MODULE_PATH 引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_module_path	string	プラットフォームによって異なる	有効	\$\$SYBASE の下位の相対パスまたは完全に修飾されたファイル名を使用して、JVM 共有ライブラリのロケーションを指定します。

表 1-15: PCA_JVM_OPT 引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_abort	switch	On	有効	(重大な) 障害の場合に abort(2) をアボートします。
pca_jvm_allow_unchecked_socketops	switch	該当なし	無効	オフになっているソケット操作を許可します。
pca_jvm_debug	switch	該当なし	無効	PCA_DEBUG 要求をレポートします。
pca_jvm_except	switch	該当なし	有効	例外 PCA/VM JNI/JVM 呼び出しをレポートします。
pca_jvm_heap_ratio	string	0.3	有効	VM ヒープ/ PCI メモリの割合。
pca_jvm_jvmti	switch	該当なし	無効	Java VM ツール・インタフェース。

引数	単位の タイプ	デフォルト 値	デフォ ルトの 状態	説明
pca_jvm_min_port	number	1026	有効	VM ネットワーク・サポートを許可 します。
pca_jvm_netio	switch	該当なし	無効	VM ネットワーク・サポートを許可 します。
pca_jvm_report	switch	該当なし	無効	PCA/VM JNI/JVM 呼び出しをレポー トします。
pca_jvm_security_manager_enable d	switch	該当なし	無効	PCA/JVM の SecurityManager を有効 にします。
pca_jvm_sigcache_density	number	100	有効	PCA/VM 署名キャッシュのター ゲット密度。
pca_jvm_sigcache_enabled	switch	該当なし	有効	PCA/VM 署名キャッシュを有効に します。
pca_jvm_sigcache_fixed_ratio	number	50	有効	PCA/VM 署名キャッシュの固定サ イズの割合 (パーセント)。
pca_jvm_sigcache_freeboard	number	30	有効	キャッシュ・スweepの PCA/VM 署名キャッシュの領域リカバリの 割合 (パーセント)。
pca_jvm_sigcache_size	number	512	有効	PCA/VM 署名キャッシュのサイズ (KB)。
pca_jvm_sigcache_size_type	number	1	有効	PCA/VM 署名キャッシュの size_type が 0 の場合は AS_PCT、1 の場合は KB、2 の場合は MB です。
pca_jvm_sigcache_washcycle	number	1000	有効	PCA/VM 署名キャッシュのウォッ シュ・デーモンのサイクル・タイ ム (ミリ秒)。
pca_jvm_sigcache_washdaemon	switch	該当なし	無効	PCA/VM 署名キャッシュのウォッ シュ・デーモンを有効にします。
pca_jvm_strace	switch	該当なし	有効	エミュレートされない VM ハンド ルでスタック・トレースを生成し ます。

表 1-16: PCA_JVM_DIR_OPTIONS 引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_root_dir	string	プラットフォームによって異なる	有効	システムのルート・ディレクトリへの絶対パス。ファイル I/O に必須です。
pca_jvm_tmp_dir	string	プラットフォームによって異なる	有効	システムのテンポラリ・ディレクトリへの絶対パス。ファイル I/O に必須です。

表 1-17: PCA_JVM_WORK_DIR 引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_work_dir	array	プラットフォーム固有	無効	JVM によるファイル I/O 操作が許可される絶対パス (およびオプションのパフォーマンス・マスク)。 『Adaptive Server Enterprise における Java』の「第 8 章 Java を使用したファイルおよびネットワークへのアクセス」を参照。

表 1-18: PCA_JVM_MIN_JNI_VERSION 引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_min_jni_version	string	‘JNI_VERSION_1_2’	有効	JNI バージョンとの最小下位互換性。

表 1-19: PCA_JVM_LOGGING 引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_ase_logging	switch	該当なし	有効	Adaptive Server のロギングを設定します。

引数	単位の タイプ	デフォルト値	デフォ ルトの 状態	説明
pca_jvm_log_filename	string	'/tmp/Java_vm.log1'	無効	VM がロギングに使用する完全に修飾されたファイル名。

表 1-20: PCA_JVM_EXT_CLASS_LOADER 引数

引数	単位の タイプ	デフォルト値	デフォ ルトの 状態	説明
pca_jvm_ext_class_loader_global	array	none	無効	グローバル拡張クラス・ローダ。
pca_jvm_ext_class_loader_dbase	array	none	無効	データベース拡張クラス・ローダ。

表 1-21: PCA_JVM_JAVA_OPTIONS 引数

引数	単位の タイプ	デフォルト値	デフォ ルトの 状態	説明
pca_jvm_java_options	array	"-Djava.awt.headless=true"	有効	ヘッドレス・モードで Java を実行します。
pca_jvm_java_options	array	"-Djava.compiler=JIT"	有効	JIT コンパイルと最適化を強制的に実行します。
pca_jvm_java_options	array	"-XX:+CITune:"	無効	JIT コンパイラに要する時間 (1.4 のみ)。
pca_jvm_java_options	array	"-XX:+Use AltSigs"	無効	このオプションは J2SE のクラッシュを生じる可能性があります。
pca_jvm_java_options	array	"-XX:CodeCacheExpansionSize=512000"	有効	コード・キャッシュの拡張サイズ。
pca_jvm_java_options	array	"-Xbatch"	無効	バックグラウンド・コンパイルを無効にします。
pca_jvm_java_options	array	"-Xcheck:jni"	有効	JNI 関数の追加チェックを実行します。
pca_jvm_java_options	array	"-Xfuture"	無効	厳密なチェックを実行します。今後、デフォルト設定になる予定です。
pca_jvm_java_options	array	"-Xincgc"	無効	インクリメンタル・ガーベジ・コレクションを有効にします。

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_java_options	array	“-Xint”	無効	解釈されたモードの実行のみ。
pca_jvm_java_options	array	“-Xloggc:./myGClog”	無効	ファイルにタイム・スタンプ付きで GC ステータスのログを取ります。
pca_jvm_java_options	array	“-Xmixed”	無効	混合モードの実行 (デフォルト)。
pca_jvm_java_options	array	“-Xms64m”	無効	初期の Java ヒープ・サイズを設定します。
pca_jvm_java_options	array	“-Xmx64m”	無効	最大の Java ヒープ・サイズを設定します。
pca_jvm_java_options	array	“-XnoClassgc”	無効	クラス・ガーベジ・コレクションを無効にします。
pca_jvm_java_options	array	“-Xprof”	無効	CPU プロファイリング・データを出力します。
pca_jvm_java_options	array	“-Xrs”	無効	Java/VM による OS シグナルの使用を削減します。
pca_jvm_java_options	array	“-Xshare:auto”	無効	共有クラス・データを auto、off、または on に設定します。
pca_jvm_java_options	array	“-Xss64m”	無効	Java スレッド・スタック・サイズを設定します。
pca_jvm_java_options	array	-XX:MaxPermSize	無効	永久ヒープの最大サイズを設定します。
pca_jvm_java_options	array	“-enablesystemassertions”	有効	Java/VM システム・アサーションを有効にします。 Sun HotSpot (TM) JavaVM を使用するプラットフォームにのみ適用されます。
pca_jvm_java_options	array	“-esa”	有効	すべてのシステム・アサーションを有効にします。 Sun HotSpot (TM) JavaVM を使用するプラットフォームにのみ適用されます。
pca_jvm_java_options	array	“-verbose:class”	無効	JRE/VM 内のクラス・ロード。
pca_jvm_java_options	array	“-verbose:gc”	無効	ガーベジ・コレクションの統計。
pca_jvm_java_options	array	“-verbose:jni”	無効	Java ネイティブ・インタフェース (JNI) の呼び出し。

表 1-22: PCA_JVM_JAVA_DBG_AGENT_PORT 引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_java_dbg_agent_port	number	8000	無効	ポート番号および Java VM デバッグ・エージェントを設定します。
pca_jvm_java_dbg_agent_suspend	switch	該当なし	無効	有効にすると、Java VM デバッグ・エージェントがサスペンドの状態では起動します。

表 1-23: PCA_JVM_SYS_DEVICE_PATH 引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pca_jvm_sys_device_path	array	プラットフォームによって異なる	プラットフォームによって異なる	Sun OS の内部システム・オプション。変更しないでください。

パーミッション `sp_jreconfig` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server configuration</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [ストアド・プロシージャ `sp_pciconfig`](#)

sp_ldapadmin

説明 LDAP URL 検索文字列の作成またはリスト、LDAP URL 検索文字列またはログインの確認、アクセス・アカウントとチューニング可能な LDAPUA 関連パラメータの指定を行います。

構文 sp_ldapadmin *command* [, *option1* [, *option2*]]
有効な *command* [, *option1* [, *option2*]] オプションは、次のとおりです。

```
'set_primary_url', 'url'  
'set_secondary_url', 'url'  
'set_dn_lookup_url', 'url'  
'set_secondary_dn_lookup_url', 'url'  
'set_access_acct', 'distinguished_name', 'password'  
'set_secondary_access_acct', 'distinguished_name', 'password'  
'set_failback_interval', time_in_minutes  
'suspend', {'primary' | 'secondary'}  
'activate', {'primary' | 'secondary'}  
'list'  
'list_urls'  
'list_access_acct'  
'check_url', 'url'  
'reinit_descriptors'  
'check_login', 'name'  
'set_timeout', timeout_in_milli_seconds  
'set_log_interval', log_interval_in_minutes  
'set_num_retries', num_retries  
'set_max_ldapua_native_threads', max_ldapua_native_threads  
'set_max_ldapua_desc', max_ldapua_desc  
'set_abandon_ldapua_when_full', {true|false}  
'starttls_on_primary', {true|false}  
'starttls_on_secondary', {true|false}  
'help'
```

パラメータ set_primary_url, '*ldapurl*'
指定された検索文字列 *ldapurl* を作成します。正確に 1 つのプライマリ検索文字列を作成できます。
ldapurl の構文は次のとおりです。


```
ldapurl::=ldap://host:port/node?attributes?base | one | sub?filter
```

各パラメータの意味は次のとおりです。

- *host* – LDAP サーバのホスト名。
- *port* – LDAP サーバのポート番号。
- *node* – 検索を開始するオブジェクト階層内でのノードを指定する。
- *attributes* – 結果セットで返す属性のリスト。属性リストは、LDAP サーバによって異なることがある。
- *base* – ベース・ノードの検索を指定し、検索基準を修飾する。
- *one* – 検索基準を修飾する。*base* は、ベース・ノードの検索を指定します。*one* は、*node* で指定したノードとその 1 つ下のレベルのノードの検索を指定します。*sub* は、*node* で指定したノードとその下位レベルのすべてのノードの検索を指定します。
- *sub* – *node* で指定されたノードとその下位レベルのすべてのノードの検索を指定する。
- *filter* – 認証する属性を指定する。フィルタは、“uid=”のように簡潔にしたり、“(uid=*)(ou=group)”のように複合したりすることもできます。構文は LDAP サーバに応じて異なり、ログイン名の記述にワイルドカード (*) を使用します。

```
set_secondary_url, { 'ldapurl' | null }
```

指定されたセカンダリ検索文字列 *ldapurl* を作成するか、またはセカンダリ検索文字列を作成しません。正確に 1 つのセカンダリ検索文字列を作成できます。

```
set_dn_lookup_url, distinguished_name_url
```

set_dn_lookup_url を NULL 以外の値に設定した場合、LDAP ディレクトリ・サーバでのログインの認証には、検索された識別名アルゴリズムが使用されます。

distinguished_name_url の最大長は 255 文字です。これを使用して、ログイン名に関連する識別名を検索します。

```
set_secondary_dn_lookup_url, distinguished_name_url
```

set_secondary_dn_lookup_url を NULL 以外の値に設定した場合、LDAP ディレクトリ・サーバでのログインを認証するために、指定したセカンダリ識別名アルゴリズムを作成します。

distinguished_name_url の最大長は 255 文字です。これを使用して、ログイン名に関連する識別名を検索します。

set_access_acct, account_distinguished_name, account_password

Adaptive Server が検索やその他の読み込み専用の管理作業を実行するときに使用する ID とパスワードを指定します。ID は識別名の形式で指定します。LDAP サーバでこのユーザを認証するには、*account_distinguished_name* を使用します。*account_distinguished_name* と *account_password* は、それぞれ 255 文字に制限されています。

set_secondary_access_acct, account_distinguished_name, account_password

Adaptive Server が検索やその他の読み込み専用の管理作業を実行するときに使用するセカンダリ ID とパスワードを作成します。ID は識別名の形式で指定します。LDAP サーバでこのユーザを認証するには、*account_distinguished_name* を使用します。*account_distinguished_name* と *account_password* は、それぞれ 255 文字に制限されています。

set_failback_interval

問題が発生している LDAP サーバの有無を、Adaptive Server のハウスキーピング・ユーティリティがチェックする間隔を設定します。

suspend, {'primary' | 'secondary'}

認証のためにプライマリまたはセカンダリ URL の使用をサスペンドします。

activate, {'primary' | 'secondary'}

認証のためにプライマリまたはセカンダリ URL のセットの使用を有効にします。

list

LDAP 検索文字列を表示します。

list_urls

LDAP URL 検索文字列を表示します。

list_urls

LDAP URL 検索文字列を表示します。

list_access_acct

LDAP アクセス・アカウント識別名セットを表示します。

check_url, 'ldapurl'

LDAP URL 検索文字列を確認します。ユーザ・アカウントがあるかどうか確認できますが、ユーザの認証は行いません。

check_login, login_name

既存の LDAP URL 検索文字列のユーザ・アカウントを確認します。ユーザの認証は行いません。

'set_timeout' *timeout_in_milli_seconds*

認証要求が放棄されるまで Adaptive Server が LDAP サーバからの応答を待機する時間をミリ秒単位で設定します。

`set_timeout` のデフォルト値は 10,000 ミリ秒 (10 秒) です。有効な値は、1 ~ 3,600,000 (1 時間) です。

'set_log_interval', *log_interval*

ログ間隔を、0 ~ 480 分の分単位で設定します。デフォルト値は 3 分です。0 を指定すると、すべてのメッセージが出力されます。

`set_num_retries`, *num_retries*

一時的なエラーの後の再試行回数を設定します。`set_num_retries` の有効範囲は 1 ~ 60、デフォルト値は 3 です。

'set_max_ldapua_naptive_threads', *max_ldapua_native_threads*

LDAP 認証要求を処理するエンジンで同時に実行できるネイティブ・スレッドの最大数を設定します。

`set_max_ldapua_native_threads` の最小値は 1 です。最大値は、`max native threads` の値から、`sp_configure` を使用して指定した `number of dump threads` の値を差し引いた値です。デフォルト値は、最大値と同じです。

`sp_configure` は、`max native threads` が、`set_max_ldapua_native_threads` と設定パラメータ `number of dump threads` の値に対して十分であるか確認します。

`set_max_ldapua_desc`, *max_ldapua_desc*

エンジンあたりの LDAP 記述子の最大数を設定します。

`set_max_ldapua_desc` の有効範囲は 1 ~ 20、デフォルト値は 20 です。

`set_abandon_ldapua_when_full`, {true | false}

エンジンあたりの最大ネイティブ・スレッド数を超えた場合の LDAP ユーザ認証の代替方法を設定できます。

`set_abandon_ldapua_when_full` が true に設定されている状態で、使用可能なスレッドがない場合は、要求が放棄されます。`enable ldap user auth` が 1 に設定されている場合、クライアントは Adaptive Server の `syslogins` を使用して認証されます。`enable ldap user auth` が 2 に設定されている場合、クライアントのログインは失敗します。

`set_abandon_ldapua_when_full` が false に設定されている場合は、LDAP 記述子が新しい認証要求を受け取ることができるようになるまで認証要求はブロックされます。

help

`sp_ldapadmin` の使用方法を表示します。

reinit_descriptors

確立されているすべての LDAP サーバ記述子のバインドを解除し、LDAP ユーザ認証サブシステムを再初期化します。構文は次のとおりです。

```
sp_ldapadmin 'reinit_descriptors'
```

認証局により信頼されたルート・ファイルが変更された場合、システム・セキュリティ担当者は **reinit descriptors** を使用して LDAP ユーザ認証を再初期化する必要があります。詳細については、『リファレンス・マニュアル：プロシージャ』の **sp_ldapadmin** を参照してください。

set_log_interval, log_interval

エラー・メッセージのログ間隔を分単位で設定します。

set_log_interval の有効範囲は 0 ~ 480、デフォルト値は 3 です。

例 **例 1** LDAP SunONE Directory Server 用の LDAP URL 検索文字列を作成します。

```
sp_ldapadmin set_primary_url, 'ldap://voyager:389/  
ou=People,dc=MyCompany,dc=com??sub?uid=*'
```

検索文字列が特定するのは、ホスト名 “voyager” で受信しているディレクトリ・サーバ、ポート番号 389 (デフォルトの LDAP プロトコル・ポート)、検索を開始するベース・ノードが組織単位 (ou) “People” 内にあること、ドメインが “MyCompany.com” であることです。この検索文字列は、フィルタ **uid=*** に一致するすべての属性を返します。Adaptive Server は、ワイルドカードを認証対象となる Adaptive Server のログイン名に置換します。

例 2 例 1 で記述した基準を使用して OpenLDAP 2.0.25 で定義された、LDAP URL 検索文字列を作成します。

```
sp_ldapadmin set_primary_url, 'ldap://voyager:389/  
dc=MyCompany,dc=com??sub?cn=*'
```

例 3 セカンダリ LDAP URL 検索文字列を null に設定します。このため、フェールオーバーとセカンダリ LDAP サーバは指定されません。

```
sp_ldapadmin set_secondary_url, null
```

例 4 複合フィルタを使用して LDAP URL 検索文字列を作成します。

```
sp_ldapadmin set_primary_url, 'ldap://voyager:389/  
ou=people,dc=siroe,dc=com??sub?(&(uid=*) (ou=accounting))'
```

例 5 Windows 2000 コントローラのデフォルトの Microsoft Active Directory スキーマを使用します。

```

1> sp_ldapadmin set_access_acct, 'cn=aseadmin, cn=Users, dc=mycompany,
    dc=com', 'aseadmin secret password'
2> go

1> sp_ldapadmin set_dn_lookup_url,
    'ldap://mydomainhostname:389/cn=Users,dc=mycompany,dc=com?
    distinguishedName?sub?samaccountname=*'
2> go

1> sp_ldapadmin set_primary_url,'ldap://mydomainhostname:389/'
2> go

```

“aseadmin”というユーザ名が Active Directory サーバに追加され、ユーザが検出されたツリーとオブジェクトに対する読み込みアクセスが付与されています。このユーザを認証するために、`distinguishedName` で指定された LDAP 属性が取得され、使用されます。フィルタで属性 `samaccountname=*` の検索が指定されています。ワイルドカード (*) は Adaptive Server ログイン・レコードの名前で置き換えられます。

たとえば、“`samaccountname=jqpublic`”は、“`cn=John Q. Public, cn=Users,dc=mycompany, dc=com`”という値の DN 属性 “`distinguishedName`”を Adaptive Server に返します。Adaptive Server はこの文字列を使用して `ldap://mydomainhostname:389` にバインドします。バインドが成功すると、認証が成功します。

例 6 ネイティブ・スレッドの最大値を 12 に設定します。

```
sp_ldapadmin 'set_max_ldapua_native_threads', '12'
```

例 7 認証要求が放棄されるまで Adaptive Server が LDAP サーバからの応答を待機する時間を 25,000 ミリ秒に設定します。

```
sp_ldapadmin, 'set_timeout', '25000'
```

例 8 LDAP 記述子が新しい認証要求を受け取ることができるまで認証要求を無効にします。

```
sp_ldapadmin 'set_abandon_ldapua_when_full', 'false'
```

例 9 現在の LDAP 値を表示します。

```

sp_ldapadmin

Primary:
URL:                'ldap://linuxpuneeng1:50917/'
DN Lookup URL:
'ldap://linuxpuneeng1:50917/dc=sybase,dc=com??sub?uid=*'
Access Account:    'cn=Directory Manager'
Active:            'TRUE'
Status:            'READY'

```

```
Secondary:
URL:          ''
DN Lookup URL: ''
Access Account: ''
Active:       'FALSE'
Status:       'NOT SET'
Timeout value: '5000' milliseconds
Log interval: '1' minutes
Number of retries: '3'
Maximum LDAPUA native threads per Engine:'400'
Maximum LDAPUA descriptors per Engine:'3'
Abandon LDAP user authentication when full:'false'

(return status = 0)
```

使用法

- LDAP ベンダが検索文字列の構文を決めています。どの場合でも、検索文字列はユーザをユニークに特定する属性名を指定します。形式は「属性= ワイルドカード」で、たとえば “cn=” のようになります。
- 複合フィルタの最初の属性では、RDN (相対識別名) を定義する必要があります。たとえば、“...sub?(uid=*)(ou=group)” のようになります。そうでない場合、認証に失敗します。
- 検索文字列が追加されると、Adaptive Server は、検索文字列中に有効な LDAP URL 構文が使用され、実在のノードを参照していることを確認します。有効な文字列が確実に予測した値を返すようにするには、Adaptive Server の設定時に検索文字列を慎重に選択し確認します。
- セカンダリ URL 検索文字列は、別の LDAP サーバへのフェールオーバーを有効にします。LDAP サーバが非アクティブになるか、検索文字列が無効にならない限り、Adaptive Server は、プライマリ URL 検索文字列を使用します。LDAP サーバが非アクティブになったり、検索文字列が無効になったりした場合、Adaptive Server は、認証にセカンダリ URL 検索文字列を使用します。
- 検索 DN アルゴリズムのログイン手順では、Adaptive Server で、検索を実行する前にアクセス・アカウントを使用して LDAP サーバにバインドする必要があります。Adaptive Server は、バインドの結果として LDAP 記述子 (ハンドル) を取得します。この記述子は、LDAP サーバでログインの DN を検索するために使用されます。

パーミッション `sp_ldapadmin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage security configuration</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sso_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

sp_listener

説明	任意のポートでの Adaptive Server のリスナを、サーバ単位で動的に起動したり停止したりします。
プロセス・モードの考慮事項	プロセス・モードで実行された場合、sp_listener は、任意のポートでの Adaptive Server のリスナを、エンジン単位で動的に起動したり停止したりします。
構文	スレッド・モードでは、構文は次のとおりです。 <pre>sp_listener "command", "server_name network"</pre> <p>または</p> <pre>sp_listener "command", "[protocol:]machine:port:CN=common_name"</pre> <p>プロセス・モードでは、構文は次のとおりです。</p> <pre>sp_listener "command", "server_name network", engine remaining</pre> <p>または</p> <pre>sp_listener "command", "[protocol:]machine:port:CN=common_name", engine</pre>
パラメータ	<p><i>command</i></p> <p>次のいずれかになります。</p> <ul style="list-style-type: none"> • start — 指定の各サーバで、指定ポートのリスナを起動します。 • stop — 指定リスナを停止します。 • suspend — リスナがこれ以上、接続を受け入れるのを防ぎます。 • resume — 中断状態のリスナに、受信を再開するよう指示します。 • status — パラメータにより指定されたリスナの状態に関してレポートします。状態は、active (アクティブ)、stopped (停止)、または suspended (中止) のいずれかです。 • help — sp_listener 構文を表示します。 <p><i>server_name network</i></p> <p>インタフェース・ファイルで指定された Adaptive Server の名前、またはネットワークの名前です。</p>

engine

(プロセス・モードでのみ使用) このコマンドの影響を受けるエンジン番号を指定します。**engine** は、引用符で囲まれた単一エンジン番号 (“2”)、リスト (“3,5,6”)、範囲 (“2-5”)、またはそれらの混合 (“2,3-5,7”) のいずれかを指定できます。

注意 **engine** パラメータは Windows NT では無視されます。

remaining

コマンドを実際に適用できるエンジン (つまりコマンドを適用できる状態のリスナがあるエンジン) すべてに対してコマンドを適用することを指定します。

protocol

プロトコルのタイプです。tcp、tli、ssltcp、ssltli、winsock、sslnlwsock、sslwinsock のいずれかになります。

machine:port

リスナ接続先のマシン名とポート番号です (インタフェース・ファイルで指定)。

CN=common_name

SSL 証明書の共有名を指定します。

プロトコルとして ssltcp を指定する場合にのみ、CN=common_name を使用します。Adaptive Server は、SSL 証明書の common_name に照らして指定する common_name を検証します。CN=common_name を含めない場合、Adaptive Server は server_name を使用して SSL 証明書の共通名に照らして検証します。証明書に完全修飾ドメイン名を含める場合、このドメイン名は CN=common_name と一致する必要があります。

例

例 1 サーバ orion に対応するインタフェース・ファイル内の各 master エントリに対して、リスナを起動します。

```
sp_listener "start", "orion"
```

例 2 ポート番号 4226 の TCP リスナを作成します。

```
sp_listener "start", "goldie:4226"
```

例 3 サーバ orion に対して、インタフェース・ファイル内のすべての master エントリのリスナを作成します。

```
sp_listener "start", "orion", "remaining"
```

例 4 ポート 4226 をまだ受信していないすべてのエンジンに対して、マシン goldie のポート 4226 の TCP リスナを起動します。

```
sp_listener "start", "goldie:4226", "remaining"
```

例 5 ポート番号 4226 の TCP リスナを停止します。

```
sp_listener "stop", "tcp:goldie:4226"
```

例 6 ポート番号 4226 のすべてのリスナを停止します。このコマンドには、**remaining** パラメータが含まれているので、ポート 4226 を受信していないエンジンがあっても、このコマンドは失敗しません。

```
sp_listener "stop", "tcp:goldie:4226", "remaining"
```

例 7 ポート 4226 の NT Winsock リスナを中断します。

```
sp_listener "suspend", "winsock:clouds:4226"
```

例 8 ポート 4226 のすべてのアクティブなリスナを再開します。

```
sp_listener "resume", "tcp:goldie:4226", "remaining"
```

例 9 共通名 `ase1.big server 1.com` を指定するには、次のように入力します。

```
sp_listener 'start','ssltcp:blad1:17251:
"CN=ase1.big server 1.com",'0'
```

使用法

- `sp_listener` では、前述の構文の項で説明した、2 とおりの構文のどちらかを使用します。最初の構文は、インタフェース・ファイルにリストされている、すべての **Adaptive Server master** ポートに影響を及ぼします。2 番目の構文では、インタフェース・ファイルにリストされていないリスナを管理できます。
- 属性名 “CN” は大文字と小文字を区別しません (“CN”、“cn”、または “Cn” を使用できます) が、共通名の属性値は大文字と小文字を区別します。
- `sp_listener` は、スレッド・モードで実行中に *engine* パラメータを含んでいる場合、そのパラメータを無視します。
- `sp_listener` のセマンティクスは、アトミックです。コマンドを完了できない場合、そのコマンドはアボートされます。
- `sp_listener` コマンドの `host` 要素が IPv6 アドレスである場合は、カッコで囲む必要があります。次に例を示します。

```
tcp:[2001:ec8:4008:1::123]:80
```
- `status` パラメータは、単独で発行できます。`status` パラメータは、インタフェース・ファイル内のすべてのリスナの状態を表示します。

- リスナの状態は、停止、中断、アクティブのいずれかです。sp_listener を使用すると、リスナをこれらの状態間で遷移できます。許可できない状態への遷移要求は、失敗します(存在しないリスナの停止要求など)。リスナの状態を判断するには、sp_listener “status” を使用します。
- remaining パラメータは、実行中のコマンド (start、stop、resume など) のために、変更 (たとえば、起動から停止状態への遷移) を許可する状態にあるすべてのリスナに対して、そのコマンドを正常に実行することを指定します。たとえば、エンジン 1 ~ 6 でリスナを起動しようとしたときにエンジン 1、4、5 が使用不可である場合、sp_listener... “remaining” はオフラインのエンジンを無視して、エンジン 2、3、6 でリスナを起動します。remaining パラメータを含めた場合、エンジン番号は指定できません。
- リスナの最大数は、32 です。インタフェース・ファイルで、master ポートを 2 つ持つ Adaptive Server を作成すると、他のポートでは、最大 30 個までしかリスナを起動できません。最初のリスナとは別に、各補助リスナが、ユーザ接続のようなリソースを消費します。したがって、25 個のユーザ接続が設定されている環境では、起動時に 3 個のリスナ (3 個の master エントリに対応) を起動すると、残っているユーザ接続件数は 30 となります。

パーミッション

sp_listener のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage server 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> Roles — 現在のアクティブな役割 Keywords or options — NULL Previous value — NULL Current value — NULL Other information — すべての入力パラメータ Proxy information — set proxy が有効な場合は元のログイン名

参照

sp_listener の IPV6 に関連する制限については、『セキュリティ管理ガイド』の「データの機密保持」に関する章にある「共通名の指定」を参照してください。

sp_listsuspect_db

説明 リカバリ時に検出された破壊が原因で現在オフラインになっているページがある、すべてのデータベースをリストします。

構文 sp_listsuspect_db

パラメータ なし

例 suspect (疑わしい) ページのあるデータベースをリストします。

```
sp_listsuspect_db
```

使用法

- sp_listsuspect_db は、データベース名、suspect ページの数、suspect ページを含むオブジェクトの数をリストします。
- suspect ページを特定するには、sp_listsuspect_page を使用します。

パーミッション sp_listsuspect_db は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_listsuspect_page](#)、[sp_setsuspect_granularity](#)、[sp_setsuspect_threshold](#)

sp_listsuspect_object

説明	リカバリ時に検出された破壊が原因で現在オフラインになっている (データベース内にある) すべてのインデックスをリストします。
構文	<code>sp_listsuspect_object [dbname]</code>
パラメータ	<i>dbname</i> データベースの名前を指定します。
例	<p>例 1 現在のデータベースにある <code>suspect</code> (疑わしい) インデックスをリストします。</p> <pre>sp_listsuspect_object</pre> <p>例 2 <code>pubs2</code> データベースにある <code>suspect</code> (疑わしい) インデックスをリストします。</p> <pre>sp_listsuspect_object pubs2</pre>
使用法	<ul style="list-style-type: none"> • データオンリー・ロック・テーブルのインデックスに <code>suspect</code> ページがある場合は、リカバリ中のインデックス全体がオフラインになります。オフラインのインデックスは、クエリ・オブティマイザの対象になりません。 • 修復するためにオフラインのインデックスをオンラインにするには、システム・プロシージャ <code>sp_forceonline_object</code> を使用します。 • リカバリ中に、すべてのページ・ロック・テーブルのインデックスが完全にオフラインになるわけではありません。これらのインデックスの個々のページだけがオフラインになります。これらのページは、<code>sp_forceonline_object</code> によってオンラインにできます。 • <code>sp_listsuspect_object</code> は、指定されたデータベース内にある、または <i>dbname</i> が省略されていれば現在のユーザ・データベース内にある、<code>suspect</code> インデックスすべてのデータベース名、オブジェクト ID、オブジェクト名、インデックス ID、アクセス・ステータスをリストします。 • <code>access</code> カラムの <code>SA_ONLY</code> の値は、インデックスが、システム管理者だけが使用できるオンライン状態に変更されていることを示します。<code>BLOCK_ALL</code> の値は、インデックスがすべてのユーザに対してオフラインであることを示します。
パーミッション	<code>sp_listsuspect_object</code> は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの 実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 **マニュアル** リカバリ・フォールト・アイソレーションの詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_forceonline_object](#)

sp_listsuspect_page

説明	リカバリ時に検出された破壊が原因で現在オフラインになっている、データベース内のすべてのページをリストします。
構文	<code>sp_listsuspect_page [dbname]</code>
パラメータ	<i>dbname</i> データベースの名前を指定します。
例	<p>例 1 現在のデータベースにある <code>suspect</code> (疑わしい) ページをリストします。</p> <pre>sp_listsuspect_page</pre> <p>例 2 <code>pubs2</code> データベースにある <code>suspect</code> (疑わしい) ページをリストします。</p> <pre>sp_listsuspect_page pubs2</pre>
使用法	<ul style="list-style-type: none"> <code>sp_listsuspect_page</code> は、指定されたデータベース内にある、データベース名、ページ ID、オブジェクト、インデックス ID、アクセス・ステータスをリストします。<i>dbname</i> が省略されている場合は、現在のユーザ・データベース内です。 “access” カラムの <code>SA_ONLY</code> の値は、ページがシステム管理者だけが使用できるオンライン状態に変更されていることを示します。<code>BLOCK_ALL</code> の値は、ページがすべてのユーザに対してオフラインであることを示します。
パーミッション	<code>sp_listsuspect_page</code> は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [システム・プロシージャ `sp_listsuspect_db`](#)、[`sp_setsuspect_granularity`](#)、[`sp_setsuspect_threshold`](#)

sp_lmconfig

説明 Adaptive Server のライセンス管理に関する情報を設定します。

構文

```
sp_lmconfig
    [ 'edition' [, edition_type] ]
    [ 'license type' [, license_type_name] ]
    [ 'smtp host' [, smtp_host_name] ]
    [ 'smtp port' [, smtp_port_number] ]
    [ 'email sender' [, sender_email_address] ]
    [ 'email recipients' [, email_recipients] ]
    [ 'email severity' [, email_severity] ]
    ]
```

パラメータ sp_lmconfig
パラメータを指定しない場合、以下のライセンス・ステータス情報が表示されます。

- Server Name
- ライセンス名
- バージョン
- 数量ステータス
- 有効期限

edition

ライセンス・エディションを表す静的設定パラメータです。

edition_type

エディションの種類を表します。次の値のいずれかです。

- null — デフォルト値。null 値を指定した場合、製品のエディションは設定されておらず、Adaptive Server は任意のエディション用のライセンスで起動する。
- EE — Enterprise Edition。
- SE — Small Business Edition。
- DE — Developer's Edition。
- XE — Express Edition。

license type

Adaptive Server のインストール環境のライセンスの種類を示す静的設定パラメータです。edition に null 以外を指定した場合にのみ有効です。

license_type_name

Adaptive Server の特定のインストールのライセンスの種類を指定します。Developer's Edition (DE) または Express Edition (XE) を使用している場合は、*license type* を指定する必要はありません。一般的な有効値は次のとおりです。

- SRST — ネットワーク・シート付きサーバ・ライセンス
- SVST — ネットワーク・シート付きスタンバイ・サーバ・ライセンス
- SRCU — 同時ユーザ・シート付きサーバ・ライセンス
- SVCU — 同時ユーザ・シート付きスタンバイ・サーバ・ライセンス
- SRIA — インターネット・アクセス・ライセンス付きサーバ・ライセンス
- SVIA — インターネット・アクセス・ライセンス付きスタンバイ・サーバ・ライセンス
- CP — CPU ライセンス
- SF — スタンバイ CPU ライセンス
- null — デフォルト

注意 *sp_lmconfig* では、このリストに加えて、2 文字の特別なレガシー・ライセンスも指定できます。このライセンスの種類が受け入れられない場合は、種類を *null* に設定し、ネットワーク・ライセンス・サーバ・オプション・ファイルを使用して、この Adaptive Server が使用するライセンスを制御します。

smtp host, smtp host name

ライセンス・イベント通知用の電子メールの送信に使用する SMTP ホストを指定します。

smtp port, smtp port number

ライセンス・イベント通知用の電子メールの送信に使用する SMTP ポートを指定します。

email sender, sender email address

ライセンス・イベントの電子メール通知の送信者のアドレスとして使用する電子メール・アドレスを指定します。

email recipients, *email recipients*

ライセンス・イベントの電子メール通知を受け取る電子メールの受信者のカンマ区切りのリストです。

email severity, *email severity*

どの重大度以上のエラーについて電子メール通知を送信するかを指定します。デフォルトはエラーで、その他に警告と情報を指定できます。

例

システムの基本的なライセンス設定情報を表示します。

```
1>sp_lmconfig
2> go
```

Parameter Name	Config Value
edition	EE
license type	CP
smtp host	null
email recipients	null
email severity	null
smtp port	null
email sender	null

License Name	Version	Quantity	Status	Expiry Date	Server Name
ASE_HA	2010.03314	2	expirable	Apr 1 2010 12:00AM	cuprum
ASE_ASM	null	0	not used	null	null
ASE_EJB	null	0	not used	null	null
ASE_EFTS	null	0	not used	null	null
ASE_DIRS	null	0	not used	null	null
ASE_XRAY	null	0	not used	null	null
ASE_ENCRYPTION	null	0	not used	null	null
ASE_CORE	2010.03314	2	expirable	Apr 1 2010 12:00AM	cuprum
ASE_PARTITIONS	null	0	not used	null	null
ASE_RLAC	null	0	not used	null	null
ASE_MESSAGING_TIBJMS	null	0	not used	null	null
ASE_MESSAGING_IBMMQ	null	0	not used	null	null
ASE_MESSAGING_EASJMS	null	0	not used	null	null

Property Name	Property Value
PE	EE
LT	CP
ME	null
MC	null
MS	null
MM	null
CP	0
AS	A

```
(return status = 0)
```

- 使用法**
- パラメータを 1 つも指定しないと、`sp_lmconfig` はライセンスがチェックアウトされる場所のサーバ名も表示します。
 - エディションを指定しない場合や “null” を使用した場合、`Adaptive Server` が起動時にライセンス・エディションを検索し、検出されたエディションを使用します。
 - `sp_lmconfig` で設定する設定オプションは、`syloapi` プロパティ・ファイルに格納されます。

パーミッション `sp_lmconfig` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server configuration</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 マニュアル 使用しているプラットフォームの『`Adaptive Server` インストール・ガイド』

sp_lock

説明 現在ロックを保持しているプロセスのオブジェクト名と ID をレポートします。

構文 sp_lock [spid1[, spid2]] | [@verbose = int]

パラメータ

spid1

master.dbo.sysprocesses テーブルが提供する Adaptive Server プロセス ID 番号です。sp_who を実行して、ロック・プロセスの *spid* を取得します。

spid2

ロックを確認するための、別の Adaptive Server プロセス ID 番号です。

@verbose = int

spid の代わりに test..testa のようにテーブル名を連結した名前を *spid* ごとに表示します。

注意 sp_lock は @verbose の値が null かどうかを確認するだけであるため、int には任意の数を指定できます。

例 **例 1** この例は、spid 7、18、23 のシリアル・プロセスのロック・ステータスと 2 つのプロセス・ファミリを示しています。fid 1 のファミリには、spid 1 のコーディネート・プロセスと spid 8、9、10 のワーカー・プロセスがあります。fid 11 のファミリには、spid 11 のコーディネート・プロセスと spid 12、13、14 のワーカー・プロセスがあります。

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

```

例2 現在 spid 7 が保持しているロックについての情報を表示します。

```
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

```

例3 最初に、現在ロックを保持している実行プロセスの ID (1056003762) について pubs2 データベースに問い合わせ、次に @verbose オプションを使用して pubs2 データベースに問い合わせます。この問い合わせでは、プロセス ID の他にオブジェクト名 (master..spt_values) も返されます。

```

1> use pubs2
2> go
1> sp_lock
2> go

```

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 loid locktype table_id page row dbname class context
-----
0 15 30 Sh_intent 1056003762 0 0 master Non Cursor Lock

```

```

(1 row affected)
(return status = 0)

```

```
1> sp_lock @verbose=0
2> go
```

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	loid	locktype	page	row	objectName	id	class	context
0	15	30	Sh_intent	0	0	master..spt_values	1056003762	Non	Cursor Lock

```
(1 row affected)
(return status = 0)
```

使用法

- パラメータなしで `sp_lock` を使用すると、現在ロックを保持しているすべてのプロセスについての情報を表示します。
- `select` 文内で `holdlock` キーワードを使用した場合だけ、ユーザがロックを制御できます。
- テーブルの ID 番号からテーブル名を検索するには、`object_name` システム関数を使用します。
- Cluster Edition の 15.0.3 以前のバージョンの `sp_lock` は、ストアド・プロシージャを発行したインスタンスに関連付けられたロックに関する情報のみを表示していました。Cluster Edition バージョン 15.0.3 以降の `sp_lock` は、クラスタ内のすべてのロックに関する情報を表示します。
- `sp_lock` は `fid` の順に並べて出力し、次に `spid` の順に並べて出力します。
- `loid` カラムには、ブロック・トランザクションのユニークなロック所有者 ID が示されます。`loid` 値が偶数になっていることは、ローカル・トランザクションがロックを所有していることを示します。奇数値は、外部トランザクションがロックを所有していることを示します。
- `locktype` カラムでは、ロックの種類が共有ロック ("Sh" プレフィックス)、排他ロック ("Ex" プレフィックス)、更新ロックであるかどうかを示し、そのロックが保持されている場所がテーブル上 ("table" または "intent")、ページ ("page") 上であるかどうかを示します。

“locktype” カラム内に “blk” サフィックスが表示されている場合は、このプロセスが、ロックを取得する必要がある他のプロセスをブロックしていることを示しています。このプロセスが完了すると、すぐに他のプロセス (1 つまたは複数) が動作を開始します。“locktype” カラム内に “demand” サフィックスがあれば、プロセスが排他ロックを取得しようとしていることを示します。ロック・タイプの詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。

- **class** カラムは、ロックがカーソルと対応しているかどうかを示します。次のうちの 1 つが表示されます。
 - “Non Cursor Lock” は、ロックがカーソルに対応していないことを表します。
 - “Cursor Id number” が表示されている場合は、ロックが、その Adaptive Server プロセス ID のカーソル ID 番号と対応していることを示します。
 - カーソル名が表示されている場合は、ロックが、sp_lock を実行している現在のユーザが所有している cursor_name カーソルと対応していることを示します。
- **fid** カラムでは、ロックが属するファミリー (コーディネーティング・プロセスとそのワーカー・プロセス) が識別されます。fid の値は次のとおりです。
 - ゼロの値 : spid で表わされているタスクが逐次実行されることを示します。このタスクは並列実行には関与しません。
 - ゼロ以外の値 : ロックを保持しているタスク (spid) が、文を並列実行しているプロセス (fid のファミリー ID 番号によって識別されているプロセス) のファミリー・メンバであることを示します。この値が spid と同じ場合、そのタスクがファミリー内でクエリを並列実行しているコーディネーティング・プロセスであることを示します。
- **context** カラムでは、ロックのコンテキストを識別します。同一のファミリー内のワーカー・プロセスには、同じコンテキスト値が割り当てられます。“context” では、次のような有効値が表示されます。
 - “NULL” は、このロックを保持しているタスクが、クエリを逐次実行するタスクなのか、それともトランザクション独立性レベル 1 で並列実行されているクエリなのかを示します。

- **Sync-pt duration request** は、ロックを保持しているタスクが、クエリが完了するまでロックを保持することを示します。
 ロックのコンテキスト値が **Sync-pt duration request** になる可能性があるのは、ロックが並列クエリの一部として保持されているテーブル・ロックである場合や、ロックがトランザクション独立性レベル3でワーカー・プロセスによって保持されている場合です。または、ロックが並列クエリ内でワーカー・プロセスによって保持されていて、トランザクション中にそのロックが保持される必要がある場合です。
- **“Ind pg”** は、インデックス・ページ (全ページロック・テーブルのみ) に対するロックを示します。
- **“Inf key”** は、無限キー・ロックを示します (特定の範囲のクエリがデータオンリー・ロック・テーブル上で、トランザクション独立性レベル3の場合)。
- **“Range”** は、範囲ロックを示します (特定の範囲のクエリがデータオンリー・ロック・テーブル上でトランザクション独立性レベル3の場合)。

これらの新しい値は、該当するものがあれば、(“Sync pt duration”と置き換わる) “Fam dur”との組み合わせ、および相互との組み合わせで示される場合があります。

- row カラムには、ローレベル・ロックのローの数が表示されます。
- sp_lock 出力には、次のロック・タイプも表示されます。
 - **“Sh_row”** : 共有ロー・ロック。
 - **“Update_row”** : 更新ロー・ロック。
 - **“Ex_row”** : 排他ロー・ロック。

パーミッション

sp_lock は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [kill](#)、[select](#)

システム・プロシージャ [sp_familylock](#)、[sp_who](#)

sp_locklogin

説明 ユーザがログインできないように Adaptive Server アカウントをロックするか、またはロックされたすべてのアカウントのリストを表示する。

構文 `sp_locklogin login | NULL | wildcard_string , "lock" | "unlock",
[except_login_name | except_role_name]
[, number_of_inactive_days]`

または

`sp_locklogin`

パラメータ

`sp_locklogin`

パラメータを指定しない場合、ロックされたログインがすべて表示されます。

loginame

ロックまたはロック解除されるアカウントの名前です。

wildcard_string

一連のログインを識別する、ワイルドカードを使用した文字列です。

NULL

`sa_role` のすべてのログインがロックされています。

`lock | unlock`

アカウントをロックまたはロック解除するかどうかを指定します。

except_login_name

ロックの対象から除外するログインの名前です。

except_role_name

ロックの対象から除外する役割の名前です。たとえば、ある役割のすべてのログインを除外できます。

number_of_inactive_days

アカウントが非アクティブになっている 1 ~ 32,767 の日数。

例 **例 1** ユーザ “charles” のログイン・アカウントをロックします。

```
sp_locklogin charles, "lock"
```

例 2 `sa_role` 以外のすべてのログインをロックします。

```
sp_locklogin NULL, "lock", sa_role
```

例 3 ロックされているすべてのアカウントのリストを表示します。

```
sp_locklogin
```

例 4 過去 60 日間に認証を受けなかったすべてのログイン・アカウントをロックします。

```
sp_locklogin NULL, 'lock', NULL, 60
```

注意 sp_passwordpolicy オプション “enable last login updates” を “0” に設定した場合、このコマンドは無効になります。

使用法

- パラメータを指定しないで `sp_locklogin` を実行すると、ロックされているすべてのログインが表示されます。
- `syslogins` カラム `lockdate`、`locksuid`、`lockreason` は、ログインのロック/ロック解除時に更新されます。
- `sp_locklogin` を使用する条件は次のとおりです。
 - 除外する対象の指定では、ワイルド・カードは使用できません。
 - 既存の機能は影響を受けません。
 - 指定した除外の対象は、まずログインに対してマッチングされます。該当するログインが存在しない場合、次に役割に対してチェックされます。
 - ログインに値 `NULL` を指定した場合、すべてのログインを表します。
 - 指定したログイン名または除外の対象が存在しない場合、エラーが表示されます。
 - 指定したログインのセットが実質的に空の場合、何も行われません。
 - 除外の対象が `NULL` の場合、`login` パラメータで指定した一連のログインがロックされます。
 - Sybase 高可用性フェールオーバーのみ — Adaptive Server 15.0 より前のバージョンの `sp_locklogin` では、ロックまたはロック解除する対象のログインがリモートの高可用性サーバに存在するかどうかをチェックされました。その際には、そのログインの `suid` (サーバ・ユーザ ID) がそのサーバに存在するかどうかを調べるという方法が使用されていました。

Adaptive Server version 15.0 の `sp_locklogin` では、`suid` とログイン名の両方がチェックされます。

- `lock` と `unlock` 以外の単語を指定すると、エラーが表示されます。

パーミッション

`sp_locklogin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any login</code> 権限を持つユーザーであることが必要。 <code>maxfailedlogin</code> が原因でロックされたログイン・アカウントのロックを解除するには、 <code>change password</code> 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sso_role</code> が付与されたユーザーであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド `create login`、`alter login`

sp_logdevice

説明 データと同じデバイス上にあるデータベースのトランザクション・ログを、別のデータベース・デバイスに移動させます。

構文 `sp_logdevice dbname, devname`

パラメータ

dbname

`syslogs` テーブルを格納しているデータベースの名前です。トランザクション・ログを収納しているこのテーブルは、指定の論理デバイスに配置されます。

devname

`syslogs` テーブルの配置先デバイスの論理名です。このデバイスは、(`create database` または `alter database` で指名された)データベースに対応するデータベース・デバイスでなければなりません。データベースのデバイスに関するレポートを表示させるには、`sp_helpdb` を実行します。

例 **例 1** `products` データベースを作成し、`logs` データベース・デバイスに `products.syslogs` テーブルを配置します。

```
create database products on default = "10M", logs = "2M"
go
sp_logdevice products, logs
go
```

例 2 同じデバイス上にログとデータを持つ `test` データベースの場合、`test` 用のログを、`logdev` ログ・デバイスに配置します。

```
alter database test log on logdev
go
sp_logdevice test, logdev
go
```

使用法

- `sp_logdevice` は、シングルユーザ・モードでのみ実行できます。
- `sp_logdevice` プロシージャは、`syslogs` に今後割り付けられる領域だけに影響します。使用しているログの最初のページが、同じデバイス上にユーザのデータとして配置されたままになっていると、その間、動作の信頼性が低くなる脆弱性が発生します。したがって、別のデバイスにトランザクション・ログを配置する場合は、`create database` に `log on` オプションを使用する方法をおすすめします。こうすると、別のデバイスにトランザクション・ログ全体を最初から配置できます。

- リカバリとパフォーマンス上の理由から、トランザクション・ログを別のデータベース・デバイスに入れてください。

非常に小規模で、重要ではないデータベースの場合には、ログをデータベースの他の部分と一緒に保持できます。このようなデータベースでは、`dump database` を使用してデータベースとログをバックアップし、`dump transaction with truncate_only` を使用してログをトランケートします。

- `dbcc checkalloc` と `sp_helplog` は、次の `dump transaction` が終了するまでは、元のデータベース・デバイス上に割り当てられている `syslogs` の一部のページを表示します。その後、`sp_logdevice` を実行したときに指定したデバイスに、トランザクション・ログが完全に移行します。
- トランザクション・ログに必要なデバイスのサイズは、更新処理の量やトランザクション・ログ・ダンプの頻度によって異なります。データベース自体に割り当てる領域の 10% ~ 25% を目安としてログ・デバイスに割り当てます。
- `sp_logdevice` は、同じデバイス上にログとデータを持っているデータベースだけに使用します。ログとデータを別々のデバイスに持つデータベースに対しては、`sp_logdevice` を使用しないでください。
- トランザクション・ログに割り付けられている記憶領域を増やすには、`alter database` を使用します。トランザクション・ログを別のデバイスに配置するために、`log on` オプションを付けてコマンド `create database` を実行した場合は、この構文を使用してログ・セグメントのサイズを大きくします。`log on` を使用しなかった場合は、`sp_logdevice` を実行します。

```
sp_extendsegment segname, devname
```

`syslogs` を配置するデバイスまたはセグメントは、`syslogs` だけに使用されます。データベースの残りの部分に割り当てられている記憶領域を増やすには、`alter database` を実行するときに、そのログ・デバイス以外のデバイスを指定します。

- `disk init` を使用して、データベースまたはトランザクション・ログ用の新しいデータベース・デバイスを初期化します。

パーミッション

`sp_logdevice` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データベース所有者か <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 マニュアル『システム管理ガイド』

コマンド [alter database](#)、[create database](#)、[dbcc](#)、[disk init](#)、[dump database](#)、[dump transaction](#)、[select](#)

システム・プロシージャ [sp_extendsegment](#)、[sp_helpdevice](#)、[sp_helplog](#)

sp_loginconfig

説明 (Windows NT のみ) 1 つまたはすべての統合セキュリティ・パラメータの値を表示します。

構文 sp_loginconfig ["parameter_name"]

パラメータ

parameter_name

調べたい統合セキュリティ・パラメータの名前です。値は次のとおりです。

- login mode
- default account
- default domain
- set host
- key _
- key \$
- key @
- key #

例 **例 1** すべての統合セキュリティ・パラメータの値を表示します。

```
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 #               -
```

例 2 login mode セキュリティ・パラメータの値を表示します。

```
sp_loginconfig "login mode"

name                config_item
-----
login mode          standard
```


- 使用法
- 統合セキュリティ・パラメータの値は、Windows NT レジストリに格納されます。パラメータの変更方法については、『Adaptive Server Enterprise 設定ガイド Windows NT 版』のログイン・セキュリティに関する章を参照してください。
 - `sp_loginconfig` は、Adaptive Server の起動時に有効であった `config_item` の値を表示します。Adaptive Server を起動してからレジストリ値を変更した場合、その変更内容は `sp_loginconfig` 出力には反映されません。

パーミッション `sp_loginconfig` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any login</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_revokelogin](#)

sp_logininfo

説明 (Windows のみ) [sp_grantlogin](#) によって Windows NT のユーザとグループに付与されているすべての役割を表示します。

構文 sp_logininfo ["login_name" | "group_name"]

パラメータ

login_name
Windows NT ユーザのネットワーク・ログイン名です。

group_name
Windows NT グループ名です。

例 **例 1** Windows NT のユーザ “regularjoe” に付与されているパーミッションを表示します。

```
sp_logininfo regularjoe

account name      mapped login name  type              privilege
-----
HAZE¥regularjoe  HAZE_regularjoe   user              'oper_role'
```

例 2 [sp_grantlogin](#) によって Windows NT のユーザとグループに付与されているすべてのパーミッションを表示します。

```
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'
```

使用法

- sp_logininfo は、[sp_grantlogin](#) によって Windows NT のユーザとグループに付与されているすべての役割を表示します。
- Windows NT のユーザ名またはグループ名を指定する場合、ドメイン名とドメイン・セパレータ (¥) は省略できます。

パーミッション sp_logininfo のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage roles 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [grant](#)、[setuser](#)

システム・プロシージャ [sp_displaylogin](#)、[sp_grantlogin](#)、[sp_revokelogin](#)、[sp_role](#)、[sp_who](#)

sp_logiosize

説明 現在のデータベースのトランザクション・ログに対して I/O を行っているときに、Adaptive Server が使用するログ I/O サイズを、異なるメモリ・プールに変更します。

構文 sp_logiosize ["default" | "size" | "all"]

パラメータ

default

キャッシュ内で 2 ページ・サイズの論理ページのメモリ・プールが使用可能であれば、現在のデータベースに使用されるログ I/O サイズを、Adaptive Server のデフォルト値 (2 ページ・サイズの論理ページ) に設定します。使用できない場合、Adaptive Server はログ I/O サイズを 1 ページ・サイズの論理ページに設定します。default はキーワードなので、このパラメータを指定するときには引用符が必要です。

size

現在のデータベースに対して設定するログ I/O サイズです。指定できる値は、論理ページ・サイズの倍数で、最大 4 倍までです。値は引用符で囲んでください。

all

キャッシュ名でグループ分けされているすべてのデータベースについて、設定されているログ I/O サイズを表示します。

例 **例 1** 現在のデータベースに対して設定されたログ I/O サイズを表示します。

```
sp_logiosize
```

```
The transaction log for database 'master' will use I/O size of 2 Kbytes.
```

例 2 現在のデータベースのログ I/O サイズを変更して、8K のメモリ・プールを使用します。データベースのトランザクション・ログが 8K のメモリ・プールを持たないキャッシュにバインドされている場合、Adaptive Server はエラー・メッセージを返します。このエラー・メッセージは、そのようなプールが存在しないことと、現在のログ I/O サイズは変更されていないことを示します。

```
sp_logiosize "8"
```

例 3 現在のデータベースのログ I/O サイズを、Adaptive Server のデフォルト値 (1 ページ・サイズの論理ページ) に変更します。トランザクション・ログが使用しているキャッシュに、その論理ページ・サイズのメモリ・プールが存在しない場合、Adaptive Server は 2K のメモリ・プールを使用します。

```
sp_logiosize "default"
```

例 4 すべてのデータベースに対して設定されたログ I/O サイズを表示します。

```
sp_logiosize "all"

Cache name:default data cache
Data base                               Log I/O Size
-----
master                                  2 Kb
tempdb                                  2 Kb
model                                    2 Kb
sybssystemprocs                          2 Kb
pubs3                                    2 Kb
pubtune                                  2 Kb
dbccdb                                    2 Kb
sybsyntax                                2 Kb
```

使用法

- `sp_logiosize` は、現在のデータベースに対するログ I/O サイズを変更します。すべてのユーザは、`sp_logiosize` を実行すると、現在のデータベース用に設定されたログ I/O サイズを表示できます。ログ I/O サイズを変更できるのは、システム管理者だけです。
- `sp_logiosize` にパラメータを指定しなければ、現在のデータベースのログ I/O サイズが表示されます。
- ログ I/O サイズを変更すると、ただちに有効になります。Adaptive Server は、そのデータベース用の新しい I/O サイズを、`sysattributes` テーブルに記録します。
- `sp_logiosize` 用に指定する値は、データベースのトランザクション・ログが使用するキャッシュに対して設定された既存のメモリ・プールと必ず対応させてください。これらのプールを指定するには、`sp_poolconfig` システム・プロシージャを使用します。

キャッシュ内で2ページ・サイズの論理ページのメモリ・プールが使用可能であれば、Adaptive Serverは、データベースのログ I/O サイズのデフォルト値を、2ページ・サイズの論理ページに設定します。使用できない場合は、ログ I/O サイズに1ページ・サイズの論理ページが設定されます(1ページ・サイズの論理ページのメモリ・プールは、どのキャッシュでも必ず存在します)。ほとんどの負荷状態で、ログ I/O サイズは1ページより2ページの論理ページ・サイズの方がより優れたパフォーマンスを得ることができます。したがって、トランザクション・ログが使用する各キャッシュは、2ページ・サイズの論理ページを割り当てるようにしてください。キャッシュとメモリ・プールの設定方法の詳細については、『システム管理ガイド』と『パフォーマンス&チューニング・ガイド』を参照してください。

- トランザクション・ログが **logonly** タイプのキャッシュにバインドされている場合、定義されたログ I/O サイズより大きな I/O サイズを持つ、そのキャッシュ内のメモリ・プールは使用されません。

たとえば、2K、4K、8K のメモリ・プールを持つ **logonly** タイプのキャッシュに、トランザクション・ログがバインドされているデータベースが、2K サーバ上に2つしかない場合、デフォルトでは、**sp_logiosize** は、これらのパラメータに対するログ I/O サイズを 4K、に設定しますが、8K のメモリ・プールは使用されません。したがって、キャッシュの領域を無駄にしないように、ログ I/O サイズの設定には注意してください。

- リカバリの間は、データベース用に設定されているログ I/O サイズには関係なく、デフォルト・キャッシュの論理ページ・サイズのメモリ・プールだけがアクティブになっています。トランザクション・ログは、デフォルト・キャッシュのこのプールに読み込まれます。ロールバックまたはロールフォワードされるすべてのトランザクションは、デフォルト・データ・キャッシュにデータ・ページを読み込みます。

パーミッション

すべてのユーザが、**sp_logiosize** を実行してログ I/O サイズの値を表示できます。

sp_logiosize の次のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 manage data cache 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 sa_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_cacheconfig](#)、[sp_poolconfig](#)

sp_logintrigger

説明 グローバル・ログイン・トリガを設定または表示します。このグローバル・ログイン・トリガの特性は、個人のログイン・スクリプトと同じです。これは、システム管理者やセキュリティ担当者を含め、ログインしようとしているすべてのユーザの個人のログイン・スクリプトよりも前に実行されます。

構文 `sp_logintrigger 'global login trigger name'`

パラメータ `global login trigger name`

グローバル・ログイン・トリガの名前です。

`sp_logintrigger` にパラメータを含めない場合、現在のログイン・トリガのステータスと名前が表示されます (存在する場合)。グローバル・ログイン・トリガが定義されていない場合、ローは表示されません。

例 **例 1** `sp_logintrigger` を使用してグローバル・ログイン・トリガを設定するには、次のように入力します。

```
sp_logintrigger 'master.dbo.myproc'
```

例 2 更新されたグローバル・ログイン・トリガを、次のように返します。

```
1sp_logintrigger
2> go
Global login trigger          Status
-----
sybssystemprocs.dbo.myproc    Enabled
(1 row affected)
(return status = 0)
```

例 3 グローバル・ログイン・トリガが存在しない場合は、次のように表示されます。

```
1sp_logintrigger
2> go
Global login trigger Status
-----
(0 rows affected)
```

例 4 `sp_logintrigger` で以前指定したグローバル・ログイン・トリガを削除するには、次のように入力します。

```
sp_logintrigger 'drop'
```

使用法

- グローバル変数 `@@logintrigger` で、グローバル・ログイン・トリガが定義されて有効になっているかどうかを確認します。

- このグローバル・ログインと個人のログイン・スクリプト間には違いがあります。このグローバル・ログイン・トリガは、**sysattributes** で名前別に保存されますが、個人のログイン・スクリプトはオブジェクト **ID** でのみ保存されます。

パーミッション

sp_logintrigger のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、新しいログイン・トリガを設定するには、 manage security configuration 権限を持つユーザであることが必要。 すべてのユーザは、 sp_logintrigger を実行して、現在のグローバル・ログイン・トリガを表示できます。
細密なパーミッションが無効	細密なパーミッションが無効の場合、新しいログイン・トリガを設定するには、 sso_role が付与されたユーザであることが必要。 すべてのユーザは、 sp_logintrigger を実行して、現在のグローバル・ログイン・トリガを表示できます。

sp_maplogin

説明	外部ユーザを Adaptive Server のログインにマップします。
構文	<pre>sp_maplogin (authentication_mech null), (client_username null), (action login_name null)</pre>
パラメータ	<p>authentication_mech sp_modifylogin の authenticate with オプションに指定するいずれかの有効な値です。</p> <p>client_username 外部ユーザ名。オペレーティング・システム名、LDAP サーバのユーザ名、または PAM ライブラリが認識できる任意の名前を指定できます。null 値を指定すると、すべてのログイン名が有効になります。</p> <p>action create login または drop を指定します。create login を使用すると、ログインが認証されると同時にログインが作成されます。drop はログインを削除するときに使用します。</p> <p>login_name syslogins にすでに存在する Adaptive Server ログインです。</p>
例	<p>例 1 外部ユーザ “jsmith” を Adaptive Server ユーザ “guest” にマップします。認証が行われると、“jsmith” は “guest” の権限を得ます。監査ログイン・レコードには、client_username と Adaptive Server ユーザ名の両方が表示されます。</p> <pre>sp_maplogin NULL, "jsmith", "guest"</pre> <p>例 2 PAM で認証されたすべての外部ユーザについて、ログインが存在しない場合は新規ログインを作成するように Adaptive Server に指示します。</p> <pre>sp_maplogin PAM, NULL, "create login"</pre>
使用法	sp_maplogin を使用すると、LDAP ディレクトリで定義されている “ase.open.user” のような外部名またはクライアント名を、Adaptive Server のログイン名 “aseopenuser” にマップできます。したがって、client_username は LDAP サーバの名前の規則に従い、login_name は Adaptive Server の識別子の規則に従います。

LDAP ユーザ認証を使用している場合、LDAP サーバでの名前が Adaptive Server のログイン名と異なるときは、`sp_maplogin` を使用して、LDAP サーバの認証には `client_username` を、Adaptive Server 内での ID には Adaptive Server `login_name` を使用するようにします。したがって、“`isql -U client_username...`” は、Adaptive Server 内では `login_name` という ID を持ちます。

`sp_helpmaplogin` を使用すると、`client_username` と `login_name` を確認できます。次に例を示します。

```

1> sp_helpmaplogin
2> go

authentication      client name          login name
-----
LDAP                  ase.open.user       aseopenuser

C:\¥> isql -Uase.open.user -Pasepass
1> select @@authmech
2> go

-----
ldap

```

パーミッション

`sp_maplogin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any login</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

sp_merge_dup_inline_default

説明 既存の重複インライン・デフォルト・オブジェクトを削除し、ユニークなインライン・デフォルトを共有可能なインライン・デフォルト・オブジェクトに変換する。

構文 sp_merge_dup_inline_default [@report_only = {yes | no} [, @show_progress = {yes | no}]]

パラメータ

@report_only

yes を指定すると、現在データベースのユニークなインライン・デフォルトの数をレポートしますが、変更はありません。no を指定すると、次のオプションがあります。

- sp_merge_dup_inline_default は、重複インライン・デフォルト・オブジェクトを削除し、すべてのユニークなインライン・デフォルトを共有可能なインライン・デフォルトに変換する
- 重複インライン・デフォルトを参照する既存のカラム定義は、更新されて共有可能なインライン・デフォルトを参照する

@report_only のデフォルト値は yes です。

@show_progress

yes に設定し、@report_only を no に設定した場合、sp_merge_dup_inline_default はシャープ記号を表示して進行状況を示します。

@show_progress のデフォルト値は no です。

例 **例 1** オプションを指定しないで、sp_merge_dup_inline_default を pubs2 データベースに対して実行します。sp_merge_dup_inline_default を実行しても、何も変更されませんが、ユニークなインライン・デフォルトの概数を示す情報メッセージが表示されます。

```
sp_merge_dup_inline_default
```

```
=====
```

```
sp_merge_dup_inline_default is used to identify duplicate inline default objects,
subsequently to convert one of them into sharable inline default object and remove the
rest.As the result, it will remove entries from sysobjects, syscomments and
sysprocedures.It will also update entries in syscolumns, syscomments and
sysprocedures.
```

```
Following is the current state of your inline default objects found out by
sp_merge_dup_inline_default and what it could potentially do to them.By default,
sp_merge_dup_inline_default only reports the current state and this warning
message.If
you really intend to carry out the changes, please rerun this stored procedure using
```

```
sp_merge_dup_inline_default @report_only = "NO"
```

Database pubs2 has about 0 unique inline defaults. If you convert them into sharable inline defaults, the rest of total 0 duplicate defaults can be removed from the system catalogs.

=====

例2 ユニークなインライン・デフォルトを共有可能なインライン・デフォルトに変換します。

```
sp_merge_dup_inline_default @report_only = 'NO'
```

Total 2 duplicate defaults are removed and 7 defaults are converted to sharable inline defaults. Database is modified and in single-user mode. System Administrator (SA) must reset it to multi-user mode with sp_dboption.

例3 重複インライン・デフォルトがない場合、出力は次のようになります。

```
sp_merge_dup_inline_default @report_only = 'NO'
```

Database is not modified. Please try it later if duplicate inline defaults do exist and the current resource limitation is preventing this conversion process.

例4 進行を示すために show_progress パラメータを含めます。

```
sp_merge_dup_inline_default @report_only = 'NO', @show_progress = "YES"
```

Calculating...

Converting...

```
[# ]
[##### ]
[##### ]
[##### ]
[##### ]
[##### ]
[##### ]
[##### ]
```

Total 2 duplicate defaults are removed and 7 defaults are converted to sharable inline defaults.

Database is modified and in single-user mode.

System Administrator (SA) must reset it to multi-user mode with sp_dboption

使用法

- sp_merge_dup_inline_default は、システム・データベースで実行できません。
- ユーザ・データベースをシングルユーザ・モードにしてから、sp_merge_dup_inline_default を実行する必要があります。
- システム・プロシージャがアボートされた場合、sp_merge_dup_inline_default を再実行できます。

- sp_merge_dup_inline_default が Adaptive Server がロック不足であることを通知するエラー・メッセージを発行する場合、以下を実行します。
 - ロック数の値を増やします。または
 - sp_setpglockpromote または sp_setrowlockpromote を使用したロック・プロモーション・スレッシュホールドを減らします。

sp_merge_dup_inline_default を再実行し、sp_merge_dup_inline_default の実行後に値を再設定します。
- sp_merge_dup_inline_default は、デフォルト値がリテラル文字列定数または単純な数字 (リテラル文字列定数はエスケープされた文字列のデリミタを含めることはできません) であるインライン・デフォルト・オブジェクトのみを変更します。
- sp_merge_dup_inline_default は、syscomments のソース・テキストが “encrypted” の場合、重複インライン・デフォルト・オブジェクトを削除しません。

パーミッション

sp_merge_dup_inline_default のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage database 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

sp_metrics

説明	QP 測定基準 (デフォルトの実行グループである、各データベースのグループ 1 で常に取得) と、クエリについての統計を、バックアップ、削除、フラッシュします。
構文	<code>sp_metrics ['backup' backup_group_ID 'drop', 'gid[, 'id'] 'flush' 'help', 'command']</code>
パラメータ	<p>backup 保存された QP 測定基準をデフォルトの実行グループからバックアップ・グループに移動し、QP 測定基準を古いサーバからバックアップ・グループにバックアップし、保存された QP 測定基準をデフォルトの実行グループからバックアップ・グループに移動します。</p> <p>backup_group_ID 古いサーバからバックアップ・グループへの QP 測定基準のグループの ID です。デフォルトの実行グループ内に保存された QP 測定基準をバックアップ・グループに移します。</p> <p>drop システム・カタログから QP 測定基準を削除します。sp_metrics で 'id' を指定しなかった場合、'gid' で指定したグループ全体が削除されます。</p> <p>gid システム・カタログの QP 測定基準のグループ ID です。</p> <p>id システム・カタログの QP 測定基準の ID です。</p> <p>flush メモリ内のすべての集計測定基準をシステム・カタログにフラッシュします。メモリ内のすべての文の集計測定基準はゼロ設定されます。</p> <p>'help', 'command' sp_metrics コマンドの使用についての情報を表示します。</p>
例	<p>例 1 デフォルト・グループの QP 測定基準をバックアップ・グループに移します。</p> <pre>sp_metrics 'backup', '3'</pre> <p>例 2 sp_metrics flush についての情報を示します。</p> <pre>sp_metrics 'help', 'flush'</pre>

使用法 測定基準情報には、`sysquerymetrics` ビューに対して `order by` を指定した `select` 文を使用してアクセスします。

古いサーバの QP 測定基準をバックアップ・グループにバックアップするときには使用します。保存された QP 測定基準をデフォルトの実行グループからバックアップグループに移動したり、システム・カタログから QP 測定基準を削除したりするときには使用します。すべての集計測定基準をメモリからシステム・カタログにフラッシュします。

パーミッション `sp_metrics` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限または <code>monitor qp performance</code> 権限 (<code>filter</code> 、 <code>show</code> 、 <code>help</code> の場合) を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — NULL • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — すべての入力パラメータ • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 コマンド [select](#)、[set](#)
 プロシージャ [sp_configure](#)

sp_modify_resource_limit

説明	リソース制限を変更します。この変更は、新しい制限値を指定するか、または制限値を超えた場合の動作を指定する（またはその両方を指定する）ことによって行われます。
構文	<code>sp_modify_resource_limit {name, appname} rangename, limittype, limitvalue, enforced, action, scope</code>
パラメータ	<p>name 制限が適用される Adaptive Server ログインです。name か appname またはその両方を指定してください。特定のアプリケーションを使用するすべてのユーザに適用される制限を変更するには、name に null を指定します。</p> <p>appname 制限が適用されるアプリケーションの名前です。name か appname またはその両方を指定してください。制限を name が使用するすべてのアプリケーションに適用するには、appname に null を指定します。制限を特定のアプリケーションに適用する場合は、クライアント・プログラムがログイン・パケット内で Adaptive Server に渡すアプリケーション名を指定します。</p> <p>rangename 制限が実施される時間範囲です。この値は変更できません。リソース制限をユニークに識別するには null 以外の値を指定してください。</p> <p>limittype 制限が適用されるリソースのタイプです。この値は変更できません。リソース制限をユニークに識別するには null 以外の値を指定してください。その値を次に示します。</p>

制限タイプ	説明
row_count	クエリが返すことができるローの数を制限する。
elapsed_time	クエリ・バッチまたはトランザクションが実行できる秒数を制限する。
io_cost	クエリを処理するための実際のコストまたはオペティマイザのコスト見積もりのいずれかを制限する。
tempdb_space	単一セッション間に持てる tempdb データベースのページ数を制限する。

limit_value

Adaptive Server が制限を実施する前にログインまたはアプリケーションが使用できるサーバ・リソースの最大量です。2³¹ 以下の正整数を指定するか、または null を指定して、すでに設定した値を使用します。次の表に、制限タイプ別の制限値を示します。

制限タイプ	制限値
row_count	制限値が有効になる前にクエリが返すことのできる最大ロー数。
elapsed_time	制限値が有効になる前にクエリ・バッチまたはトランザクションが実行できる最大秒数。
io_cost	オペディマイザのコスト計算式から計算した、単位のない値。
tempdb_space	単一セッションで持てるテンポラリ・データベースのページ数を制限する。

enforced

クエリの実行前、または実行中に制限を実施するかどうかを指定します。この値は変更できません。プレースホルダとして null を使用します。

action

制限値を超えたときに実行される動作です。次のコードは、すべての制限タイプに適用されます。

動作コード	説明
1	警告を出す。
2	クエリ・バッチをアボートする。
3	トランザクションをアボートする。
4	セッションを強制終了する。
null	既存の値を保持する。

scope

制限が適用されるスコープです。この値は変更できません。プレースホルダとして null を使用できます。

例

例 1 *weekends* 時間範囲中に “robin” が使用する、すべてのアプリケーションに適用されるリソース制限値を修正します。クエリが 3,000 を超えるローを返すことが予期される場合は、警告メッセージを発行します。

```
sp_modify_resource_limit robin, NULL, weekends, row_count, 3000, NULL,
1, NULL
```

例 2 すべての曜日と時間範囲で、アプリケーション *acctg* に適用されるリソース制限値を修正します。クエリ処理の予想時間が 45 秒を超えると、クエリ・バッチをアボートします。

```
sp_modify_resource_limit NULL, acctg, "at all times", elapsed_time,
45, 2, 2, 6
```

例 3 この例では、時間範囲 `tu_wed_7_10` の間、`payroll` アプリケーションのすべてのユーザに対して経過時間を制限するリソース制限の値を変更します。経過時間の制限値を 120 秒から 90 秒に減らしています。実行時間、実行するアクション、スコープの値は変更されていません。

```
sp_modify_resource_limit NULL, payroll, tu_wed_7_10,
elapsed_time, 90, null, null, 2
```

例 4 この例では、時間範囲 `saturday_night` の間、“`joe_user`” が実行するすべてのアドホック・クエリとアプリケーションのロー・カウントを制限するリソース制限によって実行されるアクションを変更します。アクションの変更前の値は 3 で、クエリが指定されたロー・カウントを超えると、トランザクションがアボートされるよう指定していました。この値を 2 に変更し、クエリ・バッチをアボートするようにします。制限タイプ、実行時間、スコープの値は変更されていません。

```
sp_modify_resource_limit joe_user, NULL,
saturday_night, row_count, NULL, NULL, 2, NULL
```

使用法

- 制限が適用されるログインやアプリケーションを変更することはできません。また、新しい時間範囲、制限タイプ、適用時間、またはスコープを指定することはできません。
- リソース制限値を修正すると、そのログインかアプリケーション（またはその両方）のセッションごとの制限が、そのセッション用に実行されるクエリ・バッチの始めに、再度バインドされます。
- クエリやトランザクションによるサーバ・リソースの独占を防ぐために、システム管理者は `Adaptive Server Enterprise` のリソース制限機能を使用できます。しかし、リソース制限は時間範囲にバインドされるまでは完全に指定されたことになりません。

パーミッション

`sp_modify_resource_limit` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage resource limit</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 **マニュアル** 詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_add_resource_limit](#)、[sp_drop_resource_limit](#)、[sp_help_resource_limit](#)

sp_modify_time_range

説明 名前付き時間範囲に対応する開始日、開始時刻、終了日、または終了時刻を変更します。

構文 `sp_modify_time_range name, startday, endday, starttime, endtime`

パラメータ

name

時間範囲の名前です。master データベースの `systemranges` システム・テーブルに保管されている時間範囲の名前を指定してください。

startday

時間範囲の開始曜日です。master データベースの `syslanguages` システム・テーブルに保管されているデフォルト・サーバ言語用の完全な曜日名を指定するか、または既存の `startday` を維持する `null` を指定してください。

endday

時間範囲の終了曜日です。master データベースの `syslanguages` システム・テーブルに保管されているデフォルト・サーバ言語用の完全な曜日名を指定するか、または既存の `end day` を維持する `null` を指定してください。 `endday` は `startday` の曜日よりも後でも前でもかまいません。また、`startday` と同じ曜日でもかまいません。

starttime

時間範囲の開始時刻です。 `starttime` は、24 時間表示形式で 00:00 ~ 23:59 の値を指定します。次のフォーマットを使用するか、または既存の `starttime` を維持する `null` を指定します。

"HH:MM"

endtime

時間範囲の終了時刻です。 `endtime` は、24 時間表示形式で 00:00 (真夜中) ~ 23:59 の値を指定します。次のフォーマットを使用するか、または既存の `endtime` を維持する `null` を指定します。

"HH:MM"

`endtime` には、`starttime` に指定した時刻より後の時刻を指定します (`endtime` に 00:00 を指定する場合を除く)。

注意 まる 1 日にわたる時間範囲の場合は、開始時刻を 00:00 に、終了時刻を 23:59 に設定します。

例

例 1 “business_hours” 時間範囲の終了日を金曜日から土曜日に変更します。既存の開始日、開始時刻、終了時刻は保持します。

```
sp_modify_time_range business_hours, NULL, Saturday, NULL, NULL
```

例 2 “before_hours” 時間範囲の新しい終了日と終了時刻を指定します。

```
sp_modify_time_range before_hours, Monday, Saturday, NULL, "08:00"
```

使用法

- “at all times” 時間範囲は変更できません。
- 複数の時間範囲と重複する時間範囲を設定できます。
- システム・プロシージャによって時間範囲が変更されても、現在進行中のセッションを対象としたアクティブな時間範囲には影響しません。
- トランザクションをスコープとして持つリソース制限に対する変更が、現在進行中のトランザクションに影響することはありません。

パーミッション

sp_modify_time_range のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage resource limit 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照

マニュアル 詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_add_resource_limit](#)、[sp_add_time_range](#)、[sp_drop_time_range](#)

sp_modifylogin

説明	Adaptive Server 15.7 以降では、このシステム・プロシージャは推奨されていません。Adapter Server のログイン・アカウントを修正するには、alter login コマンドを使用します。
構文	なし
パラメータ	なし
使用法	なし

sp_modifystats

説明 システム管理者や、このプロシージャを実行してターゲット・テーブルの統計を更新するパーミッションを付与されたすべてのユーザは、`sysstatistics` のカラムの密度値を変更できます。

構文

```
sp_modifystats [database].[owner].table_name,
               {"column_group" | "all"},
               MODIFY_DENSITY,
               {range | total},
               {absolute | factor},
               "value"
```

または

```
sp_modifystats [database].[owner].table_name,
               column_name | null,
               REMOVE_SKEW_FROM_DENSITY
               REMOVE_STICKINESS
```

パラメータ

table_name

変更するテーブルの名前を指定します。テーブルが別のデータベース内にある場合はデータベース名を指定し、データベース内にその名前のテーブルが複数ある場合は所有者名を指定します。`owner` のデフォルト値は現在のユーザで、`database` のデフォルト値は現在のデータベースです。

column_group

カラム名の順序リストを指定します。密度値などで、複数のカラムの統計を変更するには、統計の作成に使用する順序でカラムを列挙します。複数のカラム名を指定するには、カンマで区切ります。たとえば `a1`、`a2`、`a3`、`a4` カラム上に密度値を持つテーブルの場合は、次のようになります。

- “`a1`” はカラム `a1` を修正します。
- “`a1,a2,a3`” はカラム・グループ `a1`、`a2`、`a3` を修正します。
- `column_group` パラメータとともに、ワイルドカード文字 `%` を使用して、ある範囲の文字を表すこともできます。たとえば、“`a1,%,a3`” はグループ `a1`、`a2`、`a3` と `a1`、`a4`、`a3` などを修正します。“`a1,%`” はグループ `a1`、`a2` と `a1`、`a2`、`a3` などを修正しますが、`a1` は修正しません。“`a1%`” はグループ `a1`、`a2` と `a1`、`a2`、`a3` などを修正し、`a1` も修正します。

all

次のテーブルのカラム・グループを修正します。“`all`” はキーワードであるため、引用符が必要です。

MODIFY_DENSITY

カラムまたはカラム・グループの範囲の密度または合計の密度のいずれかについて、*value* パラメータで指定した粒度に修正できます。範囲セル密度は、ヒストグラムの範囲セルで示されるすべての値について重複平均値を表します。詳細については、『パフォーマンス & チューニング・ガイド』を参照してください。

REMOVE_STICKINESS

指定したカラムに関連付けられているスティッキネスを削除します。*null* を指定すると、テーブル内のすべてのカラムからスティッキネスが削除されます。

“スティッキネス”は、Adaptive Server が次の *update statistics* パラメータのメモリを保持するときに発生します。

- *using step values*
- *sampling*
- *histogram_tuning_factor*
- *hashing*
- *no_hashing*
- *partial_hashing*

フレーズがいったん “スティッキ” になると、パラメータを明示的に指定しなくても、Adaptive Server はそれ以降の *update statistics* コマンドにおいてそのカラムに対する動作を維持します。

range

範囲セル密度を修正します。

total

合計セル密度を修正します。

absolute

現在値を無視し、*value* パラメータで指定された数を使用します。

factor

value パラメータで現在の統計値を乗算します。

value

密度の指定値、または現在の密度値に掛ける値です。*absolute* を指定する場合は、0 から 1 までを指定する必要があります。

column_name

テーブル内のカラムの名前を指定します。

REMOVE_SKEW_FROM_DENSITY

システム管理者は、カラムの総密度を範囲密度と等しくなるように変更できます。これは、データが偏って存在するときに便利です。総密度は、頻度セルと範囲セルのすべての値の重複平均値を表します。ジョインに対して一致するロー数や、クエリ最適化時に値が未定の探索指数に対して一致するロー数の見積もりに、総密度が使用されます。詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。

REMOVE_SKEW_FROM_DENSITY は、このカラムが先行属性となる複合カラム統計の総密度の更新も行います。多くの場合、このカラムが先行属性となる複合インデックスでこれらの複合カラム統計は作成されますが、複合 `update statistics` コマンドを発行したときにも作成されます。

- 例**
- 例 1** `tab_1` テーブル内のカラム・グループ `c00`、`c01` の範囲密度を `0.50000000` に変更します。
- ```
sp_modifystats "tab_1", "c00, c01", MODIFY_DENSITY, range, absolute, "0.5"
```
- 例 2** `tab_1` 内のカラム・グループ `c00`、`c01` の総密度に `0.5` を掛けます。つまり、半分にします。
- ```
sp_modifystats "tab_1", "c00,c01", MODIFY_DENSITY, total, factor, "0.5"
```
- 例 3** `tab_1` テーブル内のすべてのカラムの総密度に、`0.5` を掛けます。
- ```
sp_modifystats "tab_1", "all", MODIFY_DENSITY, total, factor, "0.5"
```
- 例 4** `c12` で始まるすべてのカラム・グループの総密度は、範囲密度と等しくなるように変更されます。
- ```
sp_modifystats "tab_1", "c12" REMOVE_SKEW_FROM_DENSITY
```

使用方法

- `sysstatistics` 内のカラム (複数可) の密度値をシステム管理者が修正できます。
- `optdiag` を使用してテーブルの統計を表示します。テーブル密度と `optdiag` の使用方法の詳細については、『パフォーマンス & チューニング・シリーズ』を参照してください。
- `sp_modifystats` を使用して統計に加えた修正は、`update statistics` を実行したときに上書きされます。確実に最新の、修正後の統計値を使用するために、`update statistics` を実行してから `sp_modifystats` を実行してください。
- `sp_modifystats` では `sysstatistics` テーブルに保管されている情報が修正されるため、統計のバックアップをとってから、運用システムで `sp_modifystats` を実行してください。

パーミッション `sp_modifystats` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、オブジェクトに対する <code>update statistics</code> 権限または <code>manage any statistics</code> 権限を持つユーザであることが必要。プロシージャに対する <code>execute</code> パーミッションが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、オブジェクトに対する <code>update statistics</code> または <code>sa_role</code> を持つユーザであることが必要。プロシージャに対する <code>execute</code> パーミッションが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

使用するテーブル `sysstatistics`

参照 **コマンド** `update statistics`

sp_modifythreshold

説明 別のスレッシュヨルド・プロシージャ、空き領域レベル、またはセグメント名と対応させることによって、スレッシュヨルドを変更します。

構文 `sp_modifythreshold dbname, segname, free_space
[, new_proc_name][, new_free_space][, new_segname]`

パラメータ

dbname

スレッシュヨルドを変更するデータベースです。これは、現在のデータベースの名前でなければなりません。

segname

空き領域をモニタするためのセグメントです。デフォルトのセグメント (“default”) を指定するときは、引用符を使用してください。

free_space

スレッシュヨルドを超過した時点での空きページの数です。セグメント内の空き領域がこのレベルより下になると、Adaptive Server は対応するストアド・プロシージャを実行します。

new_proc_name

スレッシュヨルドを超過した場合に実行する新しいストアド・プロシージャです。このプロシージャは、現在の Adaptive Server または Open Server 上のどのデータベースでも見つけることができます。スレッシュヨルドは、リモートの Adaptive Server でプロシージャを実行することはできません。

new_free_space

スレッシュヨルドに対応させる新しい空きページ数です。セグメント内の空き領域がこのレベルより下になると、Adaptive Server は対応するストアド・プロシージャを実行します。

new_segname

空き領域をモニタするための新しいセグメントです。デフォルトのセグメント (“default”) を指定するときは、引用符を使用してください。

例 **例 1** セグメントの空き領域が 200 ページではなく 175 ページ未満になると実行されるように、mydb データベースの “default” セグメントのスレッシュヨルドを変更します。NULL は、プロシージャ名が変更されていないことを示すプレースホルダです。

```
sp_modifythreshold mydb, "default", 200, NULL, 175
```

例 2 new_proc プロシージャを実行するように、mydb の data_seg セグメントのスレッシュヨルドを変更します。

```
sp_modifythreshold mydb, data_seg, 250, new_proc
```

使用法

`sp_modifythreshold` を使用して、空き領域の量やラストチャンス・スレッシュホルドのセグメント名を変更することはできません。

スレッシュホルドの超過

- スレッシュホルドを超過すると、Adaptive Server は関連したストアド・プロシージャを実行します。Adaptive Server は次の検索パスを使用して、スレッシュホルド・プロシージャを検索します。
 - プロシージャ名にデータベースを指定しない場合は、Adaptive Server がスレッシュホルドを超過したデータベース内を検索します。
 - このデータベース内にプロシージャがなく、しかもプロシージャ名が “sp_” で始まっている場合、Adaptive Server は `sybsystemprocs` データベース内を検索します。

プロシージャがどちらのデータベース内にもなければ、Adaptive Server はエラー・メッセージをエラー・ログに送信します。

- Adaptive Server はヒステリシス値 (グローバル変数 `@@thresh_hysteresis`) を使用して、空き領域内での変動に対するスレッシュホルドの感度を決定します。スレッシュホルドは、そのプロシージャを実行するとアクティブでなくなります。スレッシュホルドは、セグメントの空き領域量がスレッシュホルドを超えて `@@thresh_hysteresis` ページ分に到達するまで、アクティブでない状態のままです。これによって、スレッシュホルドが空き領域のわずかな変化に反応して、繰り返しそのプロシージャを実行することがなくなります。

ラストチャンス・スレッシュホルド

- デフォルトでは、Adaptive Server は、ログが存在するセグメントの空き領域を監視し、空き領域がトランザクション・ログを正常にダンプするために必要な量を下回ると `sp_thresholdaction` を実行します。この空き領域量「ラストチャンス・スレッシュホルド」は、Adaptive Server により計算されます。ユーザは変更できません。
- トランザクションのログを取る前にラストチャンス・スレッシュホルドを超過すると、Adaptive Server は、ログの領域が解放されるまでトランザクションを中断します。特定のデータベースに対してこの動作を変更するには、`sp_dboption` を使用します。`abort tran on log full` オプションを `true` に設定すると、ラストチャンス・スレッシュホルドの超過時にログを記録していなかったすべてのトランザクションが、Adaptive Server によってロールバックされます。

- `sp_modifythreshold` を使用して、ラストチャンス・スレッシュホールドに対応する空き領域値またはセグメント名を変更することはできません。
- 別々のセグメントにログを保管するデータベースだけが、ラストチャンス・スレッシュホールドを持つことができます。トランザクション・ログを別のデバイスに移動するには、`sp_logdevice` を使用します。

その他のスレッシュホールド

- データベースにはそれぞれ、ラストチャンス・スレッシュホールドを含めて最大 256 個のスレッシュホールドを作成できます。
- 各スレッシュホールドは、最も近いスレッシュホールドから `@@thresh_hysteresis` ページの 2 倍以上離れていなければなりません。
- `sp_helpthreshold` を使用すると、既存のスレッシュホールドの情報を入手できます。
- セグメントからスレッシュホールドを削除するには、`sp_droptreshold` を使用します。

スレッシュホールド・プロシージャの作成

- データベースのスレッシュホールド・プロシージャの作成は、`create procedure` パーミッションを持つすべてのユーザが実行できます。通常、システム管理者が `master` データベースに `sp_thresholdaction` を作成し、データベース所有者がユーザ・データベースにスレッシュホールド・プロシージャを作成します。
- `sp_modifythreshold` は、指定されたプロシージャが存在しているかどうかを検証しません。存在していないプロシージャにスレッシュホールドを対応させることができます。
- `sp_modifythreshold` は、スレッシュホールド・プロシージャを変更するユーザに “`sa_role`” が直接付与されているかどうかを確認します。スレッシュホールド・プロシージャを変更した時点でアクティブになっているシステム標準の役割はすべて、プロシージャを記述するユーザに対する有効な役割として `systhresholds` 内に保管されます。ただし、スレッシュホールドの起動時にアクティブになっているのは、直接付与されたシステム標準の役割だけです。間接的に与えられたシステム標準の役割とユーザ定義の役割はアクティブになっていません。
- Adaptive Server は、次の 4 つのパラメータをスレッシュホールド・プロシージャに渡します。
 - `@dbname, varchar(30)` : データベースを特定します。

- `@segment_name, varchar(30)` : セグメントを特定します。
- `@space_left, int` : スレッシュホールドに対応した空きページ数を表します。
- `@status, int` : ラストチャンス・スレッシュホールドでは 1、その他のスレッシュホールドでは 0 です。

これらのパラメータは、名前ではなく、位置に基づいて渡されません。スレッシュホールド・プロシージャは、パラメータに対して別の名前を使用できますが、パラメータを宣言するときは、示されている順序で、正確なデータ型を使用する必要があります。

- スレッシュホールドごとに別のプロシージャを作成する必要はありません。メンテナンスをできるだけ少なくするためには、**Adaptive Server** 内のすべてのスレッシュホールドに対して実行されるスレッシュホールド・プロシージャを 1 つ、**sybsystemprocs** データベース内に作成します。
- エラー・ログに出力を送信するには、`print` 文と `raiserror` 文をスレッシュホールド・プロシージャに入れます。

スレッシュホールド・プロシージャの実行

- スレッシュホールドの超過時に開始されたタスクは、バックグラウンド・タスクとして実行されます。これらのタスクには、対応する端末セッションまたはユーザ・セッションがありません。これらのタスクの実行中に `sp_who` を実行すると、`status` カラムに“background”が表示されます。
- **Adaptive Server** は、スレッシュホールドを変更したユーザのパーミッションによってスレッシュホールド・プロシージャを実行します。このパーミッションはユーザが、`sp_modifythreshold` を実行した時点で持っていたものです。ただし、その時点以降に取り消されたパーミッションは差し引かれます。
- 各スレッシュホールド・プロシージャは、プロシージャが実行されているとみなすかぎり、1 つのユーザ接続を使用します。

空き領域の計算を無効にする

警告! 空き領域の計算が無効になっている場合、システム・プロシージャは領域の割り付けについての正確な情報を提供できません。

- ログ以外のセグメントで空き領域の計算を無効にするには、`sp_dboption` の `no free space acctg` オプションを使用します。
- ログ・セグメントの空き領域の計算を無効にすることはできません。

パーミッション sp_modifythreshold のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage database 権限を持つユーザーであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か sa_role が付与されたユーザーであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

参照 マニュアル 詳細については、『システム管理ガイド』を参照してください。

コマンド [create procedure](#)、[dump transaction](#)

システム・プロシージャ [sp_addthreshold](#)、[sp_dboption](#)、[sp_droptreshold](#)、[sp_helpthreshold](#)、[sp_thresholdaction](#)

sp_monitor

説明

Adaptive Server についての統計情報を表示する。

構文

多くのコマンド・タイプには独自のパラメータがあるため、`sp_monitor` 構文はコマンド・タイプで分けてわかりやすくしています。次のコード・パラグラフは、後ろに各コマンド・タイプのインタフェースが続く、ストアド・プロシージャ全体の構文を示します。

```
sp_monitor [[connection | statement], [cpu | diskio | elapsed time]]
           [event, [spid ]]
           [procedure, [dbname, [procname[, summary | detail]]]]
           [enable] [disable]
           [help],
           [deadlock][procstack]
```

パラメータ

connection

各接続の情報を表示します。`connection` は次のモニタリング・テーブルを使用します。

- `monProcessSQLText`
- `monProcessActivity`

statement

各文の情報を表示します。`statement` は次のモニタリング・テーブルを使用します。

- `monProcessSQLText`
- `monProcessStatement`

cpu | diskio | elapsed time

これらのパラメータによって `sp_monitor connection` または `sp_monitor statement` の出力が順序付けられます。

- `cpu` — 各接続または文で消費された CPU 時間を示します。
- `diskio` — 各接続または文で実行された物理読み込み回数を示します。
- `elapsed time` — 各接続または文の CPU 時間と待機時間の合計を示します。

event

指定すると 3 種類の情報が表示されます。指定する値は次のとおりです。

- オプションを指定しない場合、ユーザ・タスクのみが表示されます。
- `sp_monitor, event, "-1"` と指定すると、すべてのタスクに関する待機情報 (ユーザとシステムの両方) が表示されます。
- `sp_monitor, event, "spid"` と指定すると、指定されたサーバ・プロセス ID に関する待機情報のみが表示されます。

spid

特定のタスクの *spid* を入力すると、そのタスクの **event** 情報を取得できます。*spid* は数値を引用符で囲んで指定します。

procedure

ストアド・プロシージャに関する次の統計を表示します。

- **ProcName** — モニタリングされているストアド・プロシージャ。
- **DBNAME** — ストアド・プロシージャが格納されているデータベース。
- **NumExecs** — この特定のストアド・プロシージャの概算の実行回数。
- **AvgCPUTime** — ストアド・プロシージャの実行にかかる平均 CPU 時間。
- **AvgPhysicalReads** — ストアド・プロシージャによって実行される平均ディスク読み込み回数。
- **AvgLogicalReads** — ストアド・プロシージャによって実行される平均論理読み込み回数。
- **AvgMemUsed_KB** — ストアド・プロシージャで使用される平均メモリ容量 (KB 単位)。

procedure は `monSysStatement` モニタリング・テーブルを使用します。

dbname

指定されたデータベースのプロシージャに関する情報を表示します。

procname

指定されたプロシージャに関する情報を表示します。

summary | detail

プロシージャのすべてのインスタンスの平均値を示す要約情報か、ストア・プロシージャの各インスタンスの詳細情報を表示します。

enable

`sp_monitor` の新しいオプションを有効にします。モニタリングを開始するために必要な設定パラメータをオンにします。

disable

モニタリングを無効にします。

help

`sp_monitor` の構文と例が表示され、デッドロック分析にこのプロシージャを使用することに関する詳しい情報もレポートされます。

```
sp_monitor 'help', 'deadlock'
```

また、`help` オプションは、コマンド別のサンプルも提供します。

deadlock

`sp_monitor` に対し `monDeadlock` テーブルから履歴データを処理するように指示し、デッドロックの各インスタンスのブロック出力を表示します。

procstack

ネストが深いストア・プロシージャの実行コンテキストを含む、タスクの実行コンテキストを調べます。実行されたプロシージャ・スタックは、`monProcessProcedures` モニタリング・テーブルから抽出されます。

例

例 1 Adaptive Server の利用率に関する情報をレポートします。

```
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           connections
-----
394(67)                 5392(53)              0(0)                  15(1)
```

例 2 接続に関する情報の表示方法を示します。

```
1> sp_monitor "connection"
2> go
spid      LoginName      ElapsedTime    LocksHeld      SQLText
-----
12        sa              90300          2              exec get_employee_salaries
27        sa              17700          1              exec get_employee_perks
```

デフォルトでは、出力は ElapsedTime の降順でソートされます。

例 3 物理読み込みの実行回数が最も多い接続が示されます。

```
1> sp_monitor "connection","diskio"
2> go
spid      LoginName      Physical_Reads  LocksHeld      SQLText
-----
12        sa              117            2              exec get_employee_salaries
27        sa              1              0              exec get_employee_perks
```

例 4 各文についての情報を表示します。

```
1> sp_monitor "statement"
2> go
spid      LoginName      ElapsedTime      SQLText
-----
12        sa              100              exec get_employee_salaries
```

例 5 各タスクが待機していたイベントとその待機時間を、待機時間の降順で表示します。

```
1> sp_monitor "event"
2> go
SPID      WaitTime      Description
-----
6         108200       hk: pause for some time
29        108200       waiting for incoming network data
10        107800       waiting while allocating new client socket
15        17100        waiting for network send to complete
14        5900         waiting for CTLIB event to complete
14        400          waiting for disk write to complete
7         200          hk: pause for some time
7         100          waiting on run queue after yield
12        100          waiting for network send to complete
```

例 6 spid 14 のイベント・データを表示します。

```
1> sp_monitor "event", "14"
2> go
WaitTime      Description
-----
9000 waiting for CTLIB event to complete
600 waiting for disk write to complete
200 waiting for disk write to complete
100 waiting on run queue after yield
100 wait for buffer write to complete
```

例 7 最近実行されたプロシージャの要約を平均経過時間の降順で表示します。これは現在の状態ではなく履歴モニタリング情報です。

```
1> sp_monitor "procedure"
2> go

Average Procedure Statistics
=====

ProcName          DBName      AvgElapsedTime  AvgCPUTime  AvgWaitTime
AvgPhysicalReads AvgLogicalReads AvgPacketsSent  NumExecs
-----
neworder_remote  tpcc        1833            16          1083        26   96   0   6
neworder_local   tpcc        1394            13          1181        31  122   0  38
tc_startup       tpcc        1220             3          1157         0   3   0  59
delivery         tpcc        1000             0           800         23  49   0   2
```

使用法

注意 sp_monitor に関連付けられている新しいパラメータを使用する前に、それらのオプションを有効にするために必要なモニタリング・テーブルと関連するストアド・プロシージャを設定してください。詳細については、『パフォーマンス&チューニング・シリーズ：モニタリングと分析』の「モニタリング・テーブルのインストール」を参照してください。

- Adaptive Server は、これまでの作業量を一連のグローバル変数で記録します。sp_monitor は、これらのグローバル変数の現在値と、前回のプロシージャ実行時からグローバル変数がどの程度変化しているかを表示します。
- 各カラムでは、*number(number)-number%* または *number(number)* の形式で統計を表示します。

- 最初の数字は、秒数 (cpu_busy、io_busy、idle に対応) または Adaptive Server を再起動してからの総数 (他の変数に対応) を表します。
- カッコ内の数値は、前回 sp_monitor を実行してからの秒数または総数を表しています。パーセント記号は、sp_monitor を最後に実行してからの時間の比率を表します。

たとえば、レポートが cpu_busy を “4250(215)-68%” と表示しているとします。これは CPU がビジー状態であった時間が、Adaptive Server を前回起動してから 4250 秒、sp_monitor を前回実行してから 215 秒、sp_monitor を前回実行してからの総時間の 68% であったことを示しています。

total_read カラムの 394(67) の値は、Adaptive Server を前回起動してからのディスク読み込みが 394 回、sp_monitor を前回実行してからのディスク読み込みが 67 回であったことを示しています。

- 次の表は、各オプション・タイプがアクセスするモニタリング・テーブルを示しています。

表 1-24: モニタリング・タイプ別にアクセスされるモニタリング・テーブル

モニタリング・タイプ	アクセスされるテーブル	設定オプション	設定オプション・タイプ
connection	monProcessSQLext	max SQL text monitored	値
		SQL batch capture	Boolean
	monProcessActivity	wait event timing	Boolean
		per object statistics active	Boolean
procstack	monProcessProcedures	なし	該当なし
statement	monProcessSQLText	max SQL text monitored	値
		SQL batch capture	Boolean
	monProcessStatement	statement statistics active	Boolean
		per object statistics active	Boolean
		wait event timing	Boolean
event	monProcessWaits	wait event timing	Boolean
		process event waits	Boolean
procedure	monSysStatement	statement statistics active	Boolean
		per object statistics active	Boolean
		statement pipe max messages	値
		statement pipe active	Boolean
deadlock	monDeadlock	deadlock pipe max messages	値
		deadlock pipe active	Boolean

- `sp_monitor connection` は、T-SQL を実際に実行している接続のみをモニタし、すべての接続についてレポートするわけではありません。
- `sp_monitor` は master データベースから実行する必要があります。ただし、別のデータベースにアーカイブされているデッドロック・データを分析する場合は、このデータベースから `sp_monitor deadlock` を実行する必要があります。
- オプションなしで呼び出されると、`sp_monitor event` は、システム・タスクを含むすべてのタスクを表示しません。Adaptive Server バージョン 15.0.2 以降では、次の 3 種類の情報を event オプションで表示できるようになりました。
 - このオプションを指定しない場合、ユーザ・タスクのみが表示されます。
 - `sp_monitor, event, "-1"` と指定すると、すべてのタスクに関する待機情報 (ユーザとシステムの両方) が表示されます。
 - `sp_monitor, event, "spid"` と指定すると、指定されたサーバ・プロセス ID に関する待機情報のみが表示されます。
- 表 1-25 に、`sp_monitor` のレポートのカラム、同等のグローバル変数 (必要な場合)、およびそれらの意味を示します。`last_run`、`current_run`、`seconds` を除き、これらのすべてのカラム見出しは、グローバル変数の名前でもあります。ただし、すべてのグローバル変数には前に `@@` が付きます。また、グローバル変数によりレポートされる数の単位も異なっています。グローバル変数によってレポートされる数は、ミリ秒単位の CPU 時間ではなく、マシン目盛り (チック) です。

表 1-25: `sp_monitor report` 内のカラム

カラム見出し	同等の変数	意味
<code>last_run</code>		<code>sp_monitor</code> プロシージャを前回実行したときのクロック・タイム
<code>current_run</code>		現在のクロック・タイム
<code>seconds</code>		前回 <code>sp_monitor</code> を実行してからの秒数
<code>cpu_busy</code>	<code>@@cpu_busy</code>	Adaptive Server の CPU が Adaptive Server の処理に費やした CPU 時間 (秒数)
<code>io_busy</code>	<code>@@io_busy</code>	Adaptive Server が I/O 動作に費した CPU 時間 (秒数)
<code>idle</code>	<code>@@idle</code>	Adaptive Server がアイドル状態になっていた CPU 時間 (秒数)
<code>packets_received</code>	<code>@@pack_received</code>	Adaptive Server が読み込んだ入力パケット数
<code>packets_sent</code>	<code>@@pack_sent</code>	Adaptive Server が書き込みを行った出力パケット数
<code>packet_errors</code>	<code>@@packet_errors</code>	パケットの読み取りおよび書き込み中に Adaptive Server が検出したエラーの数
<code>total_read</code>	<code>@@total_read</code>	Adaptive Server によるディスク読み込み回数

カラム見出し	同等の変数	意味
total_write	@@total_write	Adaptive Server によるディスク書き込み回数
total_errors	@@total_errors	読み込みおよび書き込み中に Adaptive Server が検出したエラーの数
connections	@@connections	Adaptive Server に対するログイン回数またはログイン試行回数

- Adaptive Server を起動した後で初めて sp_monitor を実行したときには、カッコで囲まれた値は無効です。
- Adaptive Server のハウスキーピング・タスクは、サーバのアイドル・サイクルを使用して、変更したページをキャッシュからディスクに書き込みます。この処理は、sp_monitor でレポートされる cpu_busy、io_busy、idle カラムの値に影響します。ハウスキーピング・タスクを無効にしてこれらの影響を受けないようにするには、housekeeper free write percent 設定パラメータを 0 に設定します。

```
sp_configure "housekeeper free write percent", 0
```
- sp_monitor は、システムに典型的な負荷がかかっているときに実行します。
- 通常、次の順序でプロシージャを実行します。
 - sp_monitor enable を実行する。
 - sp_monitor オプションを呼び出す。
 - モニタリングが終了したら sp_monitor disable を実行する。
- sp_monitor procedure を使用すると、返されるロー数が非常に多くなることがあります。その場合は、detail オプションではなく summary オプションを使用することもできます。アクティブなシステムではこのコマンドが完了するまでにしばらく時間がかかることがあります。

パーミッション

sp_monitor のパーミッション・チェックは、細密なパーミッションが有効でも無効でも同じです。

- subsystemprocs のデータベース所有者は、sp_monitor を実行して他のユーザに execute パーミッションを付与できます。
- ストアド・プロシージャは execute as owner で作成します。所有者は sa です。所有者には mon_role が必要ですが、ユーザ sa はこれをデフォルトで持っています。

詳細については、『パフォーマンス&チューニング・シリーズ：モニタリングと分析』の「モニタリング・テーブルのインストール」を参照してください。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_who](#)

sp_monitorconfig

説明 インデックス、オブジェクト、データベース、およびカーネル・リソース・メモリ・プール用のメタデータ記述子に関するキャッシュ使用率の統計を表示します。また、`sp_monitorconfig` は参照整合性クエリに使用される補助スキャン記述子の統計とトランザクション記述子および DTX パティシパント用の使用統計をレポートします。

構文 `sp_monitorconfig "configname"[, "result_tbl_name"][, "full"]`

パラメータ

configname

モニタリング情報を問い合わせる設定パラメータ名の `all` (全体) またはその一部です。有効な設定パラメータは「使用法」セクションに記載されています。`all` を指定すると、サーバ内のすべてのインデックス、オブジェクト、データベース、補助スキャン記述子についての記述子ヘルプ情報が表示されます。

“*result_tbl_name*”

(オプション) ストアド・プロシージャの結果を保存するために作成するテーブルの名前です。*result_tbl_name* のテーブル名として存在しない名前を渡した場合は、結果セットを保存するためのテーブルが `sp_monitorconfig` によって作成されます。

“*full*”

指定した *configname* に対応する一連の値を返します。値は次のとおりです。

- `config_val` — 設定値がレポートされる。
- `system_val` — 値が設定されていない場合にシステムのデフォルト値をレポートします。
- `total_val` — 実際に使用される値がレポートされる。

例 1 開いている項目をすべて表示します。

```
sp_monitorconfig "open"
```

```
Configuration option is not unique.
```

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

例 2 すべての設定のステータスを表示します。

```
sp_monitorconfig "all"
```

```
-----
```

```
Usage information at date and time:May 6 2010 4:32PM.
```

Name	Num_free	Num_active	Pct_act	Max_Used	Reuse_cnt	Instance_Name
additional network memory	1358436	809440	37.34	825056	0	NULL
audit queue size	100	0	0.00	0	0	NULL
disk i/o structures	256	0	0.00	29	0	NULL
heap memory per user	4096	0	0.00	0	0	NULL
. . .						
size of process object he	3000	0	0.00	0	0	NULL
size of shared class heap	6144	0	0.00	0	0	NULL
size of unilib cache	306216	816	0.27	816	0	NULL
txn to pss ratio	400	0	0.00	0	0	NULL

例 3 この例は、61 個のオブジェクト・メタデータ記述子がアクティブであることを示しています。空いている記述子は 439 個あります。Adaptive Server が前回起動してからピーク時で使用された最大数は、61 です。

```
sp_monitorconfig "open objects"
```

```
Usage information at date and time:Apr 22 2002 2:49PM
```

Name	Num_free	Num_active	Pct_act	Max_Used	Reuse_cnt	Instance_Name
number of open objects	439	61	12.20	61	0	NULL

この場合に、たとえばサイズを 550 に再設定すると、メタデータ記述子で使用した最大値 439 に加えて、10 パーセント分の空き領域を確保できます。

```
sp_configure "number of open objects", 330
```

例 4 この例は、インデックス・メタデータ記述子の最大数が、44 個であることを示しています。

```
sp_monitorconfig "open indexes"
```

```
Usage information at date and time:Apr 22 2002 2:49PM
```

Name	Num_free	Num_active	Pct_act	Max_Used	Reuse_cnt	Instance_Name
number of open indexes	556	44	7.33	44	0	NULL

最小許容値である 100 まで、サイズを再設定できます。

```
sp_configure "number of open indexes", 100
```

例 5 この例では、Adaptive Server は 200 を使用するよう設定されていますが、アクティブなスキャン記述子の数は 30 と表示されています。number of aux scan descriptors 設定パラメータを使用して、値を 32 以上に再設定します。32 個のスキャン記述子の格納には 36 の設定が安全で、領域に 10 パーセントの余裕を持たせます。

```
sp_monitorconfig "aux scan descriptors"
```

```
Usage information at date and time:Apr 22 2002 2:49PM
```

Name	Num_free	Num_active	Pct_act	Max_Used	Reuse_cnt	Instance_Name
number of aux scan descri	170	30	15.00	32	0	NULL

例 6 この例では、Adaptive Server は 5 つのオープン・データベースで構成され、そのすべてが現在のセッションで使用中です。

```
sp_monitorconfig "number of open databases"
```

Name	Num_free	Num_active	Pct_act	Max_Used	Reuse_cnt	Instance_Name
number of open databses	0	5	100.00	5	Yes	NULL

ただし、Reuse_cnt カラムで示されているように、追加のデータベースを開くことが必要です。5 つのデータベースすべてが使われている場合、使用していないデータベースの記述子が再使用できないかぎり、エラーが発生することがあります。エラーの発生を防ぐには、number of open databases をより高い値に再設定します。

例 7 この例では、トランザクション記述子の 10.2 % だけが現在使用中です。ただし、Adaptive Server が前回起動されてからピーク時で使用されたトランザクション記述子の最大数は 523 です。

```
sp_monitorconfig "txn to pss ratio"
```

```
Usage information at date and time:Apr 22 2002 2:49PM
```

Name	Num_free	Num_active	Pct_act	Max_Used	Reuse_cnt	Instance_Name
txn to pss ratio	784	80	10.20	523	0	NULL

例 8 この例では、オプション・パラメータ result_tbl_name を使用してユーザ・テーブルを作成し、sp_monitorconfig の実行結果をこのテーブルに保存します。

```
create table sample_table
(Name varchar(35),Config_val int, System_val int, Total_val int,
Num_free int, Num_active int, Pct_act char(6), Max_used int,
Num_Reuse int, Date varchar(30))
```

```

create table sample_table
(Name varchar(35),
Config_val int,
System_val int,
Total_val int,
Num_free int,
Num_active int,
Pct_act char(6),
Max_Used int,
Reuse_cnt int,
Date varchar(30),
Instance_Name varchar(35))

```

sp_monitorconfig の 2 番目のパラメータが作成されたテーブルの名前になります。number of locks および number of alarms の値を sample_table で取得します。

```

sp_monitorconfig "locks", sample_table
sp_monitorconfig "number of alarms", sample_table

```

sample_table で取得した値を表示します。

```

select * from sample_table

```

Name	Config_val	System_val	Total_val	Num_free	Num_active
Pct_act	Max_used	Reuse_cnt	Date	Instance_Name	
number of locks	5000	684	5000	4915	85
1.70	117	0	Aug 23 2006	6:53AM	
number of alarms		40	0	40	28
30.00	13	0	Aug 23 2006	6:53AM	12

テーブルに保存された結果セットは、テーブルを削除するかまたはトランケートするまで累積されます。

注意 sample_table が別のデータベースにある場合、その完全修飾名を引用符で囲んで記述してください。

例 9 すべての設定の configure_value カラム、system_value カラム、run_value カラムを表示します。

```

sp_monitorconfig "all", null, "full"
go

```

Usage information at date and time:Mar 23 2004 5:15PM

Name	Configure Value	System Value	Run Value
Num_free	Num_active	Pct_act	Max_Used
			Reuse_cnt
			Instance_Name

additional network memory	0	2167876	2167876	
1358436	809440	37.34	825056	0
audit queue size	100	0	100	NULL
100	0	0.00	0	0
disk i/o structures	256	0	256	NULL
256	0	0.00	29	0
heap memory per user	4096	563	4096	NULL
4096	0	0.00	0	0
kernel resource memory	4096	0	4096	NULL
3567	529	12.92	529	0
max cis remote connection	0	100	100	NULL
100	0	0.00	0	0
. . .				
size of shared class heap	6144	0	6144	NULL
6144	0	0.00	0	0
size of unilib cache	0307032	307032	306216	NULL
816	0	.27	816	0
txn to pss ratio	16	0	16	NULL
400	0	0.00	0	0

使用法

- max cis remote connections 設定パラメータに config_value が指定されている場合、system_val はゼロ (0) の値をレポートします。
- 指定した元の値より小さい値を使用してリソースを再設定する場合、リソースは縮小されず、Num_active 設定パラメータは Total_val よりも大きい数をレポートします。リソースの縮小と数は、Adaptive Server が再起動されると正しくレポートされます。
- sp_monitorconfig は、インデックス、オブジェクト、およびデータベース用のメタデータ記述子に関するキャッシュ使用率の統計情報 (サーバが現在使用しているメタデータ記述子数など) を表示します。
- sp_monitorconfig は、使用中の補助スキャン記述子数もレポートします。クエリをテーブル上で実行した場合、スキャン記述子はテーブルのスキャンを管理します。
- sp_monitorconfig は、次のリソースをモニタします。

additional network memory	number of open indexes
audit queue size	number of open objects
heap memory per user	number of open partitions
max cis remote connection	number of remote connections
max memory	number of remote logins
max number network listeners	number of remote sites
memory per worker process	number of sort buffers
max online engines	number of user connections
number of alarms	number of worker processes
number of aux scan descriptors	partition groups
number of devices	permission cache entries
number of dtx participants	procedure cache size
number of java sockets	size of global fixed heap
number of large i/o buffers	size of process object heap
number of locks	size of shared class heap
number of mailboxes	size of unilib cache
number of messages	txn to pss ratio
number of open databases	

- `sp_monitorconfig` 出力内のカラムは、次の情報を提供します。
 - `num_free` : 現在使用されていない、利用可能なメタデータ数または補助スキャン記述子数を指定します。
 - `num_active` : キャッシュ内にインストールされた (したがってアクティブな) メタデータ数または補助スキャン記述子数を指定します。
 - `pct_active` : キャッシュされているかアクティブなメタデータ、または補助スキャン記述子の比率をパーセントで指定します。
 - `Max_Used` : サーバを起動してから使用しているメタデータまたは補助スキャン記述子の最大数を指定します。
 - `Reused` : サーバ内におけるインデックス、オブジェクト、またはデータベースの増加分を格納するために、メタデータ記述子が再使用されていたかどうかを指定します。返される値は、Yes、No、または NA (`aux scan descriptors` の数などの、再使用メカニズムをサポートしていない設定パラメータの場合) です。
- 記述子の適正な数を決定する基礎として、`Max_Used` カラム内の値を使用します。最終設定では、約 10 パーセントの追加を行ってください。たとえば、使用されているインデックス用のメタデータ記述子の最大数が 142 の場合、`number of open indexes` 設定パラメータを 157 に設定できます。

- **Reused** カラムが **Yes** の場合、設定パラメータをより高い値に再設定します。記述子を再使用する必要がある場合には、特にオープン・データベースでパフォーマンスの問題が起こることがあります。オープン・データベースにはメタデータが大量に含まれます。これは、オープン・データベースを満杯にするためには、**Adaptive Server** がディスク上のメタデータに何度もアクセスしなければならないことを意味します。また、サーバにスピンロック競合の問題が起こることがあります。スピンロック競合が起きていないかを調べるには、システム・プロシージャ **sp_sysmon** を使用します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。インデックス、オブジェクト、またはデータベースの現在の数を調べる場合は、**sp_countmetadata** を使用します。
- 正確な読み込み値を得るには、通常の **Adaptive Server** のピーク時に **sp_monitorconfig** を実行します。ピーク時に **sp_monitorconfig** を数回実行し、使用中の記述子の実際の最大数を調べることができます。
- **result_tbl_name** では、以下の構文を使用してテーブルを作成します。結果の情報は、すべてこのテーブルに保存されます。結果の情報から標準の出力が返されることはありません。

```
create table table_name(
    Name varchar(35), Num_free int,
    Num_active int, Pct_act char(6),
    Max_Used int, Reuse_cnt int,
    Date varchar(30))
```

- **number of sort buffers** や **txn to pss ratio** などの一部の設定パラメータは、設定したユーザ接続数に応じて異なりますが、**max number of network listeners** のような他の設定パラメータは、エンジンごとに異なります。
- **sp_monitorconfig** の出力では、ユーザ接続とオンライン・エンジンの数に基づいて、**num_free**、**num_active**、**pct_act**、**max_used** のカラムの値を計算します。
- パフォーマンス上の理由から、内部モニタ・カウンタの更新は、同期方式は採らずに実行されます。このため、負荷の高いマルチエンジン **Adaptive Server** は、あまり正確ではない **sp_monitorconfig** 出力の数値をレポートする可能性があります。
- アイドル状態のシステムで、アクティブなロックの数として 0 より大きい値が表示されることがあります。これらの「アクティブ」なロックは予約されており、内部的に使用されます。

パーミッション `sp_monitorconfig` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_configure](#)、[sp_countmetadata](#)、[sp_helpconfig](#)、[sp_helpconstraint](#)、[sp_sysmon](#)

sp_monitor_server

説明 サーバ全体のモニタリング情報が保持されます。

構文 sp_monitor_server [server_name]

パラメータ なし

例 現在のサーバのモニタリング情報を表示します。

```

sp_monitor_server
last_run          current_run          seconds
-----
May 10 2010  4:23PM          May 10 2010  4:23PM          1

(1 row affected)
cpu_busy          io_busy          idle
-----
0(0)-0%          0(0)-0%          21(0)-0%
packets_received  packets_sent      packet_errors
-----
0(0)              0(0)              0(0)
total_read        total_write        total_errors        connections
-----
1743(0)          146(0)          0(0)          1(0)

```

使用法 なし

パーミッション なし

監査 なし

sp_object_stats

説明 ロック競合、ロック待機時間、およびテーブルとインデックスのデッドロック統計を表示します。

構文 `sp_object_stats interval[, top_n[, dbname, objname[, rpt_option]]]`

パラメータ

interval

サンプリングの時間を指定します。これは HH:MM:SS のフォーマットにしてください。たとえば “00:20:00” のように指定します。

top_n

競合の順にレポートされるオブジェクトの数です。デフォルトは 10 です。

dbname

レポートするデータベースの名前です。データベース名が指定されていない場合は、すべてのデータベース内のオブジェクトでの競合がレポートされます。

objname

レポートするテーブルの名前です。テーブル名が指定されている場合、データベース名も指定してください。

rpt_option

レポートの種類を指定します。

- **rpt_locks** は、最も競合が大きいテーブルのロック付与数、待機数、デッドロック数、待機時間をレポートします。デフォルトは **rpt_locks** です。
- **rpt_objlist** は、最高レベルのロック・アクティビティを持つオブジェクトの名前だけをレポートします。

例 **例 1** 上位 10 のサーバワイドなオブジェクトでのロック統計情報をレポートします。

```
sp_object_stats "00:20:00"
```

例 2 **pubtune** データベース内のテーブルについてのみレポートし、最も競合の高いテーブル 5 つを示します。

```
sp_object_stats "00:20:00", 5, pubtune
```

例 3 競合やデッドロックがない場合でも、ロック・アクティビティが最も高かったテーブルの名前のみを表示します。

```
sp_object_stats "00:15:00", @rpt_option = "rpt_objlist"
```

使用法

- `sp_object_stats` は、指定したサンプル時間内にテーブル上で獲得された共有ロック、更新ロック、排他ロックに関するレポートを表示します。次のレポートは `titles` テーブルを示します。

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.

- 値の意味は次のとおりです。

出力ロー	値
Grants (付与)	ロックが即座に付与された回数
Waits (待機)	ロックを必要とするタスクが待機させられた回数
Deadlocks (デッド ロック)	デッドロックが発生した回数
Wait-times (待機時間)	すべてのタスクがロックを待機するために費やしたミリ秒単位の時間
Contention (競合)	タスクが待機しなければならなかった時間、またはデッドロックが発生した時間の割合

- `sp_object_stats` は、次のようにテーブル上の競合が 15% 以上になるとロック・スキームの変更を推奨します。
 - テーブルが全ページ・ロックを使用している場合、データページ・ロックへの変更を推奨します。
 - テーブルがデータページ・ロックを使用している場合、データロー・ロックへの変更を推奨します。
- `sp_object_stats` は、`tempdb..syskstats` という名前のテーブルを作成します。このテーブルは、ストアド・プロシージャが完了しても削除されません。したがって、システム管理者は `Transact-SQL` を使用してクエリを実行できます。

- `sp_object_stats` を実行できるのは、一度に 1 人のユーザだけです。複数のユーザが同時に `sp_object_stats` を実行しようとする、2 番目のコマンドがブロックされるか、または結果が無効になることがあります。
- `tempdb..syslkstats` テーブルは削除され、`sp_object_stats` が実行されるたびに再作成されます。
- `tempdb..syslkstats` の構造は次のとおりです。

カラム名	データ型	説明
<code>dbid</code>	<code>smallint</code>	データベース ID
<code>objid</code>	<code>int</code>	オブジェクト ID
<code>lockscheme</code>	<code>smallint</code>	整数値 1 ~ 3 : 全ページ = 1、データページ = 2、データロー = 3
<code>page_type</code>	<code>smallint</code>	データ・ページ = 0、またはインデックス・ページ = 1
<code>stat_name</code>	<code>char(30)</code>	このローによって表記される統計情報 <code>stat_name</code> カラム内の値は、次の 3 つの部分で構成されます。 <ul style="list-style-type: none"> • 最初の部分は、排他ロックの “ex”、共有ロックの “sh”、更新ロックの “up” です。 • 2 番目の部分は、ページ・ロックの “pg”、ロー・ロックの “row” です。 • 3 番目の部分は、即座に付与されるロックの “grants”、解放される他のロックのために待機しなければならなかったロックの “waits”、デッドロックの “deadlocks”、ロックを獲得するために待機した時間の “waittime” です。
<code>stat_value</code>	<code>float</code>	付与の数、待機またはデッドロック、総待機時間

- テーブル名を指定すると、`sp_object_stats` はその名前ですべてのテーブルを表示します。複数のユーザが指定した名前を所有している場合、これらのテーブルの出力は所有者名ではなくオブジェクト ID を表示します。

パーミッション

`sp_object_stats` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

コマンド [alter table](#)

sp_objectsegment

説明 指定したオブジェクトのパーティション名、セグメント名、および作成日をレポートします。

構文 `sp_objectsegment object_name`

パラメータ *object_name*
目的のオブジェクトの名前です。指定できるオブジェクトは次のとおりです。

- システム・テーブル
- ビュー
- ユーザ・テーブル
- システム・プロシージャ
- デフォルト
- ルール
- トリガ
- 参照制約
- 検査制約
- 拡張タイプ
- 関数
- 計算カラム
- パーティション

例 `authors` テーブルの情報をレポートします。

```
sp_objectsegment authors
Partition_name      Data_located_on_segment      When_created
-----
audind_576002052      default                      Feb  9 2012 11:18AM
```

使用法

パーミッション `sp_objectsegment` は、すべてのユーザが実行できます。

sp_opt_querystats

説明	選択したクエリのパフォーマンス分析をリターンします。
構文	<pre>sp_opt_querystats "query_text" help [, "diagnostic_options" null [, database_name] [, user_name]]</pre>
パラメータ	<p>"query_text" 分析しているクエリのテキストで、引用符で囲まれています。</p> <p>help sp_opt_querystats の構文および使用法に関する情報を表示します。</p> <p>diagnostic_options (省略可能) set オプションを基にした診断パラメータです。「使用法」を参照してください。</p> <p>null sp_opt_querystats では、データベースの名前を指定するために3つのパラメータが必要です。診断オプションが必要ない場合、このパラメータに null を入力し、database_name パラメータに値を指定します。</p> <p>database_name (オプション) クエリが実行されるデータベースの名前です。分析しているクエリに完全修飾テーブルがない場合、このパラメータを使用します。</p> <p>user_name (省略可能) database_name パラメータによって指定されたデータベース内でクエリを実行するユーザ名。このユーザがデータベースに既に存在する場合、sp_opt_querystats を実行するログインには、そのデータベースで setuser コマンドを実行するパーミッションが必要です。</p>
例	<p>例 1 pubs2 データベースで select コマンドを分析します。</p> <pre>sp_opt_querystats 'select * from pubs2.dbo.authors'</pre> <p>例 2 pubs2 データベースで select コマンドを分析し、set showplan、set statistics io、set option show、set statistics plancost on などの set コマンドの有効化に基づく情報を含めます。</p> <pre>sp_opt_querystats 'select * from pubs2.dbo.authors', 'showplan,statio,option_show, plancost'</pre>
使用法	<ul style="list-style-type: none"> クエリを実行するには、sp_opt_querystats の exec コマンドを含める必要があります。

- `sp_opt_querystats` を異なるユーザとして実行するには、`exec immediate` コマンドに `setuser` コマンドを含めるか、出力クエリのコンテキストに含めます。
- 結果セットを戻すには、`sp_query_stats` の `showdata` コマンドを含める必要があります。
- `set quoted_identifier on` の発行後、`sp_opt_querystats` オプションを引用符で囲みます。次に例を示します。

```
sp_opt_querystats 'select "col" from "MYTABLE"', 'all','DB'
```

- `diagnostic_option` は、次のいずれかです。

diagnostic_option	set オプション	注意
statio	set statistics io on	
stattime	set statistics time on	
showplan	set showplan on	
missingstats	set option show_missing_stats long	
resource	set statistics resource on	
switches	show switches	
option_show_long	set option show long	option_show_long および option_show を同時に指定することはできません。
option_show	set option show on	
showdata	set nodata on	showdata を含める場合、set nodata on は実行されません。
plancost	set statistics plancost on	exec または allexec オプションを指定した場合のみ使用できる。
exec	set noexec on	exec を含める場合、set noexec on は実行されません。
allrows_mix	set plan optgoal allrows_mix	allrows_mix、allrows_oltp、および allrows_dss を同時に指定することはできません。
allrows_oltp	set plan optgoal allrows_oltp	
allrows_dss	set plan optgoal allrows_dss	
diagmode	拡張進行情報を返します。	
all	最初の 7 つのオプションを有効にする	all および allexec は、他のパラメータと組み合わせることはできません。また、同時に指定することはできません。 allexec オプションには、all オプションが含まれます。
allexec	最初の 7 つのオプションを有効にする	

- 複数のオプションを含めるか、all キーワードを指定する場合、オプションのリストを引用符で囲む必要があります。
- オプションなしに `sp_opt_querystats` を実行するのは、all オプションを指定して実行するのと同じです。

パーミッション

sp_opt_querystats は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

sp_optgoal

説明	ユーザ定義の最適化目標を作成し、目標に含める一連のアクティブな基準を定義します。
構文	<code>sp_optgoal 'goal_name', action</code>
パラメータ	<p><i>goal_name</i> 作成する目標の名前です。</p> <p><i>action</i> <code>sp_optgoal</code> で実行する動作です。次のいずれかを指定します。</p> <ul style="list-style-type: none"> • <code>show null no action</code> — 目標の内容を表示します。 • <code>save</code> — 新しい目標を作成するか、既存の目標を更新します。 • <code>delete</code> — 目標を削除します。
例	<p>例 1 現在のセッションに対して次の目標を設定します。</p> <pre>SET PLAN OPTLEVEL ase_current SET PLAN OPTGOAL allrows_mix SET HASH_JOIN 1</pre> <p>次のコマンドで、これらの設定を <code>goal_1</code> という名前の目標に保存します。</p> <pre>sp_optgoal 'goal_1', 'save'</pre> <p>次のコマンドを実行すると、現在のセッションに <code>goal_1</code> の設定が使用されます。</p> <pre>set plan optgoal goal_1</pre> <p>例 2 <code>goal_1</code> を削除します。</p> <pre>sp_optgoal 'goal_1', 'delete'</pre>
使用法	<ul style="list-style-type: none"> • <i>goal_name</i> の長さは 12 文字までです。 • パラメータを指定しないで <code>sp_optgoal</code> を実行すると、ユーザが定義したオプティマイザの目標のリストが表示されます。
パーミッション	<p>目標を作成したり削除したりするには、システム管理者またはシステム・セキュリティ担当者あることが要求されます。ただし、作成された目標はすべてのユーザが使用できます。</p> <p><code>sp_optgoal 'show'</code> は、すべてのユーザが実行できます。</p>

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

sp_options

説明	オプションの値を表示します。
構文	<pre>sp_options [[show help [, option_name category_name null [, dflt non_dflt null [, spid]]]]]]</pre>
パラメータ	<p>show カテゴリーに応じてグループ化されたすべてのオプションの現在値とデフォルト値を示します。指定されたオプション名を持つ <code>sp_options</code> を発行すると、個別オプションの現在値とデフォルト値が表示されます。セッション ID と、デフォルト設定でのオプションまたはデフォルト以外の設定でのオプションの表示を指定できます。</p> <p>help 使用方法に関する情報を表示します。パラメータなしの <code>sp_options</code> を発行しても同じ結果が得られます。</p> <p>option_name オプションの名前です。</p> <p>category_name オプションのカテゴリーです。</p> <p>null 設定を表示するオプションを示します。</p> <p>dflt non_dflt null デフォルト設定のオプションを表示するか、デフォルト以外の設定のオプションを表示するかを示します。</p> <p>spid セッション ID を使用して、他のセッションの設定を表示します。</p>
例	<p>例 1 <code>sp_options</code> の使用方法を表示します。</p> <pre>1sp_options 2> go Usage: sp_options [[show help [, <option_name> <category_name> null [, dflt non_dflt null [, <spid>]]]]]</pre> <p>例 2 現在のオプションとデフォルトのオプションの全リストを表示します。</p>

sp_options

```
1> sp_options show
2> go
Category:Query Tuning
name                currentsetting      defaultsetting      scope
-----
optgoal             allows_mix          allows_mix          0
opttimeoutlimit    40                  10                  0
merge_join         1                   1                   4
hash_join          0                   0                   4
nl_join            1                   1                   4
distinct_sorted    1                   1                   4
distinct_sorting   1                   1                   4
distinct_hashing   1                   1                   4
group_sorted       1                   1                   4
group_hashing      1                   1                   4
group_inserting    0                   0                   4
order_sorting      1                   1                   4
append_union_all   1                   1                   4
merge_union_all    1                   1                   4
merge_union_distinct 1                   1                   4
hash_union_distinct 1                   1                   4
store_index        1                   1                   4
bushy_space_search 0                   0                   4
parallel_query     1                   1                   4
replicated_partition 0                   0                   4
ase125_primed     0                   0                   4
index_intersection 0                   0                   4
index_union        1                   1                   4
multi_table_store_ind 0                   0                   4
advanced_aggregation 0                   0                   4
opportunistic_distinct_view 1                   1                   4
repartition_degree 3                   1                   2
scan_parallel_degree 0                   1                   2
resource_granularity 10                  10                  2
parallel_degree    0                   1                   2
statistics_simulate 0                   0                   4
forceplan          0                   0                   7
prefetch           1                   1                   6
metrics_capture    0                   0                   6
process_limit_action quiet                quiet                2
plan_replace       0                   0                   4
plan_exists_check  0                   0                   4
plan_dump          0                   0                   4
plan_load          0                   0                   4
```

```
(39 rows affected)
(return status = 0)
```

例3 個々のオプションの現在の設定とデフォルトの設定を表示します。

```
1> sp_options show, "index_intersection"
2> go
```

name	category	currentsetting	defaultsetting	scope
index_intersection	Query Tuning	0	0	4

```
(1 row affected)
(return status = 0)
```

例4 個々のオプションのデフォルトの設定のみを表示します。

```
1> sp_options show, "index_intersection", dflt
2> go
```

name	defaultsetting
index_intersection	0

```
(1 row affected)
(return status = 0)
```

例5 カテゴリーの現在の設定とデフォルトの設定を表示します。

```
1> sp_options show, "Query Tuning"
2> go
```

Category:Query Tuning

name	currentsetting	defaultsetting	scope
optgoal	allrows_mix	allrows_mix	0
opttimeoutlimit	10	10	0
merge_join	1	1	4
hash_join	0	0	4
nl_join	1	1	4
distinct_sorted	1	1	4
distinct_sorting	1	1	4
distinct_hashing	1	1	4
group_sorted	1	1	4
group_hashing	1	1	4
group_inserting	0	0	4
order_sorting	1	1	4
append_union_all	1	1	4

merge_union_all	1	1	4
merge_union_distinct	1	1	4
hash_union_distinct	1	1	4
store_index	1	1	4
bushy_space_search	0	0	4
parallel_query	1	1	4
replicated_partition	0	0	4
ase125_primed	0	0	4
index_intersection	0	0	4
index_union	1	1	4
multi_table_store_ind	0	0	4
advanced_aggregation	0	0	4
opportunistic_distinct_view	1	1	4
repartition_degree	3	1	2
scan_parallel_degree	0	1	2
resource_granularity	10	10	2
parallel_degree	0	1	2
statistics simulate	0	0	4
forceplan	0	0	7
prefetch	1	1	6
metrics_capture	0	0	6
process_limit_action	quiet	quiet	2
plan replace	0	0	4
plan exists check	0	0	4
plan dump	0	0	4
plan load	0	0	4

(39 rows affected)
(return status = 0)

例 6 クエリ・チューニング・カテゴリーのデフォルト設定を表示します。

```
1> sp_options show, "Query Tuning", dflt
2> go
```

Category:Query Tuning

name	defaultsetting
optgoal	allows_mix
opttimeoutlimit	10
merge_join	1
hash_join	0
nl_join	1
distinct_sorted	1
distinct_sorting	1
distinct_hashing	1


```

group_sorted 1
group_hashing 1
group_inserting 0
order_sorting 1
append_union_all 1
merge_union_all 1
merge_union_distinct 1
hash_union_distinct 1
store_index 1
bushy_space_search 0
parallel_query 1
replicated_partition 0
ase125_primed 0
index_intersection 0
index_union 1
multi_table_store_ind 0
advanced_aggregation 0
opportunistic_distinct_view 1
repartition_degree 1
scan_parallel_degree 1
resource_granularity 10
parallel_degree 1
statistics simulate 0
forceplan 0
prefetch 1
metrics_capture 0
process_limit_action quiet
plan replace 0
plan exists check 0
plan dump 0
plan load 0

(39 rows affected)
(return status = 0)

```

例7 クエリ・チューニング・カテゴリーのデフォルト以外に設定されているオプションを表示します。

```

1> sp_options show, "Query Tuning", non_dflt
2> go

Category:Query Tuning

name                currentsetting      defaultsetting
-----
repartition_degree  3                   1

```

```
scan_parallel_degree 0          1
parallel_degree      0          1
```

```
(3 rows affected)
(return status = 0)
```

例 8 クエリ・チューニング・カテゴリーのオプションを表示します。

```
1> sp_options, show, null
2> go
```

Category:Query Tuning

name	currentsetting	defaultsetting	scope
optgoal	allrows_mix	allrows_mix	0
opttimeoutlimit	10	10	0
merge_join	1	1	4
hash_join	0	0	4
nl_join	1	1	4
distinct_sorted	1	1	4
distinct_sorting	1	1	4
distinct_hashing	1	1	4
group_sorted	1	1	4
group_hashing	1	1	4
group_inserting	0	0	4
order_sorting	1	1	4
append_union_all	1	1	4
merge_union_all	1	1	4
merge_union_distinct	1	1	4
hash_union_distinct	1	1	4
store_index	1	1	4
bushy_space_search	0	0	4
parallel_query	1	1	4
replicated_partition	0	0	4
asel25_primed	0	0	4
index_intersection	0	0	4
index_union	1	1	4
multi_table_store_ind	0	0	4
advanced_aggregation	0	0	4
opportunistic_distinct_view	1	1	4
repartition_degree	3	1	2
scan_parallel_degree	0	1	2
resource_granularity	10	10	2
parallel_degree	0	1	2
statistics simulate	0	0	4

```

forceplan                0                0                7
prefetch                 1                1                6
metrics_capture          0                0                6
process_limit_action     quiet           quiet            2
plan replace             0                0                4
plan exists check        0                0                4
plan dump                 0                0                4
plan load                 0                0                4
(39 rows affected)
(return status = 0)

```

例9 クエリ・チューニング・カテゴリーのデフォルト設定のリストを表示します。

```

1> sp_options show, null, dflt
2> go
Category:Query Tuning

```

name	defaultsetting
optgoal	allrows_mix
opttimeoutlimit	10
merge_join	1
hash_join	0
nl_join	1
distinct_sorted	1
distinct_sorting	1
distinct_hashing	1
group_sorted	1
group_hashing	1
group_inserting	0
order_sorting	1
append_union_all	1
merge_union_all	1
merge_union_distinct	1
hash_union_distinct	1
store_index	1
bushy_space_search	0
parallel_query	1
replicated_partition	0
ase125_primed	0
index_intersection	0
index_union	1
multi_table_store_ind	0
advanced_aggregation	0
opportunistic_distinct_view	1

```

repartition_degree          1
scan_parallel_degree        1
resource_granularity        10
parallel_degree             1
statistics_simulate         0
forceplan                   0
prefetch                    1
metrics_capture             0
process_limit_action        quiet
plan_replace                0
plan_exists_check           0
plan_dump                   0
plan_load                    0

```

```

(39 rows affected)
(return status = 0)

```

例 9 クエリ・チューニング・カテゴリのデフォルト以外に設定されているオプションを表示します。

```

1> sp_options show, null, non_dflt
2> go

```

Category:Query Tuning

name	currentsetting	defaultsetting
repartition_degree	3	1
scan_parallel_degree	0	1
parallel_degree	0	1

```

(3 rows affected)
(return status = 0)

```

例 10 sp_options が認識しないパラメータを入力すると、次のメッセージが表示されます。

```

1> sp_options show, "incorrect option"
2> go

```

```

Msg 19615, Level 16, State 1:
Procedure 'sp_options', Line 436:
No option or category matching 'incorrect option' is
found.Valid categories are:
category
-----
Query Tuning
(1 row affected)
(return status = 1)

```

例 11 適切な使用法を表示するには、以下を実行します。

```
1> sp_options help
2> go

Usage:
sp_options [ [show | help
              [, <option_name>|<category_name>|null
              [, dflt | non_dflt | null
              [, <spid>] ] ] ] ]
```

使用法

sp_options を使用して、次のオプションの設定を表示します。

- set plan dump / load
- set plan exists check
- set forceplan
- set plan optgoal
- set [optCriteria]
- set plan opttimeoutlimit
- set plan replace
- set statistics simulate
- set metrics_capture
- set prefetch
- set parallel_degree number
- set process_limit_action
- set resource_granularity number
- set scan_parallel_degree number
- set repartition_degree number

パーミッション

sp_options は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

sp_passthru

説明	コンポーネント統合サービスのみ — ユーザがリモート・サーバに SQL コマンド・バッファを渡せるようにします。
構文	<code>sp_passthru server, command, errcode, errmsg, rowcount [, arg1, arg2, ... argn]</code>
パラメータ	<p>server SQL コマンド・バッファを渡すリモート・サーバの名前です。このサーバは、サポートされたローカルでないサーバ・クラスに属している必要があります。</p> <p>command SQL コマンド・バッファです。255 文字まで格納できます。</p> <p>errcode リモート・サーバが返すエラー・コードです (エラーコードがある場合)。リモート・サーバでエラーが発生しなければ、0 の値が返されます。</p> <p>errmsg リモート・サーバが返すエラー・メッセージです。1024 文字まで格納できます。このパラメータは、errcode がゼロ以外の数字のときだけ設定されます。ゼロの場合は、NULL が返されます。</p> <p>rowcount コマンド・バッファの最後のコマンドの影響を受けたローの数です。コマンドが insert、delete、または update の場合は、何も返されなくてもこの値は影響されるローの数を表します。最後のコマンドがクエリの場合は、この値は外部サーバから返されるローの数になります。</p> <p>arg1... argn コマンド・バッファの最後のコマンドから返された最後のローから、結果を受け取ります。arg パラメータは、250 個まで指定することができます。どのパラメータも、出力パラメータとして宣言されている必要があります。</p>
例	出力パラメータ @oradate で Oracle サーバからの日付を返します。Oracle のエラーが発生した場合は、エラー・コードが @errcode に格納され、対応するメッセージが @errmsg に格納され、 @rowcount が 1 に設定されます。

```
sp_passthru ORACLE, "select date from dual", @errcode output,  
@errmsg output, @rowcount output, @oradate output
```

使用法

- `sp_passthru` を使用すると、ユーザはリモート・サーバに SQL コマンド・バッファを渡すことができます。渡される 1 つまたは複数の SQL 文の構文は、バッファを受け取るサーバ・クラスにネイティブな構文であることが想定されています。変換や解釈は行われません。リモート・サーバから返される結果は、オプションで出力パラメータに組み込むことができます。

コンポーネント統合サービスがインストールされ設定されているときのみ、`sp_passthru` を使用します。

- コマンド・バッファには、複数のコマンドを入れることができます。一部のサーバ・クラスでは、コマンドをセミコロンで区切る必要があります。パススルー・モードにおけるクエリ・バッファの処理の詳細については、『コンポーネント統合サービス・ユーザーズ・ガイド』を参照してください。

リターン・パラメータ

- 出力パラメータ `arg1 ... argn` は、コマンド・バッファの最後のコマンドから返された最後のローの、対応するカラムの値に設定されます。パラメータの位置によって、パラメータがどのカラムの値を持つかが決定されます。`arg1` はカラム 1 の値を受け取り、`arg2` はカラム 2 の値を受け取り、以下 `arg3 ~ argx` は `column3 ~ columnx` を受け取ります。
- 指定したオプション・パラメータの数が、返されるカラムの数よりも少ない場合には、パラメータに入らないカラムは無視されます。カラムの数よりもパラメータの数が多い場合は、カラムで使わないパラメータは NULL が設定されます。
- 出力パラメータのデータ型とそれぞれのカラムとの間で、データ型の変換が行われます。データ型が類似し、暗黙的な変換が可能な場合は、正常に変換されます。暗黙的な変換については、『リファレンス・マニュアル：ビルディング・ブロック』の「[第 2 章 Transact-SQL 関数](#)」を参照してください。また、パススルー・モード時に各サーバ・クラスが提供するデータ型の意味については、『コンポーネント統合サービス・ユーザーズ・ガイド』を参照してください。

パーミッション

`sp_passthru` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_autoconnect](#)、[sp_remotesql](#)

sp_password

説明	Adaptive Server 15.7 以降では、このシステム・プロシージャは推奨されていません。Adaptive Server のログイン・アカウントのパスワードを追加または変更するには、 <code>create login</code> および <code>alter login</code> コマンドを使用します。
構文	なし
パラメータ	なし
使用法	なし

sp_passwordpolicy

説明	<p><code>sso_role</code> を持つユーザがログインとパスワード・ポリシーのオプションを設定するときを使用できるインタフェースです。</p>
構文	<p>新しいパスワード複雑性オプションを指定、削除、リスト表示するには、次の構文を使用します。</p> <pre>sp_passwordpolicy {"set" "clear" "list"}, <i>policy_option</i>, <i>option_value</i></pre> <p>パスワード複雑性オプションを確認するには、次の構文を使用します。</p> <pre>sp_passwordpolicy 'validate password options'</pre> <p>ネットワーク・ログイン・パスワードを暗号化するための非対称キー・ペアを生成するには、次の構文を使用します。</p> <pre>sp_passwordpolicy "regenerate keypair"</pre> <p>パスワードを期限切れにするには、次の構文を使用します。</p> <pre>sp_passwordpolicy "expire role passwords", "[rolename wildcard]" sp_passwordpolicy "expire login passwords", "[login_name wildcard]" sp_passwordpolicy "expire stale role passwords", "datetime" sp_passwordpolicy "expire stale login passwords", "datetime"</pre> <p>すべてのコマンド、オプション、および値の簡単な説明を表示するには、次の構文を使用します。</p> <pre>sp_passwordpolicy "help"</pre>
パラメータ	<p>set</p> <p>オプションの値を設定します。<code>set</code> を使用する場合は、<i>policy_option</i> を指定する必要があります。</p> <p>clear</p> <p><code>master.dbo.sysattributes</code> テーブルで指定したオプションのローを削除します。ポリシー・オプションが指定されていない場合、<code>clear</code> によって、<code>sysattributes</code> テーブル内のすべてのオプション・ローが削除されます。<code>clear</code> を使用する場合は、<i>policy_option</i> を指定する必要があります。</p> <p>list</p> <p>指定したオプションの値をリスト表示します。<code>list</code> を使用する場合は、<i>policy_option</i> を指定する必要があります。</p>

policy_option, option_value

文字列または (varchar)。set、clear、list のオプション・パラメータで、option_value がその値です。

有効な <i>policy_option</i> の値	説明
allow password downgrade	パスワードのダウングレード期間を終了します。パスワード・ダウングレード期間には、パスワードが新しいコードと古いコードの両方で <code>syslogins</code> に保存され、サーバがたとえば <code>Adaptive Server 15.0.2</code> にダウングレードされる場合に備えてユーザ・パスワードを保持できるようにする。
disallow simple passwords	値 1 はこのオプションをオンにし、値 0 はこれをオフにする。
enable last login updates	ログインが発生するたびにタイムスタンプを記録する <code>Adaptive Server</code> 認証のコードを有効または無効にする。各パラメータの動作は次のとおり。 <ul style="list-style-type: none"> “set” – この属性の値を設定する。 “list” – この属性の現在の値を表示する。 “clear” – <code>sysattributes</code> からローを削除する。“clear” は <code>sysattributes</code> からローを削除するが、<code>Adaptive Server</code> を再起動するか、“set” で新しい値を設定するまでは前の設定が引き続き有効である。
expire login	新しいログインが作成されたときや SSO でログイン・パスワードが変更されたときに、そのログインのパスワードを <code>expired</code> (期限切れ) としてマーク付けすることで、当該ユーザの初回ログイン時にパスワードを強制的に変更させる。
keypair regeneration period	RSA キー・ペアの再生成期間を示す。オプションの値: { ([<i>keypair regeneration frequency</i>], <i>datetime of first generation</i>) (<i>keypair regeneration frequency</i> , [<i>datetime of first generation</i>]) }
<i>keypair regeneration frequency</i>	RSA キー・ペアの再生成頻度。有効な値の範囲は 1 ~ 8,760 時間。デフォルト値は NULL で、この場合、キー・ペアは 24 時間ごとに再生成される。次を使用して、期間のフォーマット指定子を指定する。 <ul style="list-style-type: none"> 'T*M' – 期間を分で表します。たとえば 2 分の場合 “T2M” のようにアスタリスク (*) を数字で置き換える。 'H' – 期間を時間で表す。 'D' – 期間を日数で表します。他の形式を指定しない場合、これがデフォルト値になる。 'W' – 期間を週数で表す。 'M' – 期間を月数で表す。 'Y' – 期間を年数で表す。

有効な <i>policy_option</i> の値	説明
<i>datetime of first generation</i>	<p>キー・ペアが最初に生成される日付と時刻。 <i>datetime of first generation</i> の値に時刻のみを指定すると、RSA キー・ペアの再生成が 24 時間以内の該当する時刻にスケジュールされます。次に従います。</p> <ul style="list-style-type: none"> • <i>datetime of first generation</i> を指定する場合 — 指定した時刻が過ぎると、Adaptive Server は新しい RSA キー・ペアを即座に作成しする。そうでない場合、Adaptive Server は、指定した時刻まで待機する。 • <i>datetime of first generation</i> を指定しない場合 — 最新の RSA キー・ペアが生成された時刻に <i>keypair regeneration period</i> の値を加算して得られた時刻に、Adaptive Server は新しい RSA キー・ペアを再生成する。ただし、この算出された時刻がすでに過ぎている場合は、即座に新しい RSA キー・ペアを再生成する。 <p>最新のキー・ペアが生成された時刻および <i>keypair regeneration period</i> の値にもとづいて、キー・ペアが次に生成されます。</p> <p>注意 <i>keypair regeneration frequency</i> と <i>datetime of first generation</i> の値を同時に NULL に設定することはできない。</p>
keypair error retry [wait count]	<p>キー・ペアの生成が失敗した後の再生成について、さまざまな設定を指定する。</p> <ul style="list-style-type: none"> • <i>wait</i> — キー・ペアを再生成する前に失敗後に待機する時間を指定します。 • <i>count</i> — 失敗後に Adaptive Server がキー・ペアを再生成する試行回数を指定します。
maximum failed logins	<p>アカウントがロックされるまでの、セッションで許可されるログイン失敗の最大回数を示す。</p>
min alpha in password	<p>パスワード内の英字の最小数を示します。</p>
min digits in password	<p>パスワードに許可される最小桁数を示します。</p>
min lower char in password	<p>パスワードに許可される小文字の最小数を示します。</p>
min special char in password	<p>パスワードに許可される特殊文字の最小数を示します。</p>
min upper char in password	<p>パスワードで許可される大文字の最小数を示す。</p>
minimum password length	<p>パスワードの最短の長さを示す。</p>
password exp warn interval	<p>パスワード有効期限の警告間隔を日数で示します。</p>
systemwide password expiration	<p>システム全体のパスワード有効期限を日数で示します。</p>

有効な <i>policy_option</i> の値	説明
unique keypair per session	<p>各ユーザに対するユニーク・キーの生成についての設定を指定します。</p> <ul style="list-style-type: none"> • 1 – ユーザ接続ごとに新しいキー・ペアを生成するよう指定する。 • 0 – すべての接続が同じ RSA キー・ペアを共有するよう指定する。 <p>注意 sp_configure "net password encryption reqd" を "3" に設定した場合は、パスワードのセキュリティを確保するためにセッションごとのユニークなキー・ペアは必要ないので、このパスワード・ポリシー・オプションは無視される。</p>
"expire login passwords", "[login_name wildcard]"	<p>ログイン・パスワード、すべてのログイン、またはワイルドカードのパターンと一致するログインを期限切れにします。master データベース・カタログ syslogins のステータス・カラムは、ステータス・ビット LOGIN_EXPIRED (0x4) で更新され、パスワードが無効になっていることを示します。</p>
"expire role passwords", "[rolename wildcard]"	<p>1 つの役割、すべての役割、またはワイルドカードのパターンと一致する役割のパスワードを期限切れにします。master データベース・カタログ sysssrvroles のステータス・カラムは、ステータス・ビット ROLE_EXPIRED (0x4) で更新され、パスワードが無効になっていることを示します。</p>
"expire stale login passwords", "datetime"	<p>指定された日時を過ぎても変更されていないログイン・パスワードを期限切れにします。master データベース・カタログ syslogins のステータス・カラムは、ステータス・ビット LOGIN_EXPIRED (0x0004) で更新され、パスワードが無効になっていることを示します。datetime の値の入力方法については、Adaptive Server 15.0 の『リファレンス・マニュアル：ビルディング・ブロック』の「第 1 章 システム・データ型とユーザ定義データ型」で「日付および時刻データの入力」を参照してください。</p>
"expire stale role passwords", "datetime"	<p>指定の日時を過ぎても変更されていない役割パスワードを期限切れにします。master データベース・カタログ sysssrvroles のステータス・カラムは、ステータス・ビット ROLE_EXPIRED (0x4) で更新され、パスワードが無効になっていることを示します。</p>

"regenerate keypair"

非対称キー・ペアを生成してネットワーク・ログイン・パスワードの暗号化で使えるようにします。このオプションではカタログは更新されず、アクションはメモリ・フィールド内のみで処理されます。

'validate password options'

長さや有効期限など、設定されたパスワード複雑性オプションの値に含まれるエラーや矛盾をレポートします。結果は表形式でレポートされ、各ローは検証手順、検証の結果、実行された検証テストを表しています。結果は、合格、不合格、または該当なし (NA) です。検証テストのいずれかで不合格になると、リターン・ステータスは1に設定されます。

例

例 1 パスワードの期限が切れるまでのパスワード有効期限の警告間隔を7日に設定します。

```
sp_passwordpolicy 'set',
  'password exp warn interval', '7'
```

例 2 特殊文字の最小文字数のオプションを一覧表示します。

```
sp_passwordpolicy 'list',
  'min special char in password'
```

例 3 disallow simple passwords をデフォルト値にリセットします。

```
sp_passwordpolicy 'clear', 'disallow simple passwords'
```

例 4 以下の例で validate password options の使用法を示します。これらの出力はわかりやすく再フォーマットされており、このプロシージャを実行した場合に画面に表示される出力とは異なります。

- 以下のパスワード複雑性オプションとその値がサーバに保存されます。

```
minimum password length:      8
min alpha in password:        2
min digits in password:       2
min upper char in password:   2
min lower char in password:   2
```

これらのオプションを検証するには、次のように入力します。

```
sp_passwordpolicy 'validate password options'
```

Validation Step	Pass/Fail/NA	Validation Test
min alpha in password	Fail	'min alpha in password' > = 'min

```

upper char in password + 'min
lower char in password'

minimum password length - 1   Pass   'minimum password length' >= 'min
digits in password' + 'min special
char in password' + 'min alpha in
password'

minimum password length - 2   Pass   'minimum password length' >= 'min
digits in password' + min special
char in password' + 'min upper
char in password' + 'min lower
char in password'

maximum password length - 1   Pass   'max password length' >= 'min
digits in password' + 'min
special char in password' + 'min
alpha in password'

maximum password length - 2   Pass   'max password length' >= 'min
digits in password' + 'min special
char in password' + 'min upper
char in password' + 'min lower
char in password'

password exp warn interval    NA     'password exp warn interval' <=
'systemwide password expiration'

```

```

(6 rows affected)
(return status = 1)

```

不合格が 1 つあります。min upper char in password と min lower char in password の合計が min alpha in password の値より大きいので、検証ステップ min alpha in password で不合格となります。

例 5 2007 年 8 月 15 日午前 12 時 01 分から 2 時間ごとにキー・ペアを自動的に再生成するように、ハウスキーピング・タスクを設定します。

```

sp_passwordpolicy "set", "keypair regeneration period",
"2H", "Aug 15 2007 12:01 AM"

```

例 6 キー・ペアの再生成失敗後に Adaptive Server が次の再生成まで待機する時間を設定します。

```

sp_passwordpolicy 'set', 'keypair error retry wait', '10'

```

例 7 失敗後に Adaptive Server がキー・ペアの再生成を試行する回数を 5 に設定します。

```

sp_passwordpolicy 'set', 'keypair error retry count', '5'

```

例 8 すべてのコマンド、オプション、およびその値についての簡単な説明を表示します。


```

sp_passwordpolicy "help"
go
sp_passwordpolicy Usage:sp_passwordpolicy 'help'
sp_passwordpolicy Usage:sp_passwordpolicy command [, option1 [, option2 [,
option3]]]
sp_passwordpolicy commands:
sp_passwordpolicy 'set',
    {'enable last login updates' | 'disallow simple passwords' |
    'min digits in password' | 'min alpha in password' |
    'min special char in password' | 'min upper char in password' |
    'min lower char in password' | 'password exp warn interval' |
    'systemwide password expiration' | 'minimum password length' |
    'maximum failed logins' | 'expire login' |
    'allow password downgrade' | 'keypair error retry wait' |
    'keypair error retry count'},
    'value'
sp_passwordpolicy 'set', 'keypair regeneration period',
    {'regeneration_period' |
    null, 'datetime' |
    'regeneration_period', 'datetime'}
sp_passwordpolicy 'list',
    ['enable last login updates' | 'disallow simple passwords' |
    'min digits in password' | 'min alpha in password' |
    'min special char in password' | 'min upper char in password' |
    'min lower char in password' | 'password exp warn interval' |
    'systemwide password expiration' | 'minimum password length' |
    'maximum failed logins' | 'expire login' |
    'allow password downgrade' |
    'keypair error retry wait' | 'keypair error retry count' |
    'keypair regeneration period']
sp_passwordpolicy 'clear',
    {'enable last login updates' | 'disallow simple passwords' |
    'min digits in password' | 'min alpha in password' |
    'min special char in password' | 'min upper char in password' |
    'min lower char in password' | 'password exp warn interval' |
    'systemwide password expiration' | 'minimum password length' |
    'maximum failed logins' | 'expire login' |
    'keypair error retry wait' |
    'keypair error retry count' | 'keypair regeneration period'}
sp_passwordpolicy 'expire login passwords'[, '{loginame | wildcard}']
sp_passwordpolicy 'expire role passwords'[, '{rolename | wildcard}']
sp_passwordpolicy 'expire stale login passwords', 'datetime'
sp_passwordpolicy 'expire stale role passwords', 'datetime'
sp_passwordpolicy 'regenerate keypair'[, 'datetime']
sp_passwordpolicy 'validate password options'
(return status = 0)

```

例 9 Adaptive Server に保存された以下のオプションを検証します。

```

minimum password length:      8
min digits in password:      2

```

```

min special char in password: 2
min alpha in password: 6
min upper char in password: 3
min lower char in password: 3

```

```
sp_passwordpolicy 'validate password options'
```

Validation Step	Pass/Fail/NA	Validation Test
min alpha in password	Pass	'min alpha in password' >= 'min upper char in password' + 'min lower char in password'
minimum password length-1	Fail	'minimum password length' >= 'min digits in password' + 'min special char in password' + 'min alpha in password'
minimum password length-2	Fail	'minimum password length' >= 'min digits in password' + 'min special char in password' + 'min upper char in password' + 'min lower char in password'
maximum password length-1	Pass	'max password length' >= 'min digits in password' + 'min special char in password' + 'min alpha in password'
maximum password length-2	Pass	'max password length' >= 'min digits in password' + 'min special char in password' + 'min upper char in password' + 'min lower char in password'
password exp warn interval	NA	'password exp warn interval' <= 'systemwide password expiration'

```

(6 rows affected)
(return status = 1)

```

ステップ 2 とステップ 3 に 2 つ不合格があります。min digits in password、min special char in password、min alpha in password の合計が minimum password length の値より大きいので、検証ステップ minimum password length -1 で不合格となります。min digits in password、min special char in password、min upper char in password、min lower char in password の合計が minimum password length の値より大きいので、検証ステップ minimum password length -2 で不合格となります。

例 10 以下に、オプション 'validate password options' の例を示します。出力はわかりやすく再フォーマットされており、このプロシージャを実行した場合に画面に表示される出力とは異なります。

以下のパスワード複雑性オプションとその値がサーバに保存されます。

```

minimum password length:      8
min alpha in password:        2
min digits in password:       2
min upper char in password:   2
min lower char in password:   2

```

```

sp_passwordpolicy 'validate password options'
Validation Step      Pass/Fail/NA  Validation Test
-----
min alpha in password  Fail          'min alpha in password' >= 'min
upper char in password + 'min
lower char in password'

minimum password length - 1  Pass          'minimum password length' >= 'min
digits in password' + 'min special
char in password' + 'min alpha in
password'

minimum password length - 2  Pass          'minimum password length' >= 'min
digits in password' + min special
char in password' + 'min upper
char in password' + 'min lower
char in password'

maximum password length - 1  Pass          'max password length' >= 'min
digits in password' + 'min
special char in password' + 'min
alpha in password'

maximum password length - 2  Pass          'max password length' >= 'min
digits in password' + 'min special
char in password' + 'min upper
char in password' + 'min lower
char in password'

password exp warn interval  NA            'password exp warn interval' <=
'systemwide password expiration'

(6 rows affected)
(return status = 1)

```

これらのオプションを検証するには、次のように入力します。

不合格が 1 つあります。min upper char in password \neq min lower char in password の合計が min alpha in password の値より大きいので、検証ステップ min alpha in password で不合格となります。

Adaptive Server に保存された以下のオプションを検証します。

```

minimum password length:      8
min digits in password:       2
min special char in password:  2
min alpha in password:        6
min upper char in password:    3
min lower char in password:    3

```

```

sp_passwordpolicy 'validate password options'
Validation Step      Pass/Fail/NA  Validation Test
-----
min alpha in password  Pass          'min alpha in password' >= 'min upper
char in password' + 'min lower
char in password'

minimum password length-1  Fail          'minimum password length' >= 'min
digits in password' + 'min special
char in password' + 'min alpha in
password'

minimum password length-2  Fail          'minimum password length' >= 'min
digits in password' + 'min special
char in password' + 'min upper
char in password' + 'min lower
char in password'

maximum password length-1  Pass          'max password length' >= 'min
digits in password' + 'min special
char in password' + 'min alpha in
password'

maximum password length-2  Pass          'max password length' >= 'min
digits in password' + 'min
special char in password' + 'min
upper char in password' + 'min
lower char in password'

password exp warn interval  NA           'password exp warn interval' <=
'systemwide password expiration'

(6 rows affected)
(return status = 1)

```

ステップ 2 とステップ 3 に 2 つ不合格があります。

min digits in password、min special char in password、min alpha in password の合計が minimum password length の値より大きいので、検証ステップ minimum password length -1 で不合格となります。min digits in password、min special char in password、min upper char in password、min lower char in password の合計が minimum password length の値より大きいので、検証ステップ minimum password length -2 で不合格となります。

Adaptive Server に保存された以下のオプションを検証します。

```

minimum password length:      8
min digits in password:      11
min special char in password: 11
min alpha in password:       11
min upper char in password:   1
min lower char in password:   1

```

```
sp_passwordpolicy 'validate password options'
```

Validation Step	Pass/Fail/NA	Validation Test
-----	-----	-----
min alpha in password	Pass	'min alpha in password' >= 'min upper char in password' + 'min lower char in password'
minimum password length-1	Fail	'minimum password length' >= 'min digits in password' + 'min special char in password' + 'min alpha in password'
minimum password length-2	Fail	'minimum password length' >= 'min digits in password' + 'min special char in password' + 'min upper char in password' + 'min lower char in password'
maximum password length-1	Fail	'max password length' >= 'min digits in password' + 'min special char in password' + 'min alpha in password'
maximum password length-2	Pass	'max password length' >= 'min digits in password' + 'min special char in password' + 'min upper char in password' + 'min lower char in password'
password exp warn interval	NA	'password exp warn interval' <= 'systemwide password expiration'

```
(6 rows affected)
(return status = 1)
```

必要なパスワード・コンポーネントの合計が、許可されている最大パスワードより大きいため最大パスワード長のテストに不合格という重大なエラーを含め、不合格が3つあります。

Adaptive Server に保存された以下のオプションを検証します。

```
minimum password length:      8
min digits in password:      2
min special char in password: 1
min alpha in password:       4
min upper char in password:   0
min lower char in password:   0
```

```
sp_passwordpolicy 'validate password options'
```

Validation Step	Pass/Fail/NA	Validation Test
min alpha in password	Pass	'min alpha in password' >= 'min upper char in password' + 'min lower char in password'
minimum password length-1	Pass	'minimum password length' >= 'min digits in password' + 'min special char in password' + 'min alpha in password'
minimum password length-2	Pass	'minimum password length' >= 'min digits in password' + 'min special char in password' + 'min upper char in password' + 'min lower char in password'
maximum password length-1	Pass	'max password length' >= 'min digits in password' + 'min special char in password' + 'min alpha in password'
maximum password length-2	Pass	'max password length' >= 'min digits in password' + 'min special char in password' + 'min upper char in password' + 'min lower char in password'
password exp warn interval	NA	'password exp warn interval' <= 'systemwide password expiration'

```
(6 rows affected)
(return status = 0)
```

使用法

これらの設定には不合格がありません。5つのローすべてが返され、リターン・ステータス 0 がレポートされています。

sp_passwordpolicy の情報は master.dbo.sysattributes テーブルに保存されます。

キー・ペアの再生成

Adaptive Server が新しい RSA キー・ペアを再生成すると、次の生成では、RSA キー・ペアの最後の生成時の式が *keypair regeneration frequency* に設定した値と組み合わせて使用されます。

keypair regeneration period の値は、新しいパスワード・ポリシー・クラスの下にある master..sysattributes に格納されます。

オプションの NULL のデフォルト値は、このローが sysattributes には存在せず、キー・ペアは Adaptive Server の再開時、およびその後 24 時間ごとに再生成されることを示します。

これら 2 つのストアド・プロシージャを使用しても同じ結果が得られます。

```
sp_passwordpolicy 'set', 'keypair regeneration period', NULL [,
    datetime of first generation]
```

```
sp_passwordpolicy 'regenerate keypair' [, datetime of first generation]
```

これらのグローバル変数は、keypair regeneration period からの情報を使用します。

- @@lastkpgendate — キー・ペアが最後に生成された日付と時刻を示します。
- @@nextkpgendate — キー・ペアが次に生成される時を示します。

複雑なログイン・パスワード・チェック

これらの複雑なログイン・パスワード・チェックは、役割パスワードに拡張されます。

- disallow simple passwords
- min digits in password
- min alpha in password
- min special char in password
- min upper char in password
- min lower char in password
- systemwide password expiration

- password exp warn interval
- minimum password length
- maximum failed logins
- expire login

高可用性およびパスワード・ポリシー・オプション

Adaptive Server の高可用性機能によって、プライマリ・サーバおよびセカンダリ・サーバ間のパスワード・ポリシー・オプションが同期化されます。

- disallow simple passwords
- min digits in password
- min alpha in password
- min special char in password
- min upper char in password
- min lower char in password
- systemwide password expiration
- password exp warn interval
- minimum password length
- maximum failed login
- expire login
- keypair regeneration period
- keypair error retry wait
- keypair error retry count

Adaptive Server は「パスワード・ポリシー」の定属性を使用して、プライマリ・サーバおよびセカンダリ・サーバの両方で、keypair regeneration period、keypair error retry wait、および keypair error retry count 以外のこれらの値の不整合性をチェックします。高可用性アドバイザー・チェックは、両方のサーバでこれらの値が同じであれば成功し、異なっていれば失敗します。次に例を示します。


```
sp_companion "MONEY1", do_advisory, 'all'
go
```

Attribute Name	Attrib Type	Local Value	Remote Value	Advisory
expire login	password po	1	0	2
maximum failed	password po	3	5	2
min alpha in pa	assword po	10	12	2

出力の `advisory` カラムの値が 2 になっていますが、これは両サーバの値が一致していないので、ユーザがクラスタ・オペレーションを進めることができないことを示します。

`sp_companion do_advisory` の出力も、両方のサーバでの特定のパスワード・ポリシー・チェックの不整合を示します。

パーミッション

`sp_passwordpolicy` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage security configuration</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>ss0_role</code> が付与されたユーザであることが必要。

監査

`sp_passwordpolicy` の `set` コマンドと `clear` コマンドが監査イベント 115「パスワード管理」を通して監査されます。

監査オプションの “password” は次のアクションを監査します。

- `sp_passwordpolicy 'set', 'option_name', 'option_value'`
- `sp_passwordpolicy 'clear', 'option_name'`
- `sp_passwordpolicy 'expire login passwords'`
- `sp_passwordpolicy 'expire stale login passwords'`
- `sp_passwordpolicy 'regenerate keypair'`
- `sp_passwordpolicy 'expire role passwords'`
- `sp_passwordpolicy 'expire stale role passwords'`

“password” 監査オプションは、`AUD_EVT_PASSWORD_ADMIN(115)` 監査イベントを生成する RSA キー・ペア再生成期間の管理も監査します。

sp_pciconfig

説明

Java PCI Bridge を管理します。引数およびディレクティブの有効化と無効化、設定値の変更、設定値のレポートを行います。

注意 Sybase 製品の保守契約を結んでいるサポート・センタの指示がないかぎり、`sp_pciconfig` を使用してその他の引数またはディレクティブを変更しないでください。

構文

```
sp_pciconfig {
    disable { directive | argument } |
    enable { directive | argument } |
    list { list_type [ , formatted ] | units | units, units_type[ , formatted ] } |
    report { directive[ , formatted ] |
        directive, args[ , formatted ] |
        argument[ , formatted ] } |
    update { number_arg, old_value new_value }
```

パラメータ

disable

指定したディレクティブまたは引数を無効にします。

directive

有効なディレクティブの名前です。

argument

有効な引数の名前です。

enable

指定したディレクティブまたは引数を有効にします。

list

たとえば、`sp_pciconfig "list", "directive"` や `sp_pceiconfig "list", "enabled"` のように、関連する引数のグループをリスト表示します。また、`sp_pciconfig "list", "units", "switch"` のように、特定の型のすべての引数をリスト表示します。

list_type

リストのタイプを指定します。値は次のとおりです。

- `directives` — ディレクティブのリスト
- `enabled` — 有効な引数のリスト
- `disabled` — 無効な引数のリスト
- `argnames` — 引数名のリスト

formatted

表示されたリストを読みやすくフォーマットするように指定します。

注意 フォーマットされたレポートでは、読みやすさを重視してワイド・カラムがトランケートされる場合があります。また、カラムの見出しがオーバーライドされ、実際のテーブル・カラム名と一致しない可能性があります。出力を解析する場合やデータのトランケーションが適切でない可能性がある場合は、レポートをフォーマットしないでください。

units

list とともに使用すると、現在使用している *units_type* のリストを生成します。

report

指定した引数に基づいてレポートを作成します。通常は、引数の現在値と引数が有効になっているかどうかを確認するため、引数のレポートの生成に使用されます。ディレクティブまたはその引数のレポートの生成にも使用できます。

directive

指定したディレクティブ内のすべての引数を指定します。

update

units = number の引数の数値を変更します。**units = switch** の引数では使用できません。

number_arg

units = number の引数です。

old_value

number_arg_name の現在の値です。

new_value

number_arg_name の新しい値です。

使用法

ディレクティブの有効と無効は切り替えられます。ディレクティブが次の場合：

- **Enabled** — Adaptive Server は各引数の設定値 (有効または無効) を使用します。この値は **sybpcidb** に保存されています。
- **Disabled** — Adaptive Server は、各引数の設定値 (有効または無効) を無視し、各引数のベース値が **sybpcidb** に保持されているにもかかわらず、ディレクティブのすべての引数を無効として扱います。

引数は、有効／無効を個別に切り替えることができます。sp_pciconfig デイレクティブの引数は以下の型です。

- **switch** — これらの引数は、機能のオンまたはオフを切り替えられます。たとえば、ロギング用の引数がある状態になっている場合は、ログ・ファイルが生成され、無効になっている場合は、ログ・ファイルが生成されません。
- **string** — これらの引数は文字列か数字用で、文字列として扱われます。文字列の引数を有効にすると、Adaptive Server は設定値を使用します。文字列の引数を無効にすると、Adaptive Server は設定値を無視してデフォルト値を使用します。設定値とデフォルト値は同じ場合と異なる場合があります。

表 1-26: sp_pciconfig の設定ディレクティブ

ディレクティブ	説明
PCI_BRIDGE_X_OPT	PCI Bridge 設定パラメータ
PCI_BRIDGE_LOGOPT	プラグイン <code>diagserver</code> のレポート機能
PCI_BRIDGE_INSTR	PCI Bridge インストールメンターションの設定

表 1-27: PCI_BRIDGE_X_OPT の引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pci_xopt_maxthreads	number	1056	有効	最大使用可能 PCI Bridge PLB 制御スレッド。
pci_xopt_event_scheduling	number	0	有効	デフォルトの PCI Bridge スケジューリング。
pci_xopt_failover_engine	number	-1	有効	スロットがフェール・オーバーするデフォルト・エンジン。
pci_xopt_runtime_alloc_escape	number	1	有効	PC Bridge の最大メモリ・アロケーション・ユニットを超えるメモリ割り当てが要求されたときにランタイム・エスケープを許可します。
pci_xopt_slotring_cycle	number	-1	有効	PCI Bridge スロットリング・ウォッシングを無効にします。
pci_xopt_slotring_wash_th	number	76	有効	デフォルトの PCI Bridge スロットリング・ウォッシング・スレッシュホールド・パーセンテージ。

表 1-28: PCI_BRIDGE_LOGOPT の引数

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pci_logopt_asehi	switch	なし	無効	PCI Bridge ASE ホスト・インタフェースのディスパッチ・ロギング。
pci_logopt_jst	switch	なし	無効	PCI Bridge ジョブ・スケジューラのタスク・ディスパッチ・ロギング。
pci_logopt_jvm	switch	なし	無効	PCI Bridge JVM のディスパッチ・ロギング。
pci_logopt_omni	switch	なし	無効	PCI Bridge OMNI のディスパッチ・ロギング。
pci_logopt_pci	switch	なし	無効	一般的な PCI Bridge のロギング (probe [pci/pca])。
pci_logopt_runtime	switch	なし	無効	PCI Bridge ランタイムのディスパッチ・ロギング。
pci_logopt_xml	switch	なし	無効	PCI Bridge XML のディスパッチ・ロギング。

表 1-29: PCI_BRIDGE_INSTR arguments

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
BRIDGE	number	1	無効	完全なインストルメンテーションを強制します (ノイズ)。
CELL	number	1	無効	すべての CELL 同期をレポートに強制します。
JAVA	number	1	無効	Java 関連のすべてのエントリをレポートに強制します。
JCS	number	1	無効	すべての JCS エントリをレポートに強制します。
JDBC	number	1	無効	すべての JDBC エントリをレポートに強制します。
JVMHOST	number	1	無効	すべての ASE JVM ホスト API エントリをレポートに強制します。
JVMJNI	number	1	無効	すべての JVM JNI 外部 extries をレポートに強制します。
PCIS	number	1	無効	すべての PCI サービス・コードをレポートに強制します。
PLB	number	1	無効	すべての PLB コードをレポートに強制します。
SLOTRING	number	1	無効	すべての「スロットリング」コードをレポートに強制します。
SYNC	number	1	無効	すべての SYNChronization コードをレポートに強制します。
TPM	number	1	無効	すべての TPM コードをレポートに強制します。
fetch_classdata	number	1	有効	すべての fetch_classdata ヒットをレポートに強制します。

引数	単位のタイプ	デフォルト値	デフォルトの状態	説明
pcis_service	number	2	無効	すべての pcis_service ヒットをフリーズに強制します。

パーミッション sp_pciconfig のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage server configuration 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles – 現在のアクティブな役割 • Keywords or options – NULL • Previous value – NULL • Current value – NULL • Other information – すべての入力パラメータ • Proxy information – set proxy が有効な場合は元のログイン名

参照 [ストアド・プロシージャ sp_jreconfig](#)

sp_placeobject

説明	テーブルまたはインデックスに対する今後の領域の割り当てを、特定のセグメントで行います。
構文	<code>sp_placeobject segname, objname</code>
パラメータ	segname テーブルまたはインデックスを配置するセグメントの名前です。 objname <i>segname</i> で指定したセグメントに今後の領域を割り当てるテーブルまたはインデックスの名前です。インデックス名は“ <i>tablename.indexname</i> ”という形式で指定します。
例	例 1 “segment3” という名前のセグメント上の <code>authors</code> テーブルに対して、後続の領域をすべて割り当てます。 <pre>sp_placeobject segment3, authors</pre> 例 2 <code>indexes</code> という名前のセグメント上にある <code>employee</code> テーブルの <code>employee_nc</code> という名前のインデックスに後続の領域をすべて割り当てます。 <pre>sp_placeobject indexes, 'employee.employee_nc'</pre>
使用法	<ul style="list-style-type: none">システム・テーブルに対する今後の領域の割り付け位置は変更できません。特定のセグメントにテーブルまたはインデックスを配置しても、既存のテーブル・データやインデックス・データの位置は変わりません。今後の領域の割り付けだけが変更されます。これには、テーブルやインデックスの既存のすべてのパーティションと、新しいパーティションにセグメントが指定されていない場合に後から追加された新しいパーティションが含まれます。テーブルまたはインデックスが使用しているセグメントを変更することによって、複数のセグメント間にデータを分散することができます。クラスタード・インデックスを指定して <code>sp_placeobject</code> を使用した場合、テーブルはインデックスと一緒に移動します。

- `create table` または `create index` を使用してテーブルまたはインデックスを作成するときに、セグメントを指定できます。パーティション定義の一部として、パーティション・レベルでセグメントを指定することもできます。セグメントの指定のないパーティションでは、テーブル/インデックス・レベルで指定されたセグメントが使用されます。テーブル/インデックス・レベルでセグメントを指定しなければ、データはデフォルトのセグメントに配置されます。
- `sp_placeobject` を使用して複数のディスク・フラグメントにまたがるテーブルまたはインデックスを分割した場合、`sp_placeobject` を実行する前に記憶領域として使用されていたフラグメントに常駐していたデータについてのメッセージが、`dbcc` コマンドによって表示されます。それらのメッセージは、無視してかまいません。

パーミッション

`sp_placeobject` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、テーブル所有者か <code>manage database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者、テーブル所有者、または <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド `alter table`、`dbcc`
 システム・プロシージャ `sp_addsegment`、`sp_dropsegment`、`sp_extendsegment`、`sp_helpindex`、`sp_helpsegment`

sp_plan_dbccdb

説明 新しく作成する dbccdb と dbccalt データベースのサイズの推奨値、dbccdb と dbccalt の適切なデバイスのリスト、ターゲット・データベースのキャッシュ・サイズと適切なワーカー・プロセス数を提示します。

構文 sp_plan_dbccdb [dbname]

パラメータ *dbname*
 ターゲット・データベースの名前を指定します。*dbname* を指定しなければ、sp_plan_dbccdb は、master..sysdatabases にあるすべてのデータベースについての推奨値を提示します。

例 **例 1** master データベースの検査に適した dbccdb データベースの作成用に、推奨設定値を返します。このコマンドが実行された時点ですでに dbccdb データベースが存在していたため、既存のデータベースのサイズが比較用に表示されています。

```
sp_plan_dbccdb master

Recommended size for dbccdb database is 50MB (data = 48MB, log = 2MB).
dbccdb database already exists with size 280MB.
Recommended values for workspace size, cache size and process count are:
dbname          scan ws  text ws  cache  comp mem  process count
master          128K    48K     640K   0K      1
```

例 2 サーバ内のすべてのデータベースの検査に適した dbccdb データベースの作成用に、推奨設定値を返します。出力には、圧縮デバイスを使用するアーカイブ・データベースにのみゼロ以外の値がある圧縮メモリ要件が含まれます。このコマンドを実行した時点では、dbccdb データベースは存在していませんでした。

```
sp_plan_dbccdb

Recommended size for dbccdb database is 50MB (data = 48MB, log = 2MB).
dbccdb database already exists with size 280MB.
Recommended values for workspace size, cache size and process count are:
dbname          scan ws  text ws  cache  comp mem  process count
master          128K    48K     640K   0K      1
tempdb          656K   176K    1280K  0K      2
model           64K     48K     640K   0K      1
sybsystemdb     64K     48K     640K   0K      1
sybsystemprocs 1488K   384K     640K   0K      1
sybsecurity     272K    80K    1280K  0K      2
adb             80K     64K    1920K  12M     3
```

例 3 pubs2 の検査に適した dbccdb の作成用に推奨設定値を返します。

```
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

使用方法

- sp_plan_dbccdb は、dbccdb と dbccalt データベースを新しく作成するための適切なサイズを推奨し、新しいデータベースに適したデバイスをリストします。また、ターゲット・データベースに適したワーカー・プロセスの数とキャッシュ・サイズも提示します。
- dbccdb を指定すると、sp_plan_dbccdb は、代替データベース dbccalt の推奨値を提示します。dbccalt を指定すると、sp_plan_dbccdb は、dbccdb の推奨値を提示します。
- sp_plan_dbccdb は、既存の dbccdb や dbccalt データベース用の値についてはレポートしません。既存の dbccdb または dbccalt データベースの設定パラメータを確認するには、[sp_dbcc_evaluatedb](#) を使用します。
- dbccdb のメンテナンスと dbccdb からのレポート生成に使用する dbcc ストアド・プロシージャについては、「[第 4 章 dbcc ストアド・プロシージャ](#)」を参照してください。

パーミッション

sp_plan_dbccdb のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、すべてのユーザがこのプロシージャを実行できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か sa_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド dbcc

システム・プロシージャ [sp_dbcc_evaluatedb](#)

sp_poolconfig

説明 データ・キャッシュ内のメモリ・プールを作成、削除、サイズ変更し、情報を表示します。

構文 既存のキャッシュ内にメモリ・プールを作成したり、プール・サイズを変更する場合：

```
sp_poolconfig cache_name[, "mem_size [P | K | M | G]", "config_poolK"  
[, "affected_poolK"], instance instance_name]
```

プールのウォッシュ・サイズを変更する場合：

```
sp_poolconfig cache_name, "affected_poolK", "wash=size[P|K|M|G]"
```

プールの非同期プリフェッチ率を変更する場合：

```
sp_poolconfig cache_name, "affected_poolK",  
"local async prefetch limit=percent "
```

パラメータ

cache_name

既存のデータ・キャッシュの名前です。

mem_size

作成するメモリ・プールのサイズ、または、指定した I/O サイズの既存のプールの新しい総サイズです。プールの最小サイズは、論理サーバ・ページの 256 倍です。2K 論理ページ・ページ・サーバの場合、最小サイズは 256K です。サイズの単位は、ページなら P、キロバイトなら K、メガバイトなら M、ギガバイトなら G として指定します。デフォルトは、キロバイトです。

config_pool

メモリが割り当てられる、または削除されるメモリ・プール内の I/O サイズです。

有効な I/O サイズは、論理ページ・サイズの倍数で、最大 4 倍までです。

affected_pool

メモリの割り当てを解除する、または 'wash size' および 'prefetch limit' などのプールの属性が変更されるメモリ・プール内で実行される I/O サイズです。affected_pool が指定されなければ、メモリは最小の論理ページ・サイズのメモリ・プールから使われます。

instance_name

クラスタ環境の場合 – 調整するバッファ・プールのインスタンスの名前です。

wash=size

メモリ・プールに対するウォッシュ・サイズ (Adaptive Server がダーティ・ページをディスクに書き込むキャッシュ内の位置) を変更します。

local async prefetch limit=percent

プールのバッファのうち、非同期プリフェッチでキャッシュに読み込まれたが、まだ使用されていないバッファを保持するために使用できるバッファの比率を設定します。プリフェッチ制限値を 0 に設定すると、プールでの非同期プリフェッチは行われなくなります。

例

例 1 10MB の領域を持つデータ・キャッシュ `pub_cache` 内に 16K のプールを作成します。すべての領域は、デフォルトの 2K のメモリ・プールから使用されます。

```
sp_poolconfig pub_cache, "10M", "16K"
```

例 2 `pub_cache` の 64K のプールからの 32K のプールに 16MB の領域を作成します。

```
sp_poolconfig pub_cache, "16M", "32K", "64K"
```

例 3 `pub_cache` の現在の設定をレポートします。

```
sp_poolconfig "pub_cache"
```

例 4 `pub_cache` から 16K のメモリ・プールを削除し、そのプールに割り当てられていたメモリを、すべて 2K のプールに移動します。

```
sp_poolconfig pub_cache, "0K", "16K"
```

例 5 `pubs_cache` 内の 2K のプールのウォッシュ・サイズを 508K に変更します。

```
sp_poolconfig pub_cache, "2K", "wash=508K"
```

例 6 2K のプールの非同期プリフェッチの制限を 15% に変更します。

```
sp_poolconfig pub_cache, "2K", "local async prefetch limit=15"
```

例 7 クラスタ環境の場合 – インスタンス `blade1` のデフォルト・データ・キャッシュ内にサイズが 25MB の 16KB バッファ・プールを作成します。

```
sp_poolconfig 'default data cache', '25M', '16K', 'instance blade1'
```

例 8 クラスタ環境の場合 – インスタンス `blade1` のデフォルト・データ・キャッシュのバッファ・プール設定を表示します。

```
sp_poolconfig 'default data cache', 'instance bladel1'
```

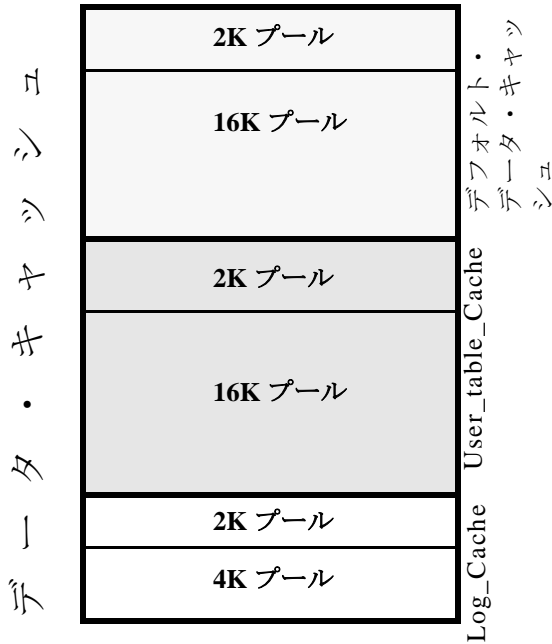
例 9 クラスタ環境の場合 – クラスタ内のすべてのインスタンスの名前付きキャッシュ `c_log` のバッファ・プール設定を表示します。

```
sp_poolconfig c_log
```

使用法

- `sp_cacheconfig` によってデータ・キャッシュを作成すると、すべての領域は論理ページ・サイズのメモリ・プールに割り当てられます。`sp_poolconfig` は、データ・キャッシュを、より大容量の I/O サイズを持つ追加プールに分けます。
- キャッシュ内に大容量 I/O メモリ・プールが存在しない場合、Adaptive Server はキャッシュにバインドされたオブジェクトすべてに対して論理ページ・サイズ単位でデータ・ページの I/O を行います。16K のメモリ・プールは、2K の論理ページ・サイズ・サーバに対して単一の I/O で 8 つのデータ・ページを読み込みおよび書き込みします。
- キャッシュ名と I/O サイズの組み合わせは、必ずユニークにしてください。つまり、`sp_poolconfig` コマンドでは、特定のデータ・キャッシュ内で所定の I/O サイズのプールを 1 つだけ持つことができます。
- 1 つの `sp_poolconfig` コマンドは、同時に 1 つのキャッシュに対してだけアクティブにできます。最初の `sp_poolconfig` コマンドが完了する前に 2 つ目の `sp_poolconfig` コマンドを実行しようとした場合、最初のコマンドが完了するまで 2 つ目のコマンドはスリープします。
- [図 1-3](#) は、次の項目を持つ 2K の論理ページを使用するサーバ上のデータ・キャッシュを示します。
 - 2K プールと 16K プールを持つデフォルト・データ・キャッシュ
 - 2K プールと 16K プールを持つユーザ・キャッシュ
 - 2K プールと 4K プールを持つログ・キャッシュ

図 1-3: デフォルト・キャッシュとユーザ定義キャッシュを持つデータ・キャッシュ



- ページ・サイズ 2K のサーバのデフォルト・データ・キャッシュでは、I/O サイズが最大 16K のプールを作成できます。
- メモリ・プールの最小サイズは、256 論理ページです (たとえば、2K の論理ページ・サイズ・サーバの場合、最小サイズは 512K になります)。メモリを別のプールに転送しても、キャッシュ内のメモリ・プールのサイズを 256 ページ未満には縮小できません。
- 次の 2 つの状況では、512K よりも小さいプールが作成されることがあります。
 - ユーザがプールのサイズをゼロに設定してプールを削除しようとしたが、まだ一部のページが使用されていた場合には、sp_poolconfig はプール・サイズをできるかぎり小さくするとともに、警告メッセージを出力します。プールのステータスは、“Unavailable/deleted” に設定されます。

- ユーザが新しいプールを作成するためにバッファを移動しようとしたが、新しいプールに移動するだけのバッファが利用できない場合には、`sp_poolconfig` はできるだけ多くのバッファを移動しようとします。キャッシュのステータスは、“Unavailable/too small” に設定されます。

いずれの場合も、後でコマンドをリトライすることができます。サーバが再起動されたときも、プールが削除されたり、望ましいサイズに変更されます。

- **Adaptive Server** がアクティブである間はメモリ・プールを作成できます。メモリ・プールを反映させるために再起動する必要はありません。**Adaptive Server** は「空き」バッファ（使用されていないバッファ、またはディスクにまだ書き込まれていない変更を含まないバッファ）だけを移動できます。プールを設定したりそのサイズを変更したりしたとき、**Adaptive Server** はできるだけ多くのメモリをプールに移動し、要求されたサイズと実際のプール・サイズを示す情報メッセージを表示します。**Adaptive Server** を再起動すると、設定されたサイズですべてのプールが作成されます。
- 一部の `dbcc` コマンドおよび `drop table` は論理ページ・サイズ I/O のみを実行します。`dbcc checkstorage` は大容量の I/O を実行できません。`dbcc checkdb` はテーブルに対して大容量の I/O を実行し、インデックスに対して論理ページ・サイズの I/O を実行します。
- ほとんどの場合 **Adaptive Server** では、トランザクション・ログに対して設定された論理ページ・サイズの 2 倍の I/O を使うと、より優れたパフォーマンスが得られます。デフォルト・キャッシュまたはそれにバインドされたトランザクション・ログを持つキャッシュが、論理ページ・サイズの 2 倍のメモリ・プールによって設定されている場合、**Adaptive Server** は論理ページ・サイズの 2 倍のデフォルト I/O サイズを使用します。それ以外の場合、論理ページ・サイズのメモリ・プールを使用します。
- `sp_logiosize` システム・プロシージャを使用して、データベースに対するデフォルト・ログ I/O サイズを増やすことができます。指定する I/O サイズは、必ずトランザクション・ログにバインドされたキャッシュ内にあるメモリ・プールと同じサイズにしてください。それ以外の場合、**Adaptive Server** は論理ページ・サイズのメモリ・プールを使用します。

ウォッシュ率

- ウォッシュ・サイズのデフォルト値は、次のように計算されます。
 - プール・サイズが 300MB 未満の場合には、デフォルトのウォッシュのサイズは、プールにあるバッファの 20% に設定されます。
 - プールのサイズが 300MB を超える場合には、デフォルトのウォッシュ・サイズは、300MB に含まれるバッファの数の 20% になります。
- ウォッシュ・サイズの最小設定は 10 バッファ、最大設定はプールのサイズの 80% です。
- それぞれのメモリ・プールでは、そのプールにあるバッファ・チェーンの LRU (Least recently used) の終わりに、ウォッシュ・エリアが含まれています。ダーティ・ページ (キャッシュにある間に変更されたページ) がウォッシュ・エリアに移動すると、Adaptive Server はこれらのページに非同期書き込みを開始します。ページがプールの LRU 終端に達する前に、ディスクにページを書き込むことができるよう、ウォッシュ・エリアには十分な大きさを用意してください。Adaptive Server がクリーン・バッファを待つ必要がある場合は、パフォーマンスが低下します。
 - ウォッシュ・エリアのバッファのデフォルトの比率は、20% です。この比率でほとんどのアプリケーションには充分です。極端に大きいメモリ・プールを使用していて、アプリケーションのデータ修正率が非常に高い場合、プールの 1 または 2% のサイズを増やします。sp_sysmon を実行して推奨のサイズを確認するか、有効なウォッシュ・サイズを選択する方法について、Sybase 製品の保守契約を結んでいるサポート・センタに連絡してください。

ローカル非同期プリフェッチ率

- プールの非同期プリフェッチ率のデフォルト値は、設定パラメータ `global async prefetch limit` で設定します。プールの制限は、常にグローバル制限を無効にします。
- プールのプリフェッチを無効にするには (グローバル制限がゼロでない値の場合)、プールの制限値を 0 に設定します。
- 非同期プリフェッチ制限の変更がパフォーマンスに及ぼす影響については、『パフォーマンス&チューニング・ガイド』を参照してください。

パーミッション sp_poolconfig のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、メモリ・プールを再設定するには <code>manage data cache</code> 権限を持つユーザであることが必要。 すべてのユーザは、 <code>sp_poolconfig</code> を実行してメモリ・プールの情報を取得できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、メモリ・プールを再設定するには <code>sa_role</code> が付与されたユーザであることが必要。 すべてのユーザは、 <code>sp_poolconfig</code> を実行してメモリ・プールの情報を取得できる。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_cacheconfig](#)、[sp_helpcache](#)、[sp_logiosize](#)、[sp_unbindcache](#)、[sp_unbindcache_all](#)

sp_post_xpload

説明	エンディアン・タイプの異なるプラットフォーム間で load database を行ったときに使用され、インデックスをチェックして再構築します。
構文	sp_post_xpload [force]
パラメータ	force 指定すると、sp_post_xpload で dbcc reindex に reindex_opt_force が使用されます。
例	別のプラットフォームからデータベースをロードしたとき、そのインデックスを再構築するために次のように実行します。 <pre>sp_post_xpload</pre>
使用法	<ul style="list-style-type: none">• データベースのすべてのユーザ・テーブルで次のインデックスが再構築されます。<ul style="list-style-type: none">• APL テーブルのノンクラスタード・インデックス• DOL テーブルのクラスタード・インデックス• DOL テーブルのノンクラスタード・インデックス• sp_post_xpload のみではシステム・テーブルのインデックスは処理されません。システム・テーブルのインデックスは、online database を実行したとき再構築されます。• インデックスの再構築は、drop index と create index を使用しても実行できます。• sp_post_xpload は、エンディアン・タイプの異なるプラットフォーム間でデータベースをロードしたときだけ実行してください。• インデックスのステータスが suspect (疑わしい) のときは、sp_xpload、drop index、または create index を実行してインデックスを再設定してください。• ストアド・プロシージャは、load database の実行後最初に実行される時 syscomments 内の SQL テキストから再コンパイルされません。テキストから再コンパイルするパーミッションがない場合は、dbcc upgrade_object を使用してオブジェクトをアップグレードします。

プラットフォーム間のダンプおよびロード操作時の疑わしいパーティションの処理

- 初めての `online database` コマンドの実行中、エンディアン・タイプの異なる2つのプラットフォーム間で `load database` を実行すると、ハッシュ・パーティションに `suspect` のマークが付けられます。
- `unichar` または `univarchar` パーティション・キーで内部生成されたパーティション条件を持つラウンドロビン分割のグローバル・クラスタード・インデックスに、`suspect` のマークが付けられます。
- データベースをオンラインにした後に、`sp_post_xpload` を使用して疑わしいパーティションとインデックスを修正します。

パーミッション

`sp_post_xpload` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データベースに対する <code>load database</code> 権限または <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

`dump database`、`load database`

sp_primarykey

説明	テーブルまたはビューのプライマリ・キーを定義します。
構文	<code>sp_primarykey <i>tablename</i>, <i>col1</i> [, <i>col2</i>, <i>col3</i>, ..., <i>col8</i>]</code>
パラメータ	<p><i>tablename</i> プライマリ・キーを定義するテーブルまたはビューの名前です。</p> <p><i>col1</i> プライマリ・キーを構成する最初のカラムの名前です。プライマリ・キーは、1 カラムから 8 カラムで構成することができます。</p>
例	<p>例 1 <code>au_id</code> フィールドを <code>authors</code> テーブルのプライマリ・キーとして定義します。</p> <pre>sp_primarykey authors, au_id</pre> <p>例 2 <code>lastname</code> フィールドと <code>firstname</code> フィールドの組み合わせを、<code>employees</code> テーブルのプライマリ・キーとして定義します。</p> <pre>sp_primarykey employees, lastname, firstname</pre>
使用法	<ul style="list-style-type: none"> • <code>sp_primarykey</code> を実行すると、<code>syskeys</code> テーブルにキーが追加されません。テーブルまたはビューの所有者だけがそのプライマリ・キーを定義できます。<code>sp_primarykey</code> は、参照整合性制約を実行しません。プライマリ・キーを定義、変更するには、<code>create table</code> または <code>alter table</code> コマンドの <code>primary key</code> 句を使用してください。 • <code>sp_primarykey</code>、<code>sp_commonkey</code>、<code>sp_foreignkey</code> によってキーを定義して、使用しているデータベース設計の暗黙の論理関係を明示的にしてください。この情報は、アプリケーション・プログラムで使用できます。 • テーブルまたはビューが所有できるプライマリ・キーは 1 つだけです。定義されているキーについてのレポートを表示するには、<code>sp_helpkey</code> を実行します。 • インストール・プロセスによって、システム・テーブルの適切なカラムに対し <code>sp_primarykey</code> が実行されます。
パーミッション	<code>sp_primarykey</code> を実行するには、テーブルの所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [alter table](#)、[create table](#)、[create trigger](#)

システム・プロシージャ [sp_commonkey](#)、[sp_dropkey](#)、[sp_foreignkey](#)、[sp_helpjoins](#)、[sp_helpkey](#)

sp_processmail

説明	(Windows のみ) xp_findnextmsg 、 xp_readmail 、 xp_sendmail 、および xp_deletemail の各システム ESP (拡張ストアド・プロシージャ) を使用して、Adaptive Server メッセージ・インボックス内のメッセージの読み込み、処理、送信、削除を行います。
構文	<pre>sp_processmail [subject] [, originator [, dbuser [, dbname [, filetype [, separator]]]]</pre>
パラメータ	<p>subject メッセージのサブジェクト・ヘッダです。 <i>subject</i> を指定して、<i>originator</i> を指定しなければ、<code>sp_processmail</code> は、指定したサブジェクト・ヘッダを持つインボックス内のすべての未読メッセージを処理します。 <i>subject</i> と <i>originator</i> の両方を指定すると、<code>sp_processmail</code> は指定した送信者が送信したメッセージのうち、指定されたサブジェクト・ヘッダを持つすべての未読メッセージを処理します。 <i>subject</i> または <i>originator</i> のどちらも指定しなければ、<code>sp_processmail</code> は、Adaptive Server のメッセージ・インボックス内のすべての未読メッセージを処理します。</p> <p>originator 受信メッセージの送信側です。 <i>originator</i> を指定し、<i>subject</i> を指定しないと、<code>sp_processmail</code> は、指定した送信側が送信したインボックス内のすべての未読メッセージを処理します。</p> <p>dbuser メッセージ内のクエリを実行するために、ユーザ・コンテキストで使用する Adaptive Server ログイン名を指定します。デフォルトは “<code>guest</code>” です。</p> <p>dbname メッセージ内のクエリを実行するために、データベース・コンテキストで使用するデータベース名を指定します。デフォルトは “<code>master</code>” です。</p> <p>filetype クエリの結果を格納する添付ファイルのファイル拡張子を指定します。デフォルトは “.txt” です。</p> <p>separator クエリの結果にカラムのセパレータとして使用する文字を指定します。 <code>isql</code> の <code>/s</code> オプションと同じ役割をします。デフォルトはタブ文字です。</p>

例 **例 1** Adaptive Server のインボックス内で “janet” というメール・ユーザが送信し、サブジェクト・ヘッダが “SQL Report” となっているすべての未読のメッセージを処理します。受け取ったクエリは、ユーザ “sa” として、salesdb データベースで処理し、クエリの結果は、メール・メッセージに .res ファイルを添付して “janet” に返します。返した結果のカラムは、セミコロンによって区切られます。

```
sp_processmail @subject="SQL REPORT", @originator="janet", @dbuser="sa",
  @dbname="salesdb", @filetype="res", @separator=";"
```

例 2 master データベースでユーザ “sa” として Adaptive Server のインボックス内にあるすべての未読メッセージを処理し、クエリの結果は .txt ファイルとして、メール・メッセージに添付して返されます。返した結果のカラムは、タブ文字によって区切られます。

```
sp_processmail @dbuser="sa"
```

使用法

- sp_processmail は、[xp_findnextmsg](#)、[xp_readmail](#)、[xp_sendmail](#)、[xp_deletemail](#) の各システム ESP を使用して、Adaptive Server のメッセージ・インボックス内にあるメッセージの読み込み、処理、送信、削除を行います。
- sp_processmail は、処理された受信メール・メッセージの送信側に、発信メールを送ります。
- sp_processmail は、ESP を呼び出すときに、[xp_sendmail](#) に対し、*dbuser*、*dbname*、*attachname*、*separator* パラメータを除いたデフォルトのパラメータを使用します。これらのパラメータは、sp_processmail のパラメータを使用して上書きすることができます。
- sp_processmail は、すべてのメッセージを Adaptive Server のクエリとして処理します。Adaptive Server のインボックス内にあるメッセージを読み込み、メッセージの送信側とそのメッセージのすべての cc と bcc の受信者にクエリの結果を添付ファイルとして返します。sp_processmail は、添付ファイルの名前を生成するときに、“syb” の後に 5 桁の乱数を付け、その後に *filetype* パラメータで指定された拡張子を付けます。たとえば “syb84840.txt” となります。
- sp_processmail は、処理したメッセージをインボックスから削除します。
- *subject* と *originator* パラメータは、どのメッセージを処理するかを指定します。どちらのパラメータも指定されなければ、sp_processmail は、Adaptive Server のメッセージ・インボックスにあるすべての未読メッセージを処理します。
- sp_processmail は、受信メールの添付ファイルは処理しません。クエリは受信メッセージの本文に記述してください。

パーミッション sp_processmail を実行するには execute パーミッションが必要です。このパーミッションは、sybsystemprocs のデータベース所有者が他のユーザに付与できます。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 **拡張ストアド・プロシージャ** [xp_deletemail](#)、[xp_findnextmsg](#)、[xp_readmail](#)、[xp_sendmail](#)、[xp_startmail](#)
ユーティリティ `isql`

sp_procxmode

説明 ストアド・プロシージャと対応する実行モードを表示または変更します。

構文 sp_procxmode [*procname* [, *tranmode*]]

パラメータ

procname

調査または変更するトランザクション・モードのストアド・プロシージャの名前です。

tranmode

ストアド・プロシージャの新しい実行モードです。有効な値は、"chained"、"unchained"、および "anymode"、for transaction modes、および '[No] Dynamic Ownership Chain' です。

例 **例 1** 現在のデータベース内にあるすべてのストアド・プロシージャのトランザクション・モードを表示します。

```
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
```

例 2 byroyalty ストアド・プロシージャのトランザクション・モードを表示します。

```
sp_procxmode byroyalty

procedure name      transaction mode
-----
byroyalty           Unchained
```

例 3 pubs2 データベース内の byroyalty ストアド・プロシージャのトランザクション・モードを、“unchained” から “chained” に変更します。

```
sp_procxmode byroyalty, "chained"
```

使用法

- ストアド・プロシージャのトランザクション・モードを変更するには、ストアド・プロシージャの所有者、ストアド・プロシージャが保管されているデータベースの所有者、またはシステム管理者でなければなりません。データベース所有者またはシステム管理者は、データベースとユーザ名を使用して修飾することにより、別のユーザのストアド・プロシージャのモードを変更することができます。次に例を示します。

```
sp_procxmode "otherdb.otheruser.newproc", "chained"
```

- `sp_procxmode` を使用するには、`set` コマンドの `chained` オプションを使用して、連鎖トランザクション・モードをオフにします。このオプションのデフォルトは、オフです。
- パラメータを指定しないで `sp_procxmode` を実行すると、現在のデータベース内のすべてのストアド・プロシージャのトランザクション・モードがレポートされます。
- ストアド・プロシージャのトランザクション・モードを確認するだけで変更しない場合には、次の構文を使用します。

```
sp_procxmode procname
```

- ストアド・プロシージャのトランザクション・モードを変更するには、次の構文を使用します。

```
sp_procxmode procname, tranmode
```

- ストアド・プロシージャを作成する場合、**Adaptive Server** は、現在のセッションのトランザクション・モードを関連付けます。したがって、次の点に留意してください。
 - 「連鎖」ストアド・プロシージャは、連鎖トランザクション・モードのセッションでのみ、実行できます。
 - 「非連鎖」ストアド・プロシージャは、非連鎖トランザクション・モードのセッションでのみ、実行できます。

特定のストアド・プロシージャを連鎖セッションまたは非連鎖セッションのどちらかで実行する場合は、そのトランザクション・モードを“**anymode**”に設定します。

- 誤ったトランザクション・モードでストアド・プロシージャを実行しようとする、**Adaptive Server** が警告メッセージを返しますが、現在トランザクションを実行中であっても影響はありません。

- `sp_procxmode procname, 'Dynamic Ownership Chain'` を実行すると、ストアド・プロシージャ内の `Dynamic SQL (execute immediate)` 文のパーミッションが、プロシージャ作成者に照らしてチェックされます。
- `sp_procxmode procname, 'No Dynamic Ownership Chain'` を実行 (省略した場合のデフォルトの動作) すると、ストアド・プロシージャ内の `Dynamic SQL (execute immediate)` 文のパーミッションが、プロシージャ実行者に照らしてチェックされます。

パーミッション

`sp_procxmode` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、プロシージャの所有者であるか、 <code>manage database</code> 権限を持つユーザであることが必要。自分のプロシージャに対しては、すべてのユーザが <code>sp_procxmode</code> を実行できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者、プロシージャの所有者、または <code>sa_role</code> が付与されたユーザであることが必要。すべてのユーザは、 <code>sp_procxmode</code> を実行して、トランザクション・モードを表示できる。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

コマンド `begin transaction`、`commit`、`save transaction`、`set`

sp_querysmobj

説明 (Tivoli Storage Manager のみ) Adaptive Server バックアップ・オブジェクトのリストについて Tivoli Storage Manager (TSM) を問い合わせます。

注意 sp_querysmobj は、サイトで TSM がライセンスされている場合にのみサポートされます。

構文 `sp_querysmobj "syb_tsm", "output_file", "server_name"
{, "database_name", "object_name", "dump_type",
"until_time", "bs_name"}`

パラメータ

syb_tsm

TSM との通信を有効にする libsyb_tsm.so モジュールを起動するキーワードです。

output_file

Backup Server が TSM バックアップ・オブジェクトのリストを書き込むファイルです。

server_name

TSM のバックアップ・オブジェクトに関連付けられている Adaptive Server の名前です。

database_name

TSM のバックアップ・オブジェクトに関連付けられているデータベースの名前です。アスタリスク (*) はすべてのデータベースを表します。

object_name

dump database コマンドか dump transaction コマンドに指定された TSM のバックアップ・オブジェクトの名前です。このパラメータを省略すると、すべてのバックアップ・オブジェクトを問い合わせます。アスタリスク (*) はすべてのバックアップ・オブジェクトを表します。

dump_type

問い合わせるバックアップ・オブジェクト・タイプです。有効な値は次のとおりです。

- **DB** — **dump database** コマンドによって作成されたデータベース・バックアップ・オブジェクト。
- **XACT** — **dump transaction** コマンドによって作成されたデータベース・バックアップ・オブジェクト。
- ***** — すべてのデータベース・バックアップ・オブジェクト。これがデフォルト値です。

until_time

日付のタイムスタンプです。指定した時間の前に **sp_querysmobj** に入力された基準を満たすすべてのバックアップ・オブジェクトを問い合わせます。このパラメータを省略すると、指定した基準を満たすすべてのバックアップ・オブジェクトを問い合わせます。

bs_name

リモート Backup Server の名前です。**bs_name** を省略すると、デフォルトの **SYB_BACKUP** が使用されます。

例 **例 1** Adaptive Server “demo_svr1” のすべての TSM バックアップ・オブジェクトを問い合わせ、**/tmp/qtsm/5_1.out** にリストを書き込みます。

```
sp_querysmobj "syb_tsm", "/tmp/qtsm/5_1.out", "demo_svr1"
```

例 2 Adaptive Server “demo_svr1” およびデータベース **pubs2** のすべての TSM バックアップ・オブジェクトを問い合わせ、**/tmp/qtsm/5_2.out** にリストを書き込みます。

```
sp_querysmobj "syb_tsm", "/tmp/qtsm/5_2.out", "demo_svr1", "pubs2"
```

例 3 Adaptive Server “demo_svr1” およびデータベース **pubs2** のすべての TSM データベース・バックアップ・オブジェクトを問い合わせ、**/tmp/qtsm/5_3.out** にリストを書き込みます。

```
sp_querysmobj "syb_tsm", "/tmp/qtsm/5_3.out", "demo_svr1", "pubs2", "*", "DB"
```

使用法

TSM に関する Sybase のサポートの詳細については、『IBM Tivoli Storage Manager と Backup Server の使用』を参照してください。

パーミッション

sp_querysmobj のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 dump any database 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 sa_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 [sp_deletesmobj](#)

sp_recompile

説明 指定したテーブルを使用するすべてのストアド・プロシージャとトリガが、次の実行時に再コンパイルされるように設定します。

構文 `sp_recompile objname`

パラメータ `objname`
現在のデータベース内にあるテーブルの名前です。

例 `titles` テーブルを使用するトリガとストアド・プロシージャを、次回実行時に再コンパイルします。

```
sp_recompile titles
```

使用法

- コンパイルに伴い、オプティマイザは `sysprocedures` に格納されている正規化されたクエリ・ツリーからクエリ・プランを作成してプロシージャ・キャッシュに格納します。これは、プロシージャやトリガの実行時に、それ用のフリーなプランがプロシージャ・キャッシュ内になければ必ず発生します。インデックスを追加するなど、統計に影響を及ぼす変更をデータベースに加えると、これらのクエリ・プランの効率が低下する可能性があります。テーブルを操作するストアド・プロシージャとトリガを再コンパイルすることにより、最大限の効率を得られるようにクエリを最適化できます。

注意 `create index` および `update statistics` を実行すると、スキーマがわずかに変更され、ストアド・プロシージャとトリガが自動的に再コンパイルされます。このとき、`sp_recompile` を使用すると、再コンパイルが重複します。

- `sp_recompile` は、現在のデータベース内にある `objname` のみを検索します。このコマンドを実行すると、`objname` を参照するトリガとストアド・プロシージャが次の実行時に再コンパイルされます。
- `sp_recompile` は、システム・テーブルには使用できません。
- バージョン 12.5 以前の Adaptive Server では、`sp_recompile` を実行すると、アドホック・クエリの実行に影響を与える可能性があります。そのような場合、Adaptive Server はスキーマ変更エラー (エラー番号 540) を返して、アドホック・クエリの実行をアボートしていました。このバージョンでは `sp_recompile` はそのようなアドホック・クエリに影響を与えないため、エラー番号 540 のエラーが返されることはなくなりました。

パーミッション `sp_recompile` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [create index](#)、[update statistics](#)

sp_refit_admin

説明	(クラスタ環境のみ) プロセスの現在のステータスの表示、 <code>disk refit</code> プロセスのステータスの再設定、インスタンスの <code>disk refit</code> プロセスのスキップなど、さまざまな <code>disk refit</code> 関連アクションを実行するインタフェースを提供します。
構文	<code>sp_refit_admin ['help'] 'status' ['reset' 'skiprefit' [, <i>instance_name</i>]] [['removedevice', <i>device_name</i>]</code>
パラメータ	<p>help</p> <p><code>sp_refit_admin</code> の構文および使用法に関する情報を表示します。</p> <p>status</p> <p><code>disk refit</code> プロセスの現在のステータスを表示します。<code>disk refit</code> が保留中のインスタンスとそのプライベート・デバイスがすべてリストされます。そのようなデバイスが存在しない場合は、その旨を示すメッセージが出力されます。</p> <p>reset</p> <p><code>disk refit</code> プロセスのステータスを再設定します。オプションのパラメータ <i>instance_name</i> を指定できます。</p> <p><i>instance_name</i> を指定しないでこのパラメータを使用すると、<code>disk refit</code> プロセスはフェーズ 1 の開始点に戻り、次の <code>disk refit</code> コマンドが実行されると、<code>disk refit</code> プロセスはフェーズ 1 から開始され、すべての通常の共有デバイスとインスタンスのプライベート・デバイスが修復されます。</p> <p><i>instance_name</i> を指定してこのパラメータを使用すると、<code>disk refit</code> プロセスはそのインスタンスのフェーズ 2 の開始点に戻り、そのインスタンスで次の <code>disk refit</code> コマンドが実行されると、<code>disk refit</code> プロセスがそのインスタンスのフェーズ 2 から開始され、そのインスタンスのプライベート・デバイスのみが修復されます。</p> <p>skiprefit</p> <p>デバイスを削除せずに、クラスタ内の 1 つまたはすべてのインスタンスについて、<code>disk refit</code> プロセスのフェーズ 2 の実行をスキップします。このパラメータは、<code>disk refit</code> プロセスのフェーズ 1 が完了している場合にのみ有効です。オプションのパラメータ <i>instance_name</i> を指定できます。</p> <p>removedevice</p> <p><code>disk refit</code> プロセスからデバイスを削除します。このパラメータは、入力パラメータ (<i>device_name</i> または <i>instance_name</i>) として削除対象のデバイスの名前を要求します。</p>

例

例 1 disk refit プロセスのステータスをフェーズ 1 の開始点に再設定します。

```
sp_refit_admit 'reset'
```

ユーザは、reset の実行後、disk refit プロセスのフェーズ 1 およびフェーズ 2 を実行する必要があります。

例 2 'cluster1_instance1' というインスタンスの disk refit プロセスのステータスをこのインスタンスのフェーズ 2 の開始点に再設定します。

```
sp_refit_admin 'reset', 'cluster1_instance1'
```

このインタフェースを使用すると、'cluster1_instance1' が所有するプライベート・デバイスで作成されたすべてのデータベースの sysdatabases エントリと、'cluster1_instance1' が所有するプライベート・デバイスに対応する sysusages エントリが削除されます。ユーザは、このプロシージャの実行後、'cluster1_instance1' に対して disk refit のフェーズ 2 を実行する必要があります。

例 3 インスタンス 'cluster1_instance1' の修復保留中のすべてのプライベート・デバイスについて、disk refit プロセスをスキップします。

```
sp_refit_admin 'skiprefit', 'cluster1_instance1'
```

この例では、'cluster1_instance1' が所有する修復保留中のプライベート・デバイスを使用するすべてのデータベースの sysdatabases エントリと、削除されたすべてのデータベースの sysusages のすべてのエントリが削除されます。

クラスタ内のすべてのインスタンスの修復保留中のすべてのプライベート・デバイスについて、disk refit プロセスをスキップするには、次のように入力します。

```
sp_refit_admin 'skiprefit'
```

例 4 disk refit プロセスからデバイス "device1" を削除するには、次のように入力します。

```
sp_refit_admin 'removedevice', 'device1'
```

このアクションによって、'device1' で作成されたすべてのデータベースの sysdatabases エントリと、'device1' に対応するすべての sysusages エントリが削除されます。sysdevices からは 'device1' も削除されます。

使用法	<ul style="list-style-type: none">• 通常の操作を再開する前にシステム・テーブルの一貫性を確認するため、skiprefit を実行した後、『Clusters ユーザーズ・ガイド』の「第 12 章 トラブルシューティング」の指示に従ってください。• removedevice を使用して refit プロセスからデバイスを削除できるのは、disk refit プロセスの実行中のみです。removedevice は、sp_dropdevice の代わりには使用できません。• -m オプションおよびトレース・フラグ 3608 ON を指定してインスタンスを起動した場合でも sp_refit_admin を使用できます。
パーミッション	sp_refit_admin のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、manage disk 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。
参照	disk refit で生じる問題の詳細については、『トラブルシューティング：エラー・メッセージと詳細な解決方法』を参照してください。

sp_remap

説明	Adaptive Server 15.7 ESD #2 以降では、このシステム・プロシージャは推奨されていません。
構文	なし
パラメータ	なし
使用法	なし

sp_remoteoption

説明 リモート・ログイン・オプションを表示または変更します。

構文 sp_remoteoption [remoteserver[, loginname
[, remotename[, optname[, optvalue]]]]]

パラメータ *remoteserver*
このサーバ上で RPC を実行するサーバの名前です。

注意 このマニュアル・ページで「ローカル・サーバ」というときは、「リモート・サーバ」から実行されているリモート・プロセスを実行中のサーバを指します。

loginname

remoteserver、*loginname*、*remotename* の組み合わせでリモート・ログインを識別するログイン名です。

remotename

remoteserver、*loginname*、*remotename* の組み合わせでリモート・ログインを識別するリモート・ユーザ名です。

optname

変更するオプションの名前です。現在、使用できるオプションは **trusted** オプションだけです。このオプションは、ローカル・サーバが、特定のリモート・ログインについてはユーザ・アクセスの検証を行わずに、他のサーバからのログインを受け入れることを意味します。デフォルトではパスワードの検証が行われます。**Adaptive Server** は、オプション名に含まれるあらゆるユニークな文字列を認識します。オプション名に埋め込みブランクを使用する場合は、そのオプション名を引用符で囲みます。

optvalue

true または **false** のどちらかです。オプションの設定をオンにするには **true** を指定し、設定をオフにするには **false** を指定します。

例 1 リモート・ログイン・オプションのリストを表示します。

```
sp_remoteoption

Settable remote login options.
remotelogin_option
-----
trusted
```

例 2 リモート・サーバ **GATEWAY** からのリモート・ログインを、**trusted** 設定に定義します (パスワードの確認は行われません)。

```
sp_remoteoption GATEWAY, churchy, pogo, trusted, true
```

例 3 リモート・サーバ GATEWAY からの “pogo” のリモート・ログインに対して、trusted の設定を解除します (パスワードの確認が行われます)。

```
sp_remotoption GATEWAY, churchy, pogo, trusted, false
```

例 4 GATEWAY からのすべてのログインを、ローカル・サーバ上のログイン “albert” にマップされる trusted ログインとして定義します。

```
sp_remotoption GATEWAY, albert, NULL, trusted, true
```

使用法

- リモート・ログイン・オプションのリストを表示するには、パラメータを指定しないで `sp_remotoption` を実行します。
- `sp_addremotlogin` を使用して、リモート・サーバからのすべてのユーザを同じローカル名にマップしている場合は、これらのユーザに `trusted` を指定します。たとえば、サーバ GOODSRV からのユーザで、“albert” にマップされるすべてのユーザが `trusted` ユーザである場合は、次のように指定します。

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

ログインが `trusted` として指定されていない場合は、リモート・サーバにログインするときに、ローカル・サーバのパスワードを指定しない限り、これらのログインはローカル・サーバで RPC を実行することはできません。これらのユーザは、**Open Client Client-Library** を使用するとき、`ct_remote_pwd` ルーチンを使用してサーバ対サーバ接続のパスワードを指定することができます。`isql` と `bcp` は、ユーザが RPC 接続用のパスワードを指定することを許可しません。

ユーザが「統一化ログイン」を使用してリモート・サーバにログインする場合は、そのログインにはローカル・サーバでも「`trusted`」が指定されている必要があります。そうでない場合は、ユーザはリモート・サーバにログインするときにサーバのパスワードを指定する必要があります。

リモート・プロシージャ・コールに対するサーバの設定と「統一化ログイン」の使用方法の詳細については、『システム管理ガイド』を参照してください。

パーミッション

`sp_remotoption` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any remote login</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sso_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addrmotelogin](#)、[sp_droptremotelogin](#)、[sp_helpremotelogin](#)

ユーティリティ `isql`

sp_remotesql

説明 (コンポーネント統合サービスのみ) リモート・サーバへの接続を確立し、クライアントからのクエリ・バッファをリモート・サーバに渡し、結果をクライアントに伝えます。

構文 `sp_remotesql server, query [, query2, ..., query254]`

パラメータ

server

`sp_addserver` によって定義されたりモート・サーバの名前です。

query

最長 255 文字のクエリ・バッファです。

query2 … *query254*

最長 255 文字のクエリ・バッファです。これらの引数を指定した場合は、*query1* の内容に連結され、1 つのクエリ・バッファが作成されます。

例

例 1 FREDS_SERVER にクエリ・バッファを渡し、このサーバが `select @@version` を解釈して、クライアントに結果を返します。Adaptive Server は結果を解釈しません。

```
sp_remotesql FREDS_SERVER, "select @@version"
```

例 2 ストアド・プロシージャでの `sp_remotesql` の使用例を示します。この例と例 1 は、クライアントに同じ情報を返します。

```
create procedure freds_version
as
exec sp_remotesql FREDS_SERVER, "select @@version"
go
exec freds_version
go
```

例 3 サーバは 2 つのクエリ・バッファを 1 つのバッファに連結し、全体の `insert` 文を DCO_SERVER サーバに渡し、このサーバが処理を行います。`insert` 文の構文は、DCO_SERVER が解釈できるフォーマットです。サーバは、返された情報を解釈しません。また、この例では `@@error` に返される値も調べられます。

```
sp_remotesql DCO_SERVER,
"insert into remote_table
(numbercol,intcol, floatcol,datecol )",
"values (109.26,75, 100E5,'10-AUG-85')"
```

```
select @@error
```

例 4 ローカル変数を `sp_remotesql` のパラメータとして使用する方法を説明します。

```
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
```

使用法

- `sp_remotesql` はリモート・サーバへの接続を確立し、クライアントからのクエリ・バッファをリモート・サーバに渡し、結果をクライアントに伝えます。ローカル・サーバは結果を途中で調べません。
- `sp_remotesql` は、他のストアド・プロシージャの内部でも使用できます。
- クエリ・バッファ・パラメータは、最長 255 文字までの文字表現にしてください。char や varchar でないクエリ・バッファを使用すると、データ型変換エラーが返されます。
- エラー・メッセージの重大度レベルが 10 より大きい場合、`sp_remotesql` は、グローバル変数 `@@error` の値をリモート・サーバから返された最後のエラー・メッセージの値に設定します。
- `sp_remotesql` がトランザクションの内部で発行されると、Adaptive Server は、トランザクションがリモート・サーバで開始されているかどうかを確認してから、実行用のクエリ・バッファを渡します。トランザクションが終了すると、リモート・サーバにそのトランザクションにコミットするよう指示します。クエリ・バッファの内容によって実行される作業は、トランザクションが定義した作業単位の一部です。

トランザクション制御文がクエリ・バッファの一部である場合、クライアントは、トランザクション `commit` と `rollback` が意図したとおりに行われることを保証する責任があります。クエリ・バッファのトランザクション制御コマンドを `Transact-SQL` と併用すると、予想できない結果になることがあります。

- ローカル・サーバは、リモート・サーバへの接続を管理します。クエリ・バッファに `connect to` コマンドまたは `disconnect` コマンドを埋め込むと、リモート・サーバによる解釈を必要とする結果が生成されます。この埋め込みは必須の手順ではなく、推奨できません。通常の場合、結果は構文エラーになります。

パーミッション `sp_remolesql` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 コマンド [connect to...disconnect](#)

システム・プロシージャ [sp_addserver](#)、[sp_autoconnect](#)、[sp_passthru](#)

sp_rename

説明	現在のデータベース内のユーザ作成オブジェクトまたはユーザ定義データ型の名前を変更します。
構文	<pre>sp_rename <i>objname</i>, <i>newname</i> [,"<i>index</i>" "<i>column</i>"]</pre>
パラメータ	<p><i>objname</i> ユーザ作成のオブジェクト(テーブル、ビュー、カラム、ストアド・プロシージャ、インデックス、トリガ、デフォルト、ルール、検査制約、参照制約、ユーザ定義データ型)の元の名前です。名前を変更するオブジェクトがテーブル内のカラムの場合、<i>objname</i>は“<i>table.column</i>”という形式で指定します。オブジェクトがインデックスの場合、<i>objname</i>は“<i>table.indexname</i>”という形式で指定します。</p> <p><i>newname</i> オブジェクトまたはデータ型の新しい名前です。識別子のルールに従い、現在のデータベース固有のユニークな名前を指定する必要があります。</p> <p><i>index</i> 名前を変更するオブジェクトが、カラムではなくインデックスであることを指定します。この引数を使用すると、インデックスを一度削除してから再作成しなくても、カラムと同じ名前のインデックスの名前を変更できます。</p> <p><i>column</i> 名前を変更するオブジェクトが、インデックスではなくカラムであることを指定します。この引数は、<i>index</i>引数と同様の働きをするオプションです。</p>
例	<p>例 1 titles テーブルの名前を books に変更します。</p> <pre>sp_rename titles, books</pre> <p>例 2 books テーブル内の title カラムの名前を、bookname に変更します。</p> <pre>sp_rename "books.title", bookname</pre> <p>例 3 books テーブル内の titleind インデックスの名前を、titleindex に変更します。</p> <pre>sp_rename "books.titleind", titleindex</pre> <p>例 4 tid というユーザ定義データ型の名前を、bookid に変更します。</p> <pre>sp_rename tid, bookid</pre> <p>例 5 titles テーブル内の title_id インデックスの名前を、isbn に変更します。</p> <pre>sp_rename "titles.title_id", isbn, "index"</pre>

使用法

- `sp_rename` を実行すると、ユーザ作成オブジェクトまたはデータ型の名前が変更されます。変更できるのは、`sp_rename` を発行したデータベース内のオブジェクトまたはデータ型の名前だけです。
- カラムまたはインデックスの名前を変更するときは、*newname* でテーブル名を指定しないでください。例 2、例 3、例 5 を参照してください。
- カラム名とインデックス名が同じ場合は、[*“index”* | *“column”*] 引数を使用します。この引数は、インデックスとカラムのどちらの名前を変更するのかを指定します。次の例では、`idx` という名前のインデックスとカラムがどちらも存在することが想定されています。

```
sp_rename "t.idx", new_idx, "column"
```

```
-----
```

```
Column name has been changed.(Return status = 0)
```

```
sp_rename "t.idx", new_idx, "index"
```

```
-----
```

```
Index name has been changed.(Return status = 0)
```

- ビューによって参照されるオブジェクトまたはカラムの名前を変更すると、次のような警告メッセージが表示されます。


```
Changing an object or column name could break
existing stored procedures, cached statements or
other compiled objects.
```
- `sp_engine` は、オープン・トランザクションが存在しない場合に連鎖トランザクション・モードを使用するセッションで実行できません。
- システム・オブジェクトとシステム・データ型の名前は、変更できません。

警告！名前が変更されたオブジェクトに依存するプロシージャ、トリガ、ビューは、動作しなくなります。`sp_rename` を実行する前に、すべての従属オブジェクトの定義を変更してください。従属オブジェクトは、`sp_depends` を使って検索できます。

パーミッション

`sp_rename` を実行するには、オブジェクト所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

setuser コマンドを使用すると、別のデータベース・ユーザの ID を想定して、他のユーザが所有するオブジェクトの名前を変更できます。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_depends](#)、[sp_rename](#)

sp_rename_qpgroup

説明	抽象プラン・グループの名前を変更します。
構文	<code>sp_rename_qpgroup old_name, new_name</code>
パラメータ	<p><i>old_name</i> 抽象プラン・グループの現在の名前です。</p> <p><i>new_name</i> グループの新しい名前です。<i>new_name</i> に、データベースに既存の抽象プラン・グループの名前を指定することはできません。</p>
例	<pre>sp_rename_qpgroup dev_plans, prod_plans</pre> <p>グループの名前を <code>dev_plans</code> から <code>prod_plans</code> に変更します。</p>
使用法	<ul style="list-style-type: none"> 抽象プラン・グループの名前を変更するには、<code>sp_rename_qpgroup</code> を使用します。既存のプラン・グループの名前を新しい名前に使用することはできません。 <code>sp_rename_qpgroup</code> を実行しても、名前が変更されたグループの内容には影響しません。既存の抽象プランの ID は変更されません。 デフォルトの抽象プラン・グループ <code>ap_stdin</code> と <code>ap_stdout</code> の名前を変更することはできません。 <code>sp_rename_qpgroup</code> をトランザクション内で実行することはできません。
パーミッション	<code>sp_rename_qpgroup</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage abstract plans</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_help_qpgroup](#)

sp_renamedb

- 説明** ユーザ・データベースの名前を変更します。
- 構文** `sp_renamedb dbname, newname`
- パラメータ**
- dbname*
データベースの元の名前です。
- newname*
データベースの新しい名前です。識別子のルールに従い、ユニークなデータベース名を指定する必要があります。
- 例** **例 1** `accounting` データベースの名前を `financial` に変更します。

```
sp_renamedb accounting, financial
```

例 2 Transact-SQL の予約語である `work` というデータベースの名前を、`tworkdb` に変更します。この例では、`sp_dboption` を使用して `work` データベースをシングルユーザ・モードにしてから、データベース名を変更し、後でそのデータベースをマルチユーザ・モードにリストアする方法を示しています。

```
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
```

- 使用法**
- `sp_renamedb` はデータベースの名前を変更します。システム・データベースの名前や、外部参照整合性制約があるデータベースの名前は、変更できません。
 - システム管理者は、`sp_dboption` を使ってデータベースをシングルユーザ・モードにしてから、データベース名を変更し、その後でマルチユーザ・モードにリストアする必要があります。

- `sp_renamedb` は、データベース内のテーブルが別のデータベースを参照したり、別のデータベース内のテーブルから参照されているときは、実行できません。どのテーブルまたは外部データベースが、現在のデータベース内にあるプライマリ・キー・テーブルに外部キー制約を持っているかを調べるには、次のクエリを使用します。

```
select object_name(tableid), db_name(frgndbid)
from sysreferences
where frgndbid is not null
```

どのテーブルまたは外部データベースが、現在のデータベース内にある外部キー・テーブルにプライマリ・キー制約を持っているかを調べるには、次のクエリを使用します。

```
select object_name(reftabid), db_name(pmrydbid)
from sysreferences
where pmrydbid is not null
```

これらのテーブル内のデータベース間の制約を削除するには、`alter table` を使用します。次に、`sp_renamedb` を再実行します。

- データベース名を変更する場合は、次の手順に従います。
 - データベース名を含むすべてのストアド・プロシージャ、トリガ、ビューを削除します。
 - 削除したオブジェクトのソース・テキストを変更して、新しいデータベース名を表すようにします。
 - 削除したオブジェクトを再作成します。
 - `usedatabase_name` コマンドまたは完全な修飾子の一部 (`dbname.[owner].objectname` という形式) のどちらかでデータベースを参照しているすべてのアプリケーションと SQL ソース・スクリプトを変更します。
- スクリプトを使用してデータベース上で `dbcc` コマンド、または `dump database` と `dump transaction` コマンドを実行する場合、必ずそれらのスクリプトを更新してください。

警告！ 名前が変更されたオブジェクトに依存するプロシージャ、トリガ、ビューは、それらが再コンパイルされるまでは動作しません。`sp_renamedb` を実行する場合は、すべての従属オブジェクトの定義を変更してください。従属オブジェクトは、[sp_depends](#) を使って検索できます。

パーミッション `sp_renamedb` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、データベースに対する <code>own database</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド `create database`](#)
システム・プロシージャ [sp_changedbowner](#)、[sp_dboption](#)、[sp_depends](#)、[sp_helpdb](#)、[sp_rename](#)

sp_reportstats

説明 システム使用状況についての統計をレポートします。

構文 sp_reportstats [*loginname*]

パラメータ *loginname*

ユーザのログイン名です。このログイン名のユーザについて使用量の合計値を表示します。

例 1 すべての Adaptive Server ユーザに関する、現在の使用量の合計値についてのレポートを表示します。

```
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
```

例 2 ユーザ “kathy” の現在の使用量の合計値についてレポートを表示します。

```
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
```

使用方法

- sp_reportstats は、すべてのログインに関する現在の使用量の合計値と、各ログインの個別統計と全統計データに対する比率を表示します。sp_reportstats は、レポートを作成するアカウントのログイン名を唯一のパラメータとして受け取ります。パラメータが指定されなければ、sp_reportstats はすべてのアカウントについてレポートします。
- CPU の単位は、Adaptive Server のクロック・チックです。

- “probe” ユーザは、2 フェーズ・コミットの probe プロセス用に存在しています。このユーザは チャレンジ/応答機能を使用して Adaptive Server にアクセスします。

パーミッション `sp_reportstats` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_clearstats](#)、[sp_configure](#)

sp_restore_system_role

説明	システム標準の役割またはデータベース所有者をデフォルトの役割権限設定に戻します。
構文	<code>sp_restore_system_role [role_name [, all_dbs]]</code>
パラメータ	<p><i>role_name</i></p> <p>sa_role、sso_role、oper_role、replication_role、keycustodian_role、sa_serverprivs_role、dbo のいずれかです。パラメータを指定しないと、使用法についてのメッセージが表示されます。</p> <p><i>all_dbs</i></p> <p>すべてのオンライン・データベースでデータベース所有者または役割をデフォルトの役割権限設定に戻します。all_dbs を指定しないと、現在のデータベースでのみ変更が実行されます。</p>
例	<p>例 1 sso_role をすべてのデータベースでデフォルトの役割権限設定に戻します。</p> <pre>sp_restore_system_role sso_role, all_dbs</pre> <p>例 2 db1 でのみ sso_role をデフォルトの役割権限設定に戻します。</p> <pre>use db1 sp_restore_system_role sa_role</pre> <p>例 3 master で dbo をデフォルトの権限設定に戻します。</p> <pre>use master sp_restore_system_role dbo</pre>
使用法	<ul style="list-style-type: none">sp_restore_system_role は、システム標準の役割、ユーザ定義の役割 sa_serverprivs_role、またはデータベース所有者をデフォルトの役割権限設定に戻します。使用できるシステム標準の役割は、sa_role、sso_role、oper_role、replication_role、および keycustodian_role です。デフォルトの役割権限設定でこれらの役割またはデータベース所有者に付与されている権限のリストについては、『セキュリティ管理ガイド』の「細密なパーミッションの使用」を参照してください。all_dbs を指定しても、sybsecurity データベースにはリストア・オペレーションが適用されません。sybsecurity 内の役割またはデータベース所有者の権限は、必要に応じて手動でリストアしてください。

パーミッション `sp_restore_system_role` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効 | 細密なパーミッションが有効の場合、`sa_role` をリストアするには `manage server permissions` 権限を持つユーザであることが必要で、それ以外の役割またはデータベース所有者をリストアするには `manage security permissions` を持つユーザでなければならない。`all_dbs` オプションを使用するには `use any database` 権限も必要。

細密なパーミッションが無効 | 細密なパーミッションが無効の場合、`sa_role` をリストアするには `sa_role` が付与されたユーザであることが必要で、それ以外の役割またはデータベース所有者をリストアするには `sso_role` が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
81	dbcc 監査がオンの場合	dbcc コマンドの実行	<ul style="list-style-type: none"> • <i>Roles</i> — 現在のアクティブな役割 • <i>Keywords or options</i> — <code>upgd_grantrev_sysrole_perms</code> • <i>Previous value</i> — NULL • <i>Current value</i> — NULL • <i>Other information</i> — パラメータ・リスト • <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

sp_revokelogin

説明 (Windows NT のみ) 統合セキュリティ・モードまたは混合モード (名前付きパイプ使用) がアクティブなとき、Windows NT のユーザとグループから Adaptive Server の役割とデフォルトのパーミッションを取り消します。

構文 sp_revokelogin {login_name | group_name}

パラメータ login_name
Windows NT ユーザのネットワーク・ログイン名です。

group_name
Windows NT グループ名です。

例 **例 1** “jeanluc” という名前の Windows NT ユーザから、すべてのパーミッションを取り消します。

```
sp_revokelogin jeanluc
```

例 2 Windows NT の管理者グループから、すべての役割を取り消します。

```
sp_revokelogin Administrators
```

使用法

- Adaptive Server が統合セキュリティ・モードまたは混合モード (接続が名前付きパイプの場合) で実行されているときのみ、sp_revokelogin を使用します。Adaptive Server が標準モード、または (名前付きパイプ以外の接続を使用している) 混合モード で実行されている場合は、**revoke** コマンドを使用します。
- sp_revokelogin によってユーザの役割とデフォルトの権限を取り消した場合、そのユーザは **trusted** 接続を使用して Adaptive Server にログインすることができなくなります。

パーミッション sp_revokelogin のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 manage roles 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 sa_role が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [grant](#)、[revoke](#)、[setuser](#)

システム・プロシージャ [sp_droplogin](#)、[sp_dropuser](#)、[sp_logininfo](#)

sp_role

説明	Adaptive Server ログイン・アカウントに役割を付与または取り消します。
構文	<code>sp_role {"grant" "revoke"}, rolename, loginame</code>
パラメータ	<code>grant</code> <code>revoke</code> <code>loginame</code> に役割を付与するのか、または取り消すのかを指定します。 <code>rolename</code> 付与する、または取り消す役割です。 <code>loginame</code> 役割を付与する、または取り消す対象となるログイン・アカウントです。
例	“alexander” という名前のログイン・アカウントに、システム管理者の役割を付与します。 <pre>sp_role "grant", sa_role, alexander</pre>
使用法	<ul style="list-style-type: none">• <code>sp_role</code> を実行すると、Adaptive Server ログイン・アカウントへの役割の付与またはこのログイン・アカウントからの役割の取り消しが行われます。• 役割をユーザに付与した場合、その役割が有効になるのは次回そのユーザが Adaptive Server にログインしたときです。または、<code>set role</code> コマンドを使用すると、その役割をすぐに有効にすることができます。たとえば、次のコマンドを実行すると、ユーザにシステム管理者の役割を与えることができます。 <pre>set role sa_role on</pre>ただし、ログイン時に役割を有効にするには、<code>sp_modifylogin</code> を実行する必要があります。詳細については、sp_modifylogin を参照してください。• ユーザのログインしている間は、そのユーザから役割を取り消すことはできません。• ユーザがログインすると、ユーザに付与されているすべてのシステム定義の役割がアクティブ (<code>on</code>) になります。役割を無効にするには、<code>set</code> コマンドを使用します。たとえば、システム管理者の役割を無効にするには、次のコマンドを使用します。 <pre>set role "sa_role" off</pre>

パーミッション `sp_role` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage roles</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効化されている場合は、次のようになる。 <ul style="list-style-type: none"> • <code>sp_role</code> を実行してシステム管理者の役割を他のユーザに付与できるのは、システム管理者のみ。 • <code>sp_role</code> を実行して <code>sp_role</code> 以外の役割を他のユーザに付与できるのは、システム・セキュリティ担当者のみ。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド `grant`、`revoke`、`set`](#)
[関数 `proc_role`、`show_role`](#)
システム・プロシージャ [`sp_activeroles`](#)、[`sp_displayroles`](#)、[`sp_displayroles`](#)

sp_securityprofile

説明 ログイン・プロファイルに関連付けられた属性またはバインドをリストする。

構文

```
sp_securityprofile 'attributes','login profile',
    {wildcard | login_profile_name | 'default'}

sp_securityprofile 'bindings', 'login profile'[, {wildcard | login_profile_name |
'default'}
    [, 'login' ,{wildcard | login_name}]]
```

sp_securityprofile 'help'

パラメータ

attributes

ログイン・プロファイルに関連付けられた属性またはバインドをリストするように指定する。

login profile

ログイン・プロファイルについての情報を入手するように指定する。

bindings

`login` が指定されると、ログイン・アカウントのバインドがリストされる。`login profile` が指定されると、ログイン・プロファイルのバインドがリストされる。

login

ログイン・アカウントについての情報を入手するように指定する。

wildcard | login_profile_name | default

情報を入手するログイン・プロファイルを指定する。オプションには固有のログイン・プロファイルの名前が含まれ、デフォルトのログイン・プロファイルやワイルドカードを使用してログイン・プロファイルを識別することができる。

wildcard | login_name

固有のログイン・アカウント名を使用したり、ワイルドカードを使用してログインアカウントを特定できるように指定する。

help

使用方法を表示する。

例 1 デフォルトのログイン・プロファイルのすべての属性をリストします。

```
sp_securityprofile 'attributes', 'login profile',
    'default'
```

Name	Value
-----	-----
login profile	def_login_profile

```

default                                yes
default database                       master
default language                       NULL
login script                           NULL
auto activated roles                   emp_role
auto activated roles                   def_role
manually activated roles               special_role
authenticate with                       ANY
track lastlogin                        TRUE
stale period                            180D

```

例2 すべてのログイン・プロファイルに関連付けられたすべての属性を表示します。

```
sp_securityprofile 'attributes', 'login profile', '%'
```

```

Name                                Value
-----
login profile                       def_login_profile
default                              yes
default database                    master
default language                    NULL
login script                        NULL
auto activated roles                emp_role
auto activated roles                def_role
authenticate with                    ANY
track lastlogin                     TRUE
stale period                         180D

```

```

Name                                Value
-----
login profile                       eng_login_profile
default                              work
default database                    engr_script
login script                        emp_role
auto activated roles                def_role
auto activated roles                engr_role
authenticate with                    LDAP

```

```

Name                                Value
-----
login profile                       mgr_login_profile
default                              work
default database                    mgr_script
login script                        emp_role
auto activated roles

```

```

auto activated roles      def_role
auto activated roles      mgr_role
manually activated roles  activate_emp_role
authenticate with         LDAP

```

```

Name                      Value
-----
login profile              sa_login_profile
manually activated roles  admin_role
default

```

例 3 指定のログイン・プロファイルに関連付けられているすべてのログイン・アカウントを表示します。

```

sp_securityprofile 'bindings', 'login profile',
                  'engr_login_profile'

```

```

Login name                Login profile name
-----
anderson                  eng_login_profile
gupta                    eng_login_profile
lchang                   eng_login_profile
tsato                    eng_login_profile

```

例 4 sa というログイン・アカウントのログイン・プロファイルを表示します。

```

sp_securityprofile 'bindings', 'login profile', null,
                  'login', 'sa'

```

```

Login name                Login profile name
-----
sa                        sa_login_profile

```

使用法

- プロファイルに設定されていない属性では、優先度の規則に従います。詳細については、以下を参照してください。

パーミッション

sp_securityprofile のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効

細密なパーミッションが有効の場合、manage any login profile 権限を持つユーザであることが必要。

細密なパーミッションが無効	<p>細密なパーミッションが無効の場合、すべてのログイン・プロファイルの属性とバインドを表示するには、<code>sso_role</code> が付与されたユーザであることが必要。</p> <p>権限のないログイン・アカウントの場合、次のようになる。</p> <ul style="list-style-type: none"> • ログインに関連付けられているログイン・プロファイルの属性のみを表示できる (直接またはデフォルトのログイン・プロファイル)。 • ログイン・プロファイルとログイン・アカウントのバインドは表示できない。
---------------	--

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 **コマンド** `creat login profile`、`alter login profile`

マニュアル『セキュリティ管理ガイド』の「ログイン・プロファイルとパスワード・ポリシー属性の適用」

システム・プロシージャ `sp_displaylogin`

sp_sendmsg

説明	UDP (User Datagram Protocol) ポートへメッセージを送信します。
構文	<code>sp_sendmsg ip_address, port_number, message</code>
パラメータ	<i>ip_address</i> UDP アプリケーションが稼働しているマシンの IP アドレスです。 <i>port_number</i> UDP ポートのポート番号です。 <i>message</i> 送信されるメッセージです。最大 4096 文字です。
例	<pre>sp_sendmsg "120.10.20.5", 3456, "Hello World"</pre>
使用方法	<ul style="list-style-type: none">• <code>sp_sendmsg</code> は、Windows NT ではサポートされていません。• UDP のメッセージ機能を使用できるようにするには、システム・セキュリティ担当者が <code>allow sendmsg</code> 設定パラメータを 1 に設定する必要があります。• セキュリティ検査は <code>sp_sendmsg</code> を使用して実行することはできません。ネットワーク間で機密情報を送信するために <code>sp_sendmsg</code> を使用するときは、警告文の表示を強くおすすめします。警告文を表示させることによって、ユーザは、このメッセージ機能を使用した結果に発生するあらゆるセキュリティ上の問題を承認することになります。• このサンプル C プログラムは、指定するポートで受信し、受け取ったメッセージをエコー (返送) します。以下に、上記の例で呼び出された <code>sp_sendmsg</code> を受信する場合のサンプル・コードを示します。

```
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(  填 sage: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);
}
}

```

パーミッション

sp_sendmsg は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

関数 [syb_sendmsg](#)

sp_serveroption

説明 リモート・サーバ・オプションを表示または変更します。

構文 sp_serveroption [server, optname, optvalue]

パラメータ

server

オプションを設定するリモート・サーバの名前です。

optname

設定される、または設定が解除されるオプションの名前です。表 1-30 では、オプション名をリストしています。

表 1-30: sp_serveroption オプション

オプション	意味
mutual authentication	Kerberos 認証を使用してリモート・サーバへのすべての接続の相互認証を設定する。
external engine auto start	Adaptive Server が起動されるたびに EJB サーバも起動するように指定する。デフォルトは true。Adaptive Server を起動すると EJB サーバも起動する。
net password encryption	リモート・サーバとの接続を開始するために、クライアント側のパスワード暗号化のハンドシェイク方式を使用するか、または通常の (暗号化されないパスワード) ハンドシェイク方式を使用するかを指定する。デフォルトは false。ネットワークの暗号化は行われない。

オプション	意味
net password encryption reqd	<p>Adaptive Server では非対称暗号化が使用でき、RSA パブリック・キー暗号化アルゴリズムを使用してクライアントからサーバへパスワードを安全に転送できる。Adaptive Server は非対称キーのペアを生成して、新しいログイン・プロトコルを使用するクライアントにそのパブリック・キーを送信する。クライアントはパブリック・キーを使用してユーザのログイン・パスワードを暗号化してからサーバに送信する。サーバはプライベート・キーを使用してパスワードを解読し、接続しようとしているクライアントの認証を開始する。</p> <p>Adaptive Server がクライアントにこのプロトコルの使用を要求するように設定する。Adaptive Server の設定パラメータ net password encryption reqd を設定して、ユーザ名とパスワードに基づくすべての認証で RSA 非対称暗号化の使用を要求するようにはできる。net password encryption reqd の有効な値を次に示す。</p> <ul style="list-style-type: none"> • 0 - クライアントがネットワーク・ログイン・パスワードに使用する暗号化アルゴリズムを選択できる。暗号化しないオプションもある。これがこの設定パラメータのデフォルト値である。これは旧バージョンに最も近い機能を提供する。これはネットワーク・パスワード暗号化の選択肢をクライアント・アプリケーションが確立できるようにする。 • 1 - ネットワーク・ログイン・パスワードの暗号化に、クライアントが RSA または Sybase 独自の暗号化アルゴリズムのみを使用するように制限する。これは、古いクライアントには Sybase 独自のアルゴリズムを使用した接続を、新しいクライアントには強力な RSA アルゴリズムを使用した接続を可能にする。パスワードの暗号化を使用しないで接続しようとするクライアントは接続できない。 • 2 - ネットワーク・ログイン・パスワードの暗号化に、クライアントが RSA 暗号化アルゴリズムのみを使用するように制限する。これには強力な RSA パスワード暗号化を使用する新しいクライアントが必要である。RSA 暗号化を使用しないで接続しようとするクライアントは接続できない。
allow password downgrade	
readonly	このオプションは、コンポーネント統合サービスでのみ有効。指定したサーバへのアクセスは読み込み専用であることを指定する。
security mechanism	このオプションでは、リモート・サーバのセキュリティ・メカニズムを指定する。ログインが Kerberos メカニズムで認証される場合に、リモート・サーバへの接続に対して Kerberos 認証を有効にする。
server cost	このオプションは、コンポーネント統合サービスでのみ有効。ユーザによって制御される 1 回の交換にかかるコストを、サーバごとに指定する。詳細については、『コンポーネント統合サービス・ユーザーズ・ガイド』の「第 2 章 コンポーネント統合サービスの概要」を参照。

オプション	意味
server logins	(このオプションは、コンポーネント統合サービスでのみ有効) リモート・ログインを完全にサポートするために、Client-Library には CIS からのサーバ接続要求を可能にする接続プロパティが用意されている。この接続は、受信サーバで (通常のクライアント接続ではなく) サーバ接続として認識される。これにより、リモート・サーバは、接続がサイト・ハンドラによって作成されたかのように、sysremotelogins を使用して接続を検証できるようになる。これを有効にすると、Omni は、CS_LOGIN_TYPE 接続プロパティを使用し、タイプを LREMUSER に設定して指定のサーバに接続する。また、リモート・サーバが Adaptive Server の場合、CS_LOGIN_REMOTE_SERVER プロパティにローカル・サーバ名の値が設定され、リモート・パスワードは ct_remote_pwd() を使用して設定される。
server principal	リモート・サーバのサーバ・プリンシパル名を設定する。
negotiated logins	このオプションは、コンポーネント統合サービスでのみ有効。このオプションは、XP Server または Backup Server への CIS 接続が必要な場合には必須。有効にすると、Omni は、CS_SEC_CHALLENGE プロパティを使用して指定のサーバに接続し、XP Server と Backup Server からのログイン・チャレンジに適切に応答できるコールバック・ハンドラを確立する。
timeouts	設定を解除 (false) すると、ローカル・サーバが使用している通常のタイムアウト・コードが無効になる。したがって、論理コネクションなしの状態が 1 分間続いても、サイト接続ハンドラは物理コネクションを自動的に切断しない。デフォルトは false です。
use message confidentiality	すべての接続のメッセージの機密性を Kerberos 認証を使用したリモート・サーバに設定する。
use message integrity	すべての接続のメッセージの整合性を Kerberos 認証を使用したリモート・サーバに設定する。
cis hafailover	このオプションは、コンポーネント統合サービスでのみ有効。有効にした場合、接続が失敗したときに自動フェールオーバーを使用するよう Open Client に指示する。その場合、CIS 接続の失敗は、ディレクトリ・サービス (interface ファイルと LDAP サーバなど) で指定されたサーバをフェールオーバー・サーバとして自動的にフェールオーバーされる。

Adaptive Server は、オプション名に含まれるあらゆるユニークな文字列を認識します。オプション名に埋め込みブランクを使用する場合は、そのオプション名を引用符で囲みます。

optvalue

true (on) または false (off) です。security mechanism オプション以外のすべてのオプションで指定します。

security mechanism オプションでは、セキュリティ・メカニズムの名前を指定します。サーバ上で使用できるセキュリティ・メカニズムの名前を調べるには、次のコマンドを実行してください。

```
select * from syssecmechs
```

例

例 1 サーバ・オプションのリストを表示します。

```
sp_serveroption

Settable server options.
-----
cis hafailover
enable login redirection
external engine auto start
incompatible sort order
mutual authentication
negotiated logins
net password encryption
readonly
relocated joins
security mechanism
server cost
server logins
server principal
timeouts
use message confidentiality
use message integrity
```

例 2 GATEWAY リモート・サーバとの物理接続がアクティブでない場合でも、タイムアウトしないように、サーバに通知します。

```
sp_serveroption GATEWAY, "timeouts", false
```

例 3 GATEWAY リモート・サーバに接続するときに、送信されるパスワードの暗号化に使用する暗号化キーを GATEWAY が返送するように指定します。

```
sp_serveroption GATEWAY, "net password encryption", true
```

例 4 Adaptive Server が起動されるたびに EJB サーバの SYB_EJB も起動するように指定します。

```
sp_serveroption SYB_EJB, "external engine auto start", true
```

例 5 リモート・サーバ S2 への接続の Kerberos 認証を指定します。

```
sp_serveroption S2, "security mechanism", csfkrb5
```

例 6 Kerberos 認証を使用してリモート・サーバへのすべての接続の相互認証を指定します。

```
sp_serveroption TEST3, "mutual authentication", true
```

例 7 自動起動を無効にします。SYB_EJB は EJB サーバの論理名を表します。

```
sp_serveroption 'SYB_EJB', 'external engine auto start', 'false'
```

自動起動を有効にするには、次のように入力します。

```
sp_serveroption 'SYB_EJB', 'external engine auto start', 'true'
```

`external engine auto start` の使用方法の詳細については、『EJB Server ユーザーズ・ガイド』の「第 2 章 概要」の「EJB サーバを自動的に起動する」を参照してください。

使用法

- ユーザが設定可能なサーバ・オプションのリストを表示するには、パラメータなしで `sp_serveroption` を実行します。
- `timeouts` を `false` に設定すると、2 つのサーバのうちのどちらかのサーバが停止するまでサイト・ハンドラが実行を続けます。
- `net password encryption` オプションを使用すると、クライアントは、リモート・プロシージャ・コールの開始時に、ネットワーク上にプレーン・テキスト形式と暗号化形式のどちらでパスワードを送信するかを指定することができます。`net password encryption` を `true` に設定すると、パスワードを入力しなくても初期ログイン・パケットが送信されます。クライアントは暗号化が必要であることをリモート・サーバに通知します。リモート・サーバが暗号化キーを返送します。クライアントは、この暗号化キーを使用してパスワードを暗号化します。クライアントがこのパスワードを暗号化した後、リモート・サーバは、パスワードを受信したときにこのキーを使用して認証します。
- 特定の `isql` セッションに使用するネットワーク・パスワードの暗号化を設定する場合は、`isql` のコマンド・ライン・オプションを使用できます。詳細については、『ASE ユーティリティ・ガイド』を参照してください。
- リリース 10.0 より前の `SQL Server` に接続する場合、`net password encryption` オプションは使用できません。
- `security mechanism`、`mutual authentication`、`use message confidentiality`、`use message integrity` の各オプションは、Kerberos ログインにのみ適用されます。

パーミッション

`sp_serveroption` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効

細密なパーミッションが有効の場合、`manage server` 権限を持つユーザであることが必要。共有ディスク・クラスタの場合は、`manage server` 権限と `manage cluster` 権限を持つユーザであることが必要。

すべてのユーザは、オプション・リストを表示させるために、パラメータなしで `sp_serveroption` を実行することができる。

細密なパーミッションが無効

細密なパーミッションが無効の場合、`timeouts` オプションを設定するには `sa_role` が付与されたユーザであることが必要。
 次のオプションを設定するには、`sso_role` が付与されたユーザであることが必要。

- net password encryption
- security mechanism
- 相互認証
- use message confidentiality
- use message integrity

すべてのユーザは、オプション・リストを表示させるために、パラメータなしで `sp_serveroption` を実行することができる。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル サーバ・オプションの詳細については、『システム管理ガイド』を参照してください。

システム・プロシージャ [sp_helpserver](#)、[sp_password](#)

ユーティリティ `isql`

sp_set_qplan

説明	対応するクエリを変更しないで、既存プランの抽象プランのテキストを変更します。
構文	<code>sp_set_qplan id, plan</code>
パラメータ	<p><i>id</i> 抽象プランの ID です。</p> <p><i>plan</i> 新しい抽象プランです。</p>
例	<pre>sp_set_qplan 563789159, "(g_join (scan t1) (scan t2))"</pre>
使用法	<ul style="list-style-type: none"> 既存プランの抽象プランを変更するには、<code>sp_set_qplan</code> を使用します。プランには最大 255 文字を指定できます。抽象プランが 255 文字を超える場合は、<code>sp_drop_qplan</code> によって古いプランを削除し、<code>create plan</code> によってクエリの新しいプランを作成します。 <code>sp_set_qplan</code> でプランを変更すると、そのプランは、抽象プラン構文に準拠しているかどうかの確認も SQL テキストとの互換性の確認も行われません。<code>sp_set_qplan</code> によって変更したプランはすべて、直ちに指定の ID のクエリを実行することによって、その正当性を確認する必要があります。 プランの ID を検索するには、<code>sp_help_qpgroup</code>、<code>sp_help_qplan</code>、または <code>sp_find_qplan</code> を使用します。また、プラン ID は <code>create plan</code> によって返され、<code>showplan</code> の出力に含まれます。
パーミッション	<code>sp_set_qplan</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	<p>細密なパーミッションが有効の場合、<code>manage abstract plans</code> 権限を持つユーザであることが必要。</p> <p>どのユーザも <code>sp_set_qplan</code> を実行して、自分が所有するプランのテキストを変更できる。</p>
細密なパーミッションが無効	<p>細密なパーミッションが無効の場合、データベース所有者か <code>sa_role</code> が付与されたユーザであることが必要。</p> <p>どのユーザも <code>sp_set_qplan</code> を実行して、自分が所有するプランのテキストを変更できる。</p>

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 [コマンド create plan](#)

システム・プロシージャ [sp_drop_qpgroup](#)、[sp_drop_qplan](#)、[sp_find_qplan](#)、[sp_help_qplan](#)

sp_setlangalias

説明	代替言語のエイリアスの割り当てまたは変更を行います。
構文	<code>sp_setlangalias language, alias</code>
パラメータ	<p><i>language</i> 代替言語の公式言語名です。</p> <p><i>alias</i> 代替言語の新しいローカル・エイリアスです。</p>
例	<pre>sp_setlangalias french, francais</pre> <p>このコマンドは、公式言語名 “french” に対してエイリアス名 “francais” を割り当てます。</p>
使用法	<ul style="list-style-type: none"> <i>alias</i> は、公式名に使用される <code>syslanguages.alias</code> の現在値を置き換えます。 <code>set language</code> コマンドは、公式言語名の代わりに、新しい <i>alias</i> を使用できます。
パーミッション	<code>sp_setlangalias</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage server</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — すべての入力パラメータ <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 **コマンド** `set`

システム・プロシージャ [sp_addlanguage](#)、[sp_droplanguage](#)、[sp_helplanguage](#)

sp_setpglockpromote

説明	データベース、テーブル、または Adaptive Server に対するロック・プロモーション・スレッシュホールドを設定または変更します。
構文	<pre>sp_setpglockpromote {"database" "table"}, objname, new_lwm, new_hwm, new_pct sp_setpglockpromote server, NULL, new_lwm, new_hwm, new_pct</pre>
パラメータ	<p>server ロック・プロモーション・スレッシュホールドに使用するサーバワイドな値を設定します。</p> <p>"database" "table" ロック・プロモーション・スレッシュホールドをデータベース用に設定するか、またはテーブル用に設定するかを指定します。“database”と“table”は Transact-SQL のキーワードのため、引用符で囲んでください。</p> <p>objname テーブルまたはデータベースのロック・プロモーション・スレッシュホールドを設定する場合は、そのテーブルまたはデータベースの名前を指定します。サーバワイドな値を設定する場合は、null を指定します。</p> <p>new_lwm LWM (ロー・ウォーターマーク) のスレッシュホールドに設定する値を指定します。LWM には、HWM (ハイ・ウォーターマーク) 以下の値を設定してください。LWM の最小値は 2 です。このパラメータは null でもかまいません。</p> <p>new_hwm ロック・プロモーションの HWM スレッシュホールドに設定する値を指定します。HWM には、LWM 以上の値を設定してください。LWM の最小値は 2,147,483,647 です。このパラメータは null でもかまいません。</p> <p>new_pct ロック・プロモーション・パーセンテージ (PCT) スレッシュホールドに設定する値を指定します。PCT は、1 ~ 100 の間にしてください。このパラメータは null でもかまいません。</p>
例	<p>例 1 サーバワイドなロック・プロモーションの LWM を 200 に設定し、HWM を 300 に、PCT を 50 に設定します。</p> <pre>sp_setpglockpromote "server", NULL, 200, 300, 50</pre>

例 2 master データベースに使用されるロック・プロモーション・スレッシュホールドを設定します。

```
sp_setpglockpromote "database", master, 1000, 1100, 45
```

例 3 pubs2 データベース内の titles テーブルに使用されるロック・プロモーション・スレッシュホールドを設定します。このコマンドは、pubs2 データベースから発行してください。

```
sp_setpglockpromote "table", "pubs2..titles", 500, 700, 10
```

例 4 master データベースに使用される HWM スレッシュホールドを 1600 に変更します。スレッシュホールドは、sp_setpglockpromote によってあらかじめ設定されています。このコマンドは、master データベースから発行してください。

```
sp_setpglockpromote "database", master, @new_hwm=1600
```

使用法

- データベースレベルのロック・プロモーションを表示するには、sp_helpdb *dbname* を使用します。テーブル・レベルのロックを表示するには、sp_helpdb *tablename* を使用します。
- sp_setpglockpromote は、テーブル、データベース、または Adaptive Server に使用されるロック・プロモーションの値を設定します。

Adaptive Server は、ロック数がロック・プロモーション・スレッシュホールドを超えるまで、テーブル上のページ・ロックを取得しています。sp_setpglockpromote はオブジェクト、データベース、またはサーバに使用されるロック・プロモーション・スレッシュホールドを変更します。Adaptive Server がテーブル・ロックを正常に取得すると、ページ・ロックは解放されます。

テーブルに対するロックの数が HWM スレッシュホールドを超えると、Adaptive Server はテーブル・ロックに拡大します。テーブルに対するロックの数が LWM 未満のときには、Adaptive Server はテーブル・ロックを拡大をしません。テーブルに対するロックの数が HWM と LWM の間であり、ロックの数が PCT スレッシュホールドを超えた場合は、Adaptive Server はテーブル・ロックに拡大しようとしています。

- テーブルに使用されるロック・プロモーション・スレッシュホールドは、データベースまたはサーバワイドな設定を上書きします。データベースに使用されるロック・プロモーション・スレッシュホールドは、サーバワイドな設定を上書きします。

- Adaptive Server のロック・プロモーション・スレッシュホールドを初期化する必要はありませんが、データベースとテーブルのロック・プロモーション・スレッシュホールドは、`sp_setpglockpromote` で LWM、HWM、PCT を指定して初期化してください。
`sp_setpglockpromote` は、データベースまたはテーブル用に初めて実行されたときに、`sysattributes` 内にそのオブジェクトに対応するローを作成します。スレッシュホールドの初期化が終わると、例 4 のようにスレッシュホールドを個別に修正できます。
- テーブルまたはデータベースの場合、`sp_setpglockpromote` は、LWM、HWM、PCT を 1 回のトランザクションで設定します。
`sp_setpglockpromote` で値を更新している間にエラーが発生した場合、すべての変更がアボートされてトランザクションがロールバックされます。サーバワイドなスレッシュホールドを変更する場合、1 つまたは複数のスレッシュホールドの更新に失敗し、残りのスレッシュホールドの更新は成功する場合があります。値の更新に失敗した場合、Adaptive Server はエラー・メッセージを返します。
- ロック・プロモーション・スレッシュホールドのサーバワイドな設定値を参照するには、`sp_configure "lock promotion"` を使用して、3 つすべてのスレッシュホールドの値を確認します。データベースに対するロック・プロモーションの設定値を表示させるには、`sp_helpdb` を使用します。テーブルに使用されるロック・プロモーションの設定値を表示するには、`sp_help` を使用します。

パーミッション

`sp_setpglockpromote` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage lock promotion threshold</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_configure](#)、[sp_droplockpromote](#)、[sp_help](#)、[sp_helpdb](#)

sp_setpsexex

説明	セッションがアクティブなときに、セッションのカスタム実行属性を設定します。
構文	<code>sp_setpsexex <i>spid</i>, <i>exeattr</i>, <i>value</i></code>
パラメータ	<p><i>spid</i> 実行変数を設定するセッションの ID です。 <i>spid</i> を表示するには <code>sp_who</code> を使用します。</p> <p><i>exeattr</i> 設定する実行属性を指定します。値は <code>priority</code> と <code>enginegroup</code> です。</p> <p><i>value</i> <i>exeattr</i> の新しい値です。それぞれの属性の値を次に示します。</p> <ul style="list-style-type: none">• <i>exeattr</i> が <code>priority</code> の場合、<i>value</i> は HIGH、MEDIUM、または LOW です。• <i>exeattr</i> が <code>enginegroup</code> の場合、<i>value</i> は既存のエンジン・グループの名前です。
例	<p>この例では、ID が 1 のプロセスの優先度を HIGH に設定します。</p> <pre>sp_setpsexex 1, "priority", "HIGH"</pre>
使用法	<ul style="list-style-type: none">• <code>sp_setpsexex</code> によって指定された実行属性値は、現在のセッションでのみ有効で、セッションが終了した後は適用されません。• <code>sp_setpsexex</code> は慎重に使用してください。誤って使用すると、パフォーマンスが低下することがあります。プロセスが CPU タイムを取得できない場合は、<code>sp_setpsexex</code> を使用してその場で属性を変更すると効果的な場合があります。しかし、パフォーマンスの問題がロックなどの別の原因によるものである場合は、実行属性を変更することで問題が悪化することがあります。• 実行属性はセッションにしか設定できないため、ワーカー・プロセス <i>spid</i> には <code>sp_setpsexex</code> を設定できません。• ハウスキーピング <i>spid</i> 以外は、システム <i>spid</i> に実行属性を設定できません。• 対応するエンジン・グループにオンラインのエンジンがない場合、<code>sp_setpsexex</code> は正常に機能しません。

パーミッション `sp_setpsex` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any execution class</code> 権限を持つユーザであることが必要。 すべてのユーザは、そのユーザが所有しているプロセスの優先度を低くするために <code>sp_setpsex</code> を実行できます。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。 すべてのユーザは、そのユーザが所有しているプロセスの優先度を低くするために <code>sp_setpsex</code> を実行できます。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ `sp_addexclass`、`sp_bindexclass`、`sp_dropexclass`、`sp_showexclass`

sp_setrowlockpromote

説明	データロー・ロックされたテーブル、データベース内のすべてのデータロー・ロックされたテーブル、またはサーバ上のすべてのデータロー・ロックされたテーブルに使用される、ローロック・プロモーション・スレッシュホールドを設定または変更します。
構文	<pre>sp_setrowlockpromote "server", NULL, new_lwm, new_hwm, new_pct sp_setrowlockpromote {"database" "table"}, objname, new_lwm, new_hwm, new_pct</pre>
パラメータ	<p>server ロー・ロック・プロモーション・スレッシュホールドに使用するサーバワイドな値を設定します。</p> <p>"database" "table" データベースまたはテーブルのどちらにローロック・プロモーション・スレッシュホールドを設定するかを指定します</p> <p>objname テーブルまたはデータベースに対してローロック・プロモーション・スレッシュホールドを設定する場合は、そのテーブルまたはデータベースの名前を指定します。サーバワイドな値を設定する場合は <code>null</code> を指定します。</p> <p>new_lwm LWM (ロー・ウォータマーク) のスレッシュホールドに設定する値を指定します。LWM には、HWM (ハイ・ウォータマーク) 以下の値を設定してください。LWM の最小値は 2 です。このパラメータは <code>null</code> でもかまいません。</p> <p>new_hwm HWM (ハイ・ウォータマーク) スレッシュホールドに設定する値を指定します。HWM には、LWM 以上の値を設定してください。LWM の最小値は 2,147,483,647 です。このパラメータは <code>null</code> でもかまいません。</p> <p>new_pct ロック・プロモーション・パーセンテージ (PCT) スレッシュホールドに設定する値を指定します。PCT は、1 ~ 100 の間にしてください。このパラメータは <code>null</code> でもかまいません。</p>
例	<p>例 1 engdb データベース内のすべてのデータローロック・テーブルのロー・ロック・プロモーション値を設定します。</p> <pre>sp_setrowlockpromote "database", engdb, 400, 400, 95</pre>

例 2 sales テーブルに使用されるロー・ロック・プロモーション値を設定します。

```
sp_setrowlockpromote "table", sales, 250, 250, 100
```

使用法

- データベースレベルのロック・プロモーションを表示するには、`sp_helpdb dbname` を使用します。テーブル・レベルのロックを表示するには、`sp_helpdb tablename` を使用します。
- `sp_setrowlockpromote` は、テーブル、データベース、または Adaptive Server のローロック・プロモーション・スレッショルドを設定または変更します。

Adaptive Server は、ロック数がロック・プロモーション・スレッショルドを超えるまで、データローロック・テーブル上でロー・ロックを取得しています。Adaptive Server がテーブル・ロックを正常に取得すると、ロー・ロックは解除されます。

テーブルに対するロー・ロックの数が HWM スレッショルドを超えると、Adaptive Server はテーブル・ロックを段階的に拡大しようとし、テーブルに対するロー・ロックの数が LWM 未満のときは、Adaptive Server はテーブル・ロックを拡大しようとはしません。テーブルに対するロー・ロックの数が HWM 値と LWM 値の範囲内であっても、テーブル内でのロー・ロックのパーセンテージ値が PCT スレッショルドを超えると、Adaptive Server はテーブル・ロックを拡大しようとし、

- ロックの拡大は常に 2 層で行われます。つまり、ロー・ロックはテーブル・ロックに拡大されます。Adaptive Server では、ロー・ロックからページ・ロックへの拡大は行われません。
- テーブルに使用されるロック・プロモーション・スレッショルドは、データベースまたはサーバワイドな設定を上書きします。データベースに使用されるロック・プロモーション・スレッショルドは、サーバワイドな設定を上書きします。
- データベースに使用されるロック・プロモーション・スレッショルドを変更するには、`master` データベースを使用してください。データベース内のテーブルに使用されるロック・プロモーション・スレッショルドを変更するには、テーブルがあるデータベースを使用してください。

- サーバワイドなロー・ロック・プロモーション・スレッシュールドは、[sp_configure](#) によって設定することもできます。
`sp_setrowlockpromote` を使用して値をサーバワイドな値に変更すると、設定パラメータが変更されて、設定ファイルが保存されます。初めて **Adaptive Server** をインストールするときに、サーバワイドなロー・ロック・プロモーション・スレッシュールドは、次のような設定パラメータによって設定されます。

row lock promotion HWM	200
row lock promotion LWM	200
row lock promotion PCT	100

詳細については、『システム管理ガイド』を参照してください。

- システム・プロシージャ [sp_sysmon](#) は、ロー・ロック・プロモーションについてレポートします。
- データベースレベルのロー・ロック・プロモーション・スレッシュールドは、`master..sysattributes` テーブル内に格納されます。データベースをダンプして、それを別のサーバにのみロードする場合は、ロー・ロック・プロモーション・スレッシュールドを新しいサーバに設定してください。オブジェクトレベルのロー・ロック・プロモーション・スレッシュールドは、ユーザ・データベース内の `sysattributes` テーブルに格納されます。また、ダンプに組み込まれます。

パーミッション

`sp_setrowlockpromote` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage lock promotion threshold</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_configure](#)、[sp_droprolockpromote](#)、[sp_helpdb](#)、[sp_sysmon](#)

sp_setsuspect_granularity

説明 ユーザ・データベースのリカバリ・フォールト・アイソレーション・モードを表示または設定します。このモードは、リカバリによってデータの破損が検出された場合に、どう処置するかを決定します。

構文 `sp_setsuspect_granularity [dbname
[, "database" | "page" [, "read_only"]]]`

パラメータ *dbname*
リカバリ・フォールト・アイソレーション・モードを表示または設定するデータベースの名前です。表示用のデフォルトは、現在のデータベースです。設定するには、**master** データベースの中から、ターゲット *dbname* を選択してください。

database
リカバリ・プロセスによって、データが損傷した疑いが検出された場合、疑わしいデータベース全体に **suspect** マークを付け、このデータベースにアクセスできないようにします。

page
リカバリによって、データベース内のデータが損傷した疑いが検出された場合に、損傷した疑いのあるページだけにマークを付け、このページにアクセスできないようにします。他のデータにはアクセスできます。

read_only
これを指定すると、リカバリによって疑わしいページにマークが付けられたときに、データベース全体に **read only** のマークが付けられます。

例 **例 1** 現在のデータベースのリカバリ・フォールト・アイソレーション・モードを表示します。

```
sp_setsuspect_granularity
DB Name  Cur.Suspect Gran.  Cfg.Suspect Gran.  Online mode
-----  -
pubs2    database           database            read/write
```

例 2 `pubs2` データベースの現在のリカバリ・フォールト・アイソレーション・モードと、新しく設定されたリカバリ・フォールト・アイソレーション・モードを表示します。

```
sp_setsuspect_granularity pubs2
```

例 3 次回 `pubs2` データベースでリカバリを実行したときに損傷したページが検出された場合は、疑わしいページだけをオフラインにして、データベースの残りの部分はオンラインにします。

```
sp_setsuspect_granularity pubs2, "page"
```

```
DB Name          Cur.Suspect Gran.Cfg.
```

```
-----
pubs2            database          database
sp_setsuspect_granularity:The new values will become effective
during the next recovery of the database ' pubs2' .
```

例 4 次回 pubs2 データベースでリカバリを実行したときに損傷したページが検出された場合は、疑わしいページだけをオフラインにして、データベースの残りの部分を読み込み専用モードでオンラインにします。

```
sp_setsuspect_granularity pubs2, "page", "read_only"
```

例 5 次回 pubs2 データベースでリカバリを実行したときに損傷したページが検出された場合は、データベース全体に疑わしいことを示す suspect マークを付け、オフラインにします。

```
sp_setsuspect_granularity pubs2, "database"
```

使用法

- `sp_setsuspect_granularity` は、リカバリ・フォールト・アイソレーション・モードを表示し、設定します。このモードは、要求したデータが損傷していることを検出したときに、リカバリがデータベース全体を疑わしいものとしてマーク付けするか、損傷したページだけを疑わしいものとしてマークを付けるかを決定します。詳細については、『システム管理ガイド』を参照してください。
- ユーザ・データベースのデフォルトのリカバリ・フォールト・アイソレーション・モードは、“database”です。リカバリ・フォールト・アイソレーション・モードを設定できるのは、ユーザ・データベースだけです。システム・データベースにはこのモードは設定できません。
- Cluster Edition では、`sp_setsuspect_granularity` で指定できるのは database オプションのみです。
- リカバリ・フォールト・アイソレーション・モードの設定は、master データベース内から実行してください。
- 損傷したために疑わしいものとして suspect マークが付けられたデータは、Adaptive Server を起動しても、そのままの状態が維持されます。サーバを再起動した後でも、疑わしいものとしてマークが付けられたページは、オフラインのままです。

- データベースの一部または全体に、疑わしいことを示す suspect マークが付けられた場合は、システム管理者が、[sp_forceonline_db](#) や [sp_forceonline_page](#) プロシージャによって疑わしいデータにアクセスできるようにしない限り、ユーザは疑わしいデータにアクセスできません。
- データベース・ログが損傷したり、他のページも共有しているリソースが利用できない場合など、データベース全体に関わる損傷が検出された場合は、リカバリ・フォールト・アイソレーション・モードが “page” に指定されていても、データベース全体に疑わしいことを示すマークが付けられます。
- `page` や `database` を指定しない場合、Adaptive Server は現在の設定と新しく構成された設定を表示します。現在の設定とは、データベースで前回にリカバリが実行された際に有効だった設定のことです。新しく構成された設定とは、データベースで次回にリカバリが実行されるときに有効になる設定のことです。
- データベースが `read_only` モードでオンラインになっている場合は、どのユーザもデータを変更できません。たとえ疑わしいページから影響を受けないオンラインのデータでも、ユーザは変更できません。ただし、システム管理者が、[sp_dboption](#) システム・プロシージャを使用して、`read only` を `false` に設定すれば、データベースを書き込み可能にすることができます。この場合、ユーザはオンラインでデータを変更できますが、依然として疑わしいデータにはアクセスできません。

パーミッション

`sp_setsuspect_granularity` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	<p>細密なパーミッションが有効の場合、リカバリ・フォールト・アイソレーション・モードを設定するには、指定したデータベースに対して <code>own database</code> 権限を持つユーザであることが必要。</p> <p>どのユーザも <code>sp_setsuspect_granularity</code> を実行して設定内容を表示できる。</p>
細密なパーミッションが無効	<p>細密なパーミッションが無効の場合、リカバリ・フォールト・アイソレーション・モードを設定するには、<code>sa_role</code> が付与されたユーザであることが必要。</p> <p>どのユーザも <code>sp_setsuspect_granularity</code> を実行して設定内容を表示できる。</p>

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [dump database](#)、[dump transaction](#)、[load database](#)
 システム・プロシージャ [sp_dboption](#)、[sp_forceonline_db](#)、[sp_forceonline_page](#)、[sp_listsuspect_db](#)、[sp_listsuspect_page](#)、[sp_setsuspect_threshold](#)

sp_setsuspect_threshold

説明	データベース全体に疑わしいことを示す <code>suspect</code> マークを付ける前に、データベース内で <code>Adaptive Server</code> が許容できる疑わしいページの最大値を表示または設定します。
構文	<code>sp_setsuspect_threshold [dbname [, threshold]]</code>
パラメータ	<p><i>dbname</i> 疑わしいエスカレーション・スレッシュホールドを表示または設定するデータベースの名前です。デフォルトは、現在のデータベースです。</p> <p><i>threshold</i> データベース全体に疑わしいことを示すマークを付ける前に、リカバリが許容する疑わしいデータ・ページ数の最大値を示します。デフォルトは 20 ページです。最小値は 0 です。</p>
例	<p>例 1 疑わしいページ数の最大値を 5 に設定します。疑わしいページが 5 ページを超えると、リカバリはデータベース全体にマークを付けます。</p> <pre>sp_setsuspect_threshold pubs2, 5</pre> <p>例 2 <code>pubs2</code> データベースに使用される <code>suspect</code> エスカレーション・スレッシュホールドの現在の設定と、新しく構成された設定を表示します。</p> <pre>sp_setsuspect_threshold pubs2</pre> <p>例 3 現在のユーザ・データベースに使用されるリカバリ・フォールト・アイソレーション・スレッシュホールドの現在の設定と、新しく構成された設定を表示します。</p> <pre>sp_setsuspect_threshold</pre>
使用法	<ul style="list-style-type: none">• <code>master</code> データベース内で、<code>sp_setsuspect_threshold</code> を使用して、<code>suspect</code> (疑わしい) エスカレーション・スレッシュホールドを設定してください。• ページ数を指定しなければ、<code>Adaptive Server</code> は現在の設定と新しく構成された設定を表示します。現在の設定とは、データベースで前回にリカバリが実行された際に有効だった設定のことです。新しく構成された設定とは、データベースで次回にリカバリが実行される時に有効になる設定のことです。

パーミッション `sp_setsuspect_threshold` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、エスカレーション・スレッショルドを設定するには、データベースに対する <code>own database</code> 権限を持つユーザーであることが必要。 どのユーザーも <code>sp_setsuspect_threshold</code> を実行して設定内容を表示できる。
細密なパーミッションが無効	細密なパーミッションが無効の場合、エスカレーション・スレッショルドを設定するには、 <code>sa_role</code> が付与されたユーザーであることが必要。 どのユーザーも <code>sp_setsuspect_threshold</code> を実行して設定内容を表示できる。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_forceonline_db](#)、[sp_forceonline_page](#)、[sp_listsuspect_db](#)、[sp_listsuspect_page](#)、[sp_setsuspect_granularity](#)

sp_setup_table_transfer

説明	このデータベース内に spt_TableTransfer テーブルを作成するために、増分転送のマークが付いたテーブルを含んでいる各データベースで 1 回ずつ実行します。
構文	<code>sp_setup_table_transfer</code>
使用法	<code>sp_setup_table_transfer</code> はオプションのプロシージャですが、テーブルを転送する前に実行することをおすすめします。 <code>sp_setup_table_transfer</code> を実行しない場合、Adaptive Server では、テーブルに増分転送のマークが付いているとき、または最初の転送を実行したとき、自動的に spt_TableTransfer を作成します。
パーミッション	<code>sp_setup_table_transfer</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 manage database 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か sa_role が付与されたユーザであることが必要。

sp_shmdumpconfig

説明	共有メモリ・ダンプのダンプ条件を指定したり、現在の設定を表示します。共有メモリ・ダンプを実行するには、 <code>dump on conditions</code> 設定パラメータを有効にしてください。
構文	<code>sp_shmdumpconfig "action", type, value, max_dumps, dump_dir, dump_file, option1, option2, option3, option4, option5</code>
パラメータ	<p>“action”</p> <p>要求するアクションです。次のいずれかを指定します。</p> <ul style="list-style-type: none">• <code>add</code> — 指定した共有メモリ・ダンプ条件を追加します。• <code>drop</code> — 指定した共有メモリ・ダンプ条件を削除します。• <code>update</code> — 既存の共有メモリ・ダンプ条件の設定を変更します。• <code>reset</code> — 共有メモリ・ダンプ条件のダンプ数をリセットします。• <code>display</code> — 現在の共有メモリ・ダンプ条件を表示します。• <code>config</code> — 次のいずれかを指定します。<ul style="list-style-type: none">• <code>include errorlog</code> — エラー・ログ・ファイルをダンプ・ファイルに含めるかどうかを指定します。<ul style="list-style-type: none">• <code>0</code> — エラー・ログをダンプ・ファイルに含めません。• <code>1</code> — エラー・ログをダンプ・ファイルに含めます。• <code>merge files</code> — 並列ダンプ後にダンプ・ファイルをマージするかどうかを指定します。<ul style="list-style-type: none">• <code>0</code> — ダンプ・ファイルをマージしません。• <code>1</code> — ダンプ・ファイルをマージします。

type value

- **error** — 指定したサーバ・エラー番号に対してダンプ・ファイルを生成します(例: エラー番号 1105、813 など)。
- **signal** — 指定したオペレーティング・システム・シグナルの発生時にダンプ・ファイルを生成します(例: シグナル 11、10 など)。
- **severity** — 指定した重大度以上のエラーが発生したときにダンプ・ファイルを生成します(エラーの重要度レベルについては、『システム管理ガイド 第1巻』の「第12章 システムの問題の診断」を参照してください)。
- **module** — 特定の範囲のサーバ・エラー番号に対してダンプ・ファイルを生成します。範囲は100の倍数単位で区切られます(例: 800、1200 など)。
- デフォルト
- **timeslice** — プロセスがタイムスライス・エラーを受け取ったときにダンプ・ファイルを生成します。
- **panic** — サーバ・パニック・エラーの発生時にダンプ・ファイルを生成します。サーバ・パニックが発生すると、Adaptive Server は共有メモリ・ダンプの実行後に終了します。
- **message** — 指定したエラー・ログ・メッセージの発生時にダンプ・ファイルを生成します。特定のエラー・メッセージ番号を取得するには、Sybase サポート・センタにお問い合わせください。
- **dbcc** —
- **codepoint** —
- **debug** —

max_dumps

1つのダンプ条件に対して生成される最大のダンプ数です。ダンプ数は、サーバを再起動するたびにリセットされます。**reset action** パラメータでもダンプ数をリセットできます。

dump_dir

Adaptive Server がダンプ・ファイルを作成するディレクトリです。“sybase” ユーザには、このディレクトリの読み取りと書き込みのパーミッションが必要です。

dump_dir は、既知の一貫した場所に設定することをおすすめします。必要な数のダンプ・ファイルを格納できる十分な領域がこのディレクトリにあることを確認してください。*dump_dir* 設定を削除するには、次のように *dump_dir* の値として 2 個の二重引用符 (") を指定して **update** アクションを実行します。

```
sp_shmdumpconfig 'update', signal, 11, null, null, ""
```

dump_file

ダンプのファイル名です。ファイル名を指定しないと、Adaptive Server により確実にユニークな名前が作成されます。ファイル名を指定した場合、該当する条件のすべてのファイルでこの名前が使用され、既存のファイルが上書きされます。

option1, ... ,option5

ダンプ・ファイルに Adaptive Server メモリの各領域を含めるかどうかを指定します (デフォルトでは、プロシージャ・キャッシュが含まれます)。次のいずれかを指定します。

- `include_page` —データ・キャッシュのすべてのページを含めます。
- `omit_page` —データ・キャッシュのすべてのページを除外します。
- `default_page` —データ・キャッシュ・ページを含めるときにデフォルト値を使用します。
- `include_proc` —プロシージャ・キャッシュのすべてのページを含めます。
- `omit_proc` —プロシージャ・キャッシュのすべてのページを除外します。
- `default_page` —プロシージャ・キャッシュのデフォルト値を使用します。
- `include_unused` —未使用のページを含めます。
- `omit_unused` —未使用のページを除外します。
- `default_unused` —未使用のページのデフォルト値を使用します。

これらのオプションの値は、システム全体のデフォルト設定よりも優先されます。システム全体のデフォルト設定から適切な値を継承するには、`default_cache`、`default_proc`、または `default_unused` を指定してください。

注意 Sybase テクニカル・サポートから別の指示がないかぎり、すべての共有メモリ・ダンプにプロシージャ・キャッシュを含めることをおすすめします。

halt

ダンプ・ファイルの書き込み中に Adaptive Server のエンジンを一時停止するかどうかを指定します。次のいずれかを指定します。

- **no_halt** — ダンプ中、エンジンを停止しません。共有メモリ・ダンプを使用しない場合は、このオプションを指定します(たとえば、ダウン時間が許容できない場合や、共有メモリ・ダンプをトリガするイベントが同期障害に起因しており、他のエンジンの動作を確認する必要がある場合など)。

no_halt オプションを指定して作成されたメモリ・ダンプには“ファジー”なイメージが含まれる場合があります、ダンプ・ファイルには破損したロック・テーブル、実行キューなどが含まれる可能性があります。

- **default_halt** —
- **halt** —

例

例 1 シグナル 11 についてワイトタイム・メモリ・ダンプを要求します。

```
sp_shmdumpconfig "add", signal, 11,1,"dump_dir"
```

例 2 エラー 605 の発生時にメモリ・ダンプを要求します。

```
sp_shmdumpconfig ' add' , error, 605, null, null, null,
include_page
```

Windows でこれに対応するのは STATUS_ACCESS_VIOLATION (0xc0000005) メッセージです。

```
declare @sig int
select @sig=hextoint("0xc0000005")
exec sp_shmdumpconfig 'add', signal, @sig,1,"dump_dir"
```

例 3 8xx 番台のエラーについてメモリ・ダンプを要求します。

```
sp_shmdumpconfig ' add' , module, 800
```

例 4 すでに定義されている **dump_file** を削除するには、**dump_file** の値として 2 個の二重引用符 ("") を指定して **update** アクションを実行します。

```
sp_shmdumpconfig 'update', signal, 11, null, null,
null, ""
```

使用法

- **sp_shmdumpconfig** ストアド・プロシージャは位置パラメータを使用しています。設定するパラメータが設定する必要がないパラメータより右に位置する場合は、使用しないパラメータに **null** 値を指定してください。

パーミッション

sp_shmdumpconfig のパーミッション・チェックは、細密なパーミッションの設定によって異なります。action が add、update、reset の場合、次のようになります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage server configuration 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

sp_show_options

説明 現在のセッションで設定されたすべてのサーバ・オプションを出力する。

構文 sp_show_options

パラメータ

例 次に、sp_show_options の出力例を示します。

```
create procedure sp_show_options
as
select a.number, a.name
      from master.dbo.spt_values a,
master.dbo.spt_values c
      where c.type = "P"
      and a.type='N'
      and c.low <= datalength(@@options)
      and a.number = c.number
      and convert(tinyint,substring(@@options,
c.low, 1)) & c.high != 0
      return (0)
go

1> sp_show_options
2> go

number      name
-----
          7 arithabort
          8 numeric_truncation
         13 control
         40 prefetch
         41 triggers
         42 replication
         43 replication_force_dll
         48 transactional_rpc
         58 remote_indexes
         62 statement_cache
         64 proc_return_status
         65 proc_output_params

(12 rows affected)
(return status = 0)
```

使用法

`@@options` はサーバ・オプションに対応するビットの配列を示します。すべてのオプションについて、“low” は `@@options` 内のバイト数で、“high” はオプションに対応するそのバイト内のビットです。ビットが設定されている場合、そのオプションの名前を出力する。

パーミッション

`sp_show_options` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

sp_showcontrolinfo

説明	スレッド・プールの割り当て、バインドされているクライアント・アプリケーション、ログイン、ストアド・プロシージャについての情報を表示します。
プロセス・モードの考慮事項	Adaptive Server をプロセス・モードで設定すると、sp_showcontrolinfo は、エンジン・グループの割り当て、バインドされているクライアント・アプリケーション、ログイン、ストアド・プロシージャに関する情報を表示します。
構文	sp_showcontrolinfo [object_type, object_name, spid]
パラメータ	<p><i>object_type</i></p> <p>下記のいずれか</p> <ul style="list-style-type: none"> • アプリケーションの場合、AP • ログインの場合、LG • ストアド・プロシージャの場合、PR • スレッド・プール(スレッド・モード)またはエンジン・グループ(プロセス・モード)の場合、EG • サービス・タスクの場合、SV • プロセスの場合、PS • ユーザ定義のデフォルトの実行クラスの場合、DF <p><i>object_type</i> を指定しないか、または <i>object_type</i> に null を指定すると、sp_showcontrolinfo はすべての型に関する情報を表示します。</p> <p><i>object_name</i></p> <p>アプリケーション、ログイン、ストアド・プロシージャ、エンジン・グループの名前です。<i>object_type</i> として PS または DF を指定しているときは、<i>object_name</i> は指定しないでください。<i>object_name</i> を指定しなければ(または <i>object_name</i> に null を指定すると)、sp_showcontrolinfo はすべてのオブジェクト名に関する情報を表示します。</p> <p><i>spid</i></p> <p>Adaptive Server プロセス ID です。<i>object_type</i> に PS を指定している場合のみ、spid を指定します。spid を指定しなければ(または spid に null を指定すると)、sp_showcontrolinfo は、すべての spid に関する情報を表示します。spid を確認するには、sp_who を使用します。</p>
例	例 1 ユーザが割り当てた実行クラスとオブジェクトのすべてのバインドを表示します。

```
sp_showcontrolinfo
```

例 2 isql アプリケーションの実行クラスを表示します。

```
sp_showcontrolinfo 'AP', 'isql'
```

例 3 スレッド・プールに割り当てられたすべてのプロセスの実行クラスを表示します。

```
sp_showcontrolinfo 'PS'
```

例 4 spid 7 の実行クラスを表示します。

```
sp_showcontrolinfo 'PS', null, 7
```

使用法

- `sp_showcontrolinfo` をパラメータなしで使用すると、ユーザが割り当てたすべてのスレッド・プールの割り当て、バインドされているクライアント・アプリケーション、ログイン、ストアド・プロシージャに関する情報が表示されます。`sp_showcontrolinfo` を、`object_type` パラメータと一緒に使用すると、アプリケーション、ログイン、実行クラスへのストアド・プロシージャのバインド、スレッド・プールの構成、セッションレベルの属性バインドに関する情報が、個別に表示されます。『パフォーマンス&チューニング・シリーズ：基本』の「第5章 タスク間でのエンジン・リソースの分配」を参照してください。
- プロセス・モードで実行すると、`sp_showcontrolinfo` は、`thread_pool` を `engine_group` カラムと `engine` カラムに置き換えます。
- `object_type` が PR である場合を除いて、`master` データベースから `sp_showcontrolinfo` を実行してください。`object_type` が PR である場合は、ストアド・プロシージャが格納されているデータベースから `sp_showcontrolinfo` を実行してください。
- `object_type` が次の場合の使用方法：
 - `null` — `sp_showcontrolinfo` は、他のパラメータに一致するオブジェクトについての実行クラス情報を表示します。
 - `DF` — `object_name` と `spid` は `null` である必要があります。`sp_showcontrolinfo` は、ユーザ定義のデフォルトの実行クラスに関する情報を示します。
- `object_name` が `null` のとき、`sp_showcontrolinfo` は、すべてのアプリケーション、ログイン、ストアド・プロシージャについてのバインド情報を表示します。
- `spid` が `null` のとき、`sp_showcontrolinfo` は、他のパラメータに一致するオブジェクトについての実行クラス情報を表示します。

パーミッション `sp_showcontrolinfo` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ `sp_addexclass`、`sp_bindexclass`、`sp_clearpsex`、`sp_dropengine`、`sp_dropexclass`、`sp_showexclass`、`sp_showpsex`、`sp_unbindexclass`、`sp_who`
ユーティリティ `isql`

sp_showexeclass

説明 特定の実行クラスに対応する実行クラス属性とスレッド・プール名を表示します。

プロセス・モードの考慮事項 プロセス・モードで、sp_showexeclass は、指定された実行クラスに対応する任意のエンジン・グループ内の実行クラス属性とエンジンを表示します。

構文 sp_showexeclass [execlassname]

パラメータ execlassname
実行クラスの名前です。

例 **例 1** すべての実行クラスの優先度とスレッド・プールを表示します。

```
sp_showexeclass
classname          priority          threadpool
-----
EC1                  HIGH          syb_default_pool
EC2                  MEDIUM       syb_default_pool
EC3                  LOW           syb_default_pool
```

例 2 実行クラス EC1 の属性値を表示します。

```
sp_showexeclass 'EC1'
classname          priority          threadpool
-----
EC1                  HIGH          syb_default_pool
```

使用法 execlassname が NULL の場合、または指定されなかった場合、sp_showexeclass は、システムが定義するクラス EC1、EC2、および EC3 の属性値も含め、すべての実行クラスの優先度とスレッド・プール属性の値を表示します。

パーミッション sp_showexeclass は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addexclass](#)、[sp_bindexclass](#)、[sp_dropexclass](#)、[sp_showcontrolinfo](#)、[sp_unbindexclass](#)

sp_showoptstats

説明	<p>optdiag スタンドアロン・ユーティリティの機能と同様に、XML ドキュメントで (ただし、システム・プロシージャ形式で) <code>sp_showoptstats</code> は、<code>systabstats</code>、<code>sysstatistics</code> などのシステム・テーブルからさまざまなデータ・オブジェクトの統計とヒストグラムを抽出して表示します。</p>
構文	<pre>sp_showoptstats [[database_name.[owner].]table_name], [column_name], [h]</pre>
パラメータ	<p>database_name</p> <p><code>sp_showoptstats</code> が統計とヒストグラムを表示するデータベースの名前です。 <i>database_name</i> には次の制約があります。</p> <ul style="list-style-type: none">データベース間の実行はサポートされません。<code>sp_showoptstats</code> を実行するには、現在、指定されたデータベースにいる必要があります。データベースを指定しない場合、<code>sp_showoptstats</code> は現在のデータベースに関する統計とヒストグラムを表示します。 <p>owner</p> <p>テーブル所有者の名前を指定します。所有者名を指定しなければ、現在のユーザ名または <code>dbo</code> が使用されます。</p> <p>table_name</p> <p><code>sp_showoptstats</code> が統計とヒストグラムを表示するテーブルの名前です。 <i>table_name</i> には次の制約があります。</p> <ul style="list-style-type: none">テーブルを指定しない場合、<code>sp_showoptstats</code> は現在のデータベースにあるすべてのテーブルに関する統計とヒストグラムを表示します。ただし、出力の長さを短くするために、データベース・レベルのカラムの統計とヒストグラムを表示しません。<i>table_name</i> が、現在のデータベース内に存在している必要があります。 <p>column_name</p> <p>Adaptive Server が統計とヒストグラムを表示するカラムの名前。カラムを指定しないと、Adaptive Server は指定したテーブルのすべてのカラムおよびインデックスの統計を表示します。 <i>column_name</i> を指定すると、<code>sp_showoptstats</code> はそのカラムの統計とヒストグラムのみを表示します。</p> <p>h</p> <p>プロシージャのヘルプ情報を表示します。</p>

例

例 1 pubs2 データベース内にあるすべてのユーザ・テーブルの統計を表示します。

```
1> use pubs2
2> go
1> sp_showoptstats 'pubs2..publishers'
2> go
```

例 2 pubs2 データベースの publishers テーブルの統計とヒストグラムを XML 形式で表示します。

```
1> sp_showoptstats publishers
<?xml version="1.0" encoding="UTF-8"?>
<optStats>
  <procVersion>sp_showoptstats/1.1/AnyPlatform/AnyOS/Tues April 3 14:21:21
2012</procVersion>
  <serverVersion>Adaptive Server Enterprise/15.7.0/EBF 20161 SMP ESD#02
Prelim#2/P/x86_64/Enterprise Linux/ase157x/3087/64-bit/FBO/Tue May 15
05:35:01 2012</serverVersion>
  <serverName></serverName>
  <specifiedDatabase>pubs2</specifiedDatabase>
  <specifiedTableOwner></specifiedTableOwner>
  <specifiedTable>publishers</specifiedTable>
  <specifiedCol></specifiedCol>
  <tables>
    <tableOwner>dbo</tableOwner>
    <tableName>publishers</tableName>
    <clusteredIndStats>
      <indName>pubind</indName>
      <colList>"pub_id"</colList>
      <stats>
        <pgCnt>1</pgCnt>
        <emptyPgCnt>0</emptyPgCnt>
        <rowCnt>3.0000000000000000</rowCnt>
        <fwdRowCnt>0.0000000000000000</fwdRowCnt>
        <delRowCnt>0.0000000000000000</delRowCnt>
        <CRCnt>1.0000000000000000</CRCnt>
        <oamAllocPgCnt>2</oamAllocPgCnt>
        <firstExtLeafPgs>0</firstExtLeafPgs>
        <dataRowSz>39.3333333333333357</dataRowSz>
        <indHeight>1</indHeight>
        <joinDegree>0.0000000000000000</joinDegree>
        <unusedPgCnt>14</unusedPgCnt>
        <oamPgCnt>1</oamPgCnt>
        <derivedStats>
          <clusterRatio>0.0000000000000000</clusterRatio>
          <spaceUtil>0.0072162426614481</spaceUtil>
```

```
        <IOEfficiency>0.5000000000000000</IOEfficiency>
    </derivedStats>
</stats>
</clusteredIndStats>
<colStats>
  <colName>pub_id</colName>
  <lastUpdate>May 15 2012 4:44:40:136PM</lastUpdate>
  <cellDensity>0.3333333333333333</cellDensity>
  <totalDensity>0.3333333333333333</totalDensity>
  <select>default used (0.33)</select>
  <inBetSel>default used (0.25)</inBetSel>
  <rangeVal>0.3333333333333333</rangeVal>
  <totalVal>0.3333333333333333</totalVal>
  <avgColWidth>default used (4.00)</avgColWidth>
  <statsVer>4</statsVer>
  <histogram>
    <colName>pub_id</colName>
    <dataType>char(4)</dataType>
    <requestedStepCnt>20</requestedStepCnt>
    <actualStepCnt>6</actualStepCnt>
    <samplingPct>0</samplingPct>
    <TuningFact>20</TuningFact>
    <steps>
      <step>1</step>
      <weight>0.00000000</weight>
      <equation>&lt;</equation>
      <value>"0736"</value>
    </steps>
    <steps>
      <step>2</step>
      <weight>0.33333334</weight>
      <equation>=</equation>
      <value>"0736"</value>
    </steps>
    <steps>
      <step>3</step>
      <weight>0.00000000</weight>
      <equation>&lt;</equation>
      <value>"0877"</value>
    </steps>
    <steps>
      <step>4</step>
      <weight>0.33333334</weight>
      <equation>=</equation>
      <value>"0877"</value>
    </steps>
  </histogram>
</colStats>
```

```

<steps>
  <step>5</step>
  <weight>0.00000000</weight>
  <equation>&lt;&/equation>

  <value>"1389"</value>
</steps>
<steps>
  <step>6</step>
  <weight>0.33333334</weight>
  <equation>=</equation>
  <value>"1389"</value>
</steps>
</histogram>
</colStats>
<noStatsCol>city, pub_name, state
</noStatsCol>
</tables>
</optStats>

```

例 3 プロシージャの構文を示します。

```

1> sp_showoptstats a,b,h
2> go
Usage:sp_showoptstats [[database.[owner].]table], [column], [option]
(return status = 0)

```

使用法

- sp_showoptstats は、複数のデータベース間で実行できます。
- sp_showoptstats は、システム・テーブルを明示的に指定しない場合はシステム・テーブルを含めません。
- 表示できない文字や univarchar 文字を 16 進数形式で表示します。
- sp_showoptstats は、グローバルな統計とパーティション・レベルの統計の両方を表示します。
- 出力が @@textsize で設定した値よりも大きい場合、Adaptive Server はサイズの大きい出力を表示できるように @@textsize の設定値を増やすようにメッセージを返します。
- ピリオド(.)を含むパラメータの値には、二重引用符が必要です。
- システム・テーブルに対して sp_showoptstats を発行できます。
- データベースと所有者のみを指定した場合、sp_showoptstats は統計情報を返しません。

DTD XML 出力

sp_showoptstats の XML 出力用の DTD ファイルは、次のとおりです。

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT optStats (procVersion, serverVersion, serverName?,
specifiedDatabase?, specifiedTableOwner?, specifiedTable?, specifiedCol?,
tables*)>
<!ELEMENT procVersion (#PCDATA)>
<!ELEMENT serverVersion (#PCDATA)>
<!ELEMENT serverName (#PCDATA)>
<!ELEMENT specifiedDatabase (#PCDATA)>
<!ELEMENT specifiedTableOwner (#PCDATA)>
<!ELEMENT specifiedTable (#PCDATA)>
<!ELEMENT specifiedCol (#PCDATA)>

<!ELEMENT tables (tableOwner, tableName, partitionCnt?,
(tableStats|clusteredIndStats|indStats|partitionStats|partitionClusteredIndSt
ats|partitionIndStats)*, (colStats|colPartitionStats)*, noStatsCol?)>

<!ELEMENT tableOwner (#PCDATA)>
<!ELEMENT tableName (#PCDATA)>

<!ELEMENT tableStats (tableName, stats)>

<!ELEMENT clusteredIndStats (indName, colList, stats)>

<!ELEMENT indName (#PCDATA)>
<!ELEMENT colList (#PCDATA)>

<!ELEMENT partitionStats (partition*, stats*)>

<!ELEMENT partition (#PCDATA)>

<!ELEMENT partitionIndStats (indName, partition, colList, stats)>

<!ELEMENT partitionClusteredIndStats (indName, partition, colList, stats)>

<!ELEMENT stats (pgCnt?, leafCnt?, (emptyPgCnt|emptyLeafCnt)?, CRCnt?,
indCRCnt?, indPgCRCnt?, (dataRowCRCnt|leafRowCRCnt)?, rowCnt?, fwdRowCnt?,
delRowCnt?, indPgCRCnt?, CRCnt?, oamAllocPgCnt?,
(firstExtDataPgs|firstExtLeafPgs)?, (dataRowSz|leafRowSz)?, indHeight?,
dataPages?, joinDegree?, unusedPgCnt?, oamPgCnt?, derivedStats?)>

<!ELEMENT pgCnt (#PCDATA)>
<!ELEMENT leafCnt (#PCDATA)>
<!ELEMENT CRCnt (#PCDATA)>
```

```
<!ELEMENT indCRCnt (#PCDATA)>
<!ELEMENT dataRowCRCnt (#PCDATA)>
<!ELEMENT leafRowCRCnt (#PCDATA)>
<!ELEMENT emptyPgCnt (#PCDATA)>
<!ELEMENT emptyLeafCnt (#PCDATA)>
<!ELEMENT rowCnt (#PCDATA)>
<!ELEMENT fwdRowCnt (#PCDATA)>
<!ELEMENT delRowCnt (#PCDATA)>

<!ELEMENT oamAllocPgCnt (#PCDATA)>
<!ELEMENT firstExtDataPgs (#PCDATA)>
<!ELEMENT firstExtLeafPgs (#PCDATA)>
<!ELEMENT dataRowSz (#PCDATA)>
<!ELEMENT leafRowSz (#PCDATA)>
<!ELEMENT indHeight (#PCDATA)>
<!ELEMENT dataPages (#PCDATA)>
<!ELEMENT joinDegree (#PCDATA)>
<!ELEMENT unusedPgCnt (#PCDATA)>
<!ELEMENT oamPgCnt (#PCDATA)>
<!ELEMENT rowClusterRatio (#PCDATA)>

<!ELEMENT derivedStats (clusterRatio, indClusterRatio?,
(dataClusterRatio|rowClusterRatio)?, spaceUtil?, IOEfficiency?)>

<!ELEMENT clusterRatio (#PCDATA)>
<!ELEMENT indClusterRatio (#PCDATA)>
<!ELEMENT dataClusterRatio (#PCDATA)>
<!ELEMENT spaceUtil (#PCDATA)>
<!ELEMENT IOEfficiency (#PCDATA)>

<!ELEMENT indStats (indName, colList, stats?)>

<!ELEMENT colStats ((colName|colGroup)?, lastUpdate?, cellDensity?,
totalDensity?, select?, inBetSel?, rangeVal?, totalVal?, avgColWidth?,
statsVer? statsSamDen?, statsSamU?, histogram?)>

<!ELEMENT colGroup (#PCDATA)>
<!ELEMENT lastUpdate (#PCDATA)>
<!ELEMENT cellDensity (#PCDATA)>
<!ELEMENT totalDensity (#PCDATA)>
<!ELEMENT selectivity (#PCDATA)>
<!ELEMENT inBetweenSelectivity (#PCDATA)>
<!ELEMENT rangeVal (#PCDATA)>
<!ELEMENT totalVal (#PCDATA)>
<!ELEMENT avgColWidth (#PCDATA)>
<!ELEMENT statsVer (#PCDATA)>
```

```
<!ELEMENT statsSamDen (#PCDATA)>
<!ELEMENT statsSamU (#PCDATA)>

<!ELEMENT colPartitionStats (ptnName,(colName|colGroup)?, lastUpdate?,
cellDensity?, totalDensity?, select?, inBetSel?, rangeVal?, totalVal?,
avgColWidth?, statsVer? statsSamDen?, statsSamU?, histogram?)>

<!ELEMENT ptnName (#PCDATA)>

<!ELEMENT histogram (colName, dataType, requestedStepCnt, actualStepCnt,
samplingPct?, TuningFact?, statsOutRan?, statsHashLow?, statsHashHigh?,
statsSamSt?, statsStepSt?, statsHtSt?, statsPHashSt?, statsHashSt?,
statsNoHashSt?, steps*)>

<!ELEMENT colName (#PCDATA)>
<!ELEMENT dataType (#PCDATA)>
<!ELEMENT requestedStepCnt (#PCDATA)>
<!ELEMENT actualStepCnt (#PCDATA)>
<!ELEMENT samplingPct (#PCDATA)>
<!ELEMENT TuningFact (#PCDATA)>
<!ELEMENT statsOutRan (#PCDATA)>
<!ELEMENT statsHashLow (#PCDATA)>
<!ELEMENT statsHashHigh (#PCDATA)>
<!ELEMENT statsSamSt (#PCDATA)>
<!ELEMENT statsStepSt (#PCDATA)>
<!ELEMENT statsHtSt (#PCDATA)>
<!ELEMENT statsPHashSt (#PCDATA)>
<!ELEMENT statsHashSt (#PCDATA)>
<!ELEMENT statsNoHashSt (#PCDATA)>

<!ELEMENT steps (step, weight, equation, value)>

<!ELEMENT step (#PCDATA)>
<!ELEMENT weight (#PCDATA)>
<!ELEMENT equation (#PCDATA)>
<!ELEMENT value (#PCDATA)>

<!ELEMENT noStatsCol (#PCDATA)>
```

パーミッション

sp_showoptstats は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 ユーティリティ optdiag

マニュアル『パフォーマンス&チューニング・シリーズ：統計的分析によるパフォーマンスの向上』の「第 2 章 統計テーブルおよび optdiag を使った統計の表示」、『ユーティリティ・ガイド』の optdiag リファレンス・ページ

sp_showplan

説明	現在の SQL 文または同じバッチの前の文について、すべてのユーザ接続の showplan 出力を表示します。
構文	<pre>sp_showplan <i>spid</i>, <i>batch_id</i> output, <i>context_id</i> output, <i>stmt_num</i> output</pre> <p><i>batch_id</i>、<i>context_id</i>、<i>stmt_num</i> を指定しないで現在の SQL 文の showplan 出力を表示するには、次のように指定します。</p> <pre>sp_showplan <i>spid</i>, null, null, null</pre>
パラメータ	<p><i>spid</i> 任意のユーザ接続のプロセス ID です。spid を表示するには sp_who を使用します。</p> <p><i>batch_id</i> 負でないユニークなバッチ番号です。</p> <p><i>context_id</i> バッチで実行されるすべてのプロシージャ (またはトリガ) のユニークな番号です。</p> <p><i>stmt_num</i> バッチ内の現在の文の番号です。stmt_num は、正の番号にしてください。</p>
例	<p>例 1 <i>spid</i> の値が 99 のユーザ・セッションで実行されている現在の文のクエリ・プランと <i>batch_id</i>、<i>context_id</i>、<i>statement_id</i> パラメータの値を表示します。これらの値は、<i>spid</i> の値が 99 のユーザ・セッションに対して、sp_showplan を反復して実行する場合に、クエリ・プランの検索に使用できます。</p> <pre>declare @batch int declare @context int declare @statement int exec sp_showplan 99, @batch output, @context output, @statement output</pre> <p>例 2 <i>spid</i> の値が 99 のユーザ・セッションで動作している現在の文の showplan 出力を表示します。</p> <pre>sp_showplan 99, null, null, null</pre>

- 使用法**
- `sp_showplan` は、現在実行している SQL 文か、または同じバッチの前の文について、`showplan` 出力を表示します。
 - 同じバッチの前の文のクエリ・プランを表示するには、パラメータの値は同じにして、文の番号を 1 だけ減らして `sp_showplan` を実行します。この方法を使用すると、クエリ番号 1 番までさかのぼってバッチ内のすべての文を表示することができます。
 - `sp_showplan` は、Adaptive Server の Monitor™ Server とは独立して実行することができます。
 - `sp_showplan` は、`sp_procxmode` を使用してトランザクション・モードを `anymode` に変更した後に、連鎖トランザクションを使用するセッションで実行できます。
 - SQL バッチに対して `context_id` が 0 よりも大きい場合は、最初の SQL バッチから呼び出されるストア・プロシージャ (またはトリガ) に現在の文が埋め込まれます。プロシージャ ID と文 ID を表示するには、同じ `spid` 値を持つ `sysprocesses` ローを選択します。

パーミッション `sp_showplan` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>monitor qp performance</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ `sp_who`

sp_showpsexex

説明 Adaptive Server で実行されているすべてのクライアント・セッションの実行クラス、現在の優先度、スレッド・プールの結び付きを表示します。

プロセス・モードの考慮事項 sp_showpsexex は、タスクとの結び付きの代わりにエンジン情報を表示します。

構文 sp_showpsexex [*spid*]

パラメータ *spid*
 情報を表示する Adaptive Server のセッション ID です。 *spid* は、sp_showpsexex を実行しているアプリケーションまたはログインに属している必要があります。 *spid* をリストするには、sp_who を実行します。

例 **例 1** 現在のすべてのクライアント・セッションの実行クラス、現在の優先度、結び付きを表示します。

sp_showpsexex

spid	appl_name	login_name	exec_class	current_priority	task_affinity
5	NULL	NULL	NULL	LOW	syb_default_pool
6	NULL	NULL	NULL	MEDIUM	syb_default_pool
7	NULL	NULL	NULL	LOW	syb_default_pool
26	isql	sa	EC2	MEDIUM	syb_default_pool

例 2 spid 5 のプロセスのアプリケーション名、ログイン名、現在の優先度、エンジンの結び付きを表示します。

sp_showpsexex 5

spid	appl_name	login_name	exec_class	current_priority	task_affinity
5	NULL	NULL	NULL	LOW	syb_default_pool

使用法

- sp_showpsexex は、すべてのセッション (*spid* のオブジェクト) について、実行クラス、現在の優先度、結び付きを表示します。『パフォーマンス&チューニング・シリーズ：基本』の「第4章 タスク間でのエンジン・リソースの分配」を参照してください。
- *spid* が NULL の場合、または指定されなかった場合、sp_showpsexex は、現在 Adaptive Server で実行されているすべてのセッションについてレポートします。

- `sp_showpsex` は、デッドロック、チェックポイント、ネットワーク、監査、ミラー・ハンドラの各システム・プロセスに関する情報についてはレポートしません。ハウスキーピング *spid*
- についてはその情報を表示します。

パーミッション

`sp_showpsex` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_addengine](#)、[sp_addexclass](#)、[sp_bindexclass](#)、[sp_clearpsex](#)、[sp_dropengine](#)、[sp_dropexclass](#)、[sp_showcontrolinfo](#)、[sp_showexclass](#)、[sp_unbindexclass](#)

sp_spaceusage

説明 テーブル、インデックス、トランザクション・ログの領域使用率に関するレポートを作成したり、データベースのテーブルおよびインデックスの断片化量の推定値を計算したりします。推定値は、データ・ローとインデックス・ローの平均の長さとしてテーブル内のローの数を使用して計算されます。領域使用率および断片化データをアーカイブしておき、将来、レポートや傾向分析に使用できます。sp_spaceusage プロシージャでは、help、display、archive、report など、Adaptive Server の現在の領域の使用率のレポートを作成するための多数のアクションがサポートされています。

構文 “help” アクションの構文は次のとおりです。

```
sp_spaceusage 'help'[, 'all']
sp_spaceusage 'help' [, {'display' | 'display summary'
                          | 'report' | 'report summary' | 'archive'}
                        [, {'table' | 'index' | 'tranlog'}]]
```

“display” アクションの構文は次のとおりです。

```
sp_spaceusage 'display summary [using unit={KB | MB | GB | PAGES}
]',
  {'table' | 'index'}, name
  [,where_clause [,order_by[,command] ] ]
sp_spaceusage 'display [using unit={KB | MB | GB | PAGES}]',
  {'table' | 'index'}, name
  [,select_list
  [,where_clause [,order_by[,command] ] ] ]
sp_spaceusage 'display [using unit={KB | MB | GB | PAGES}]',
  'tranlog' [, name[,select_list[,where_clause [,order_by]]]]
```

“archive” アクションの構文は次のとおりです。

```
sp_spaceusage 'archive [ using_clause ]',
  {'table' | 'index'}, name[,where_clause[,command] ]
sp_spaceusage 'archive [ using_clause ]',
  'tranlog' [,name[,where_clause] ]
```

“report” アクションの構文は次のとおりです。

```
sp_spaceusage 'report summary [ using_clause ]',
  {'table' | 'index'}, name
  [,where_clause [,order_by[,from_date [,to_date]]]]
sp_spaceusage 'report [ using_clause ]',
  {'table' | 'index'}, name
  [,select_list[,where_clause [,order_by[,from_date [,to_date]]]]]
sp_spaceusage 'report [ using_clause ]',
  'tranlog' [, name
```

```
[,select_list[,where_clause [,order_by
[,from_date [,to_date]]]]]]
using_clause = USING using_item [, using_item ...]
using_item = { unit={ KB | MB | GB | PAGES }
| dbname=database_name | prefix=string }
```

パラメータ

help

`sp_spaceusage` 構文全体を表示します。help アクションは、サポートされている個々のアクションの構文を表示します。

display

指定されたオブジェクトの現在の領域の使用量情報を表示します。

display summary

指定されたオブジェクトの現在の領域の使用量情報の概要を表示します。

archive

領域の使用量レポートをテーブルにアーカイブします。まだアーカイブ・テーブルが存在しない場合は、`sp_spaceusage` によって作成されます。新しいデータは既存のデータに追加されます。`using` 句を使用すると、アーカイブ・テーブル名とアーカイブ・テーブルが存在するデータベースのプレフィクスを指定できます。

report

以前にアーカイブされたデータから、指定されたオブジェクトの領域の使用量情報レポートを生成します。`display` アクションと同じ内容が出力されます。アーカイブ・テーブルを指定するには、`using` 句 (任意指定可) を含めます。

report summary

以前にアーカイブされたデータから、指定されたオブジェクトの領域の使用量情報の概要レポートを生成します。`display summary` アクションと同じ内容が出力されます。アーカイブ・テーブルを指定するには、`using` 句 (任意指定可) を含めます。

using_item

アーカイブ・テーブルの単位、アーカイブ・データベース名、プレフィクス文字列を指定します。`unit` サイズとして、キロバイト (KB)、メガバイト (MB)、ギガバイト (GB)、ページ数を使用できます。`unit` サイズは KB、現在のデータベースはアーカイブ・データベース、プレフィクス文字列なしがデフォルトです。

name

エンティティの名前です。エンティティ・タイプによっては、`owner_name.table_name`、`owner_name.table_name.index_name`などの複数部分からなる名前を含めることができます。エンティティ・タイプが `tranlog` の場合、名前は `syslogs` か `NULL` にしてください。複数のオブジェクトについてワンパスでレポート作成するには、複数部分で構成される名前の各部分にパターン指定子を使用できます。

select_list

`display` および `report` アクションの出力カラムに含めるために選択するカラムをカンマで区切ったリストです。出力にすべてのカラムを含めるには、`*` を使用します。カラムの名前は、`alias=name` という形式に従って変更できます。

where_clause

結果セットに適用するフィルタです。不要なデータを選択的にフィルタするには、これを `display`、`report`、または `archive` アクションで使用します。

order_by

クエリ結果を、指定されたカラムのソート順で返します。

command

オブジェクトの使用量に関する情報を収集する前に、選択されたエンティティ(テーブル、カラムなど)に対して実行されます。サポートされているコマンドは、`update statistics`、`update table statistics`、`update index statistics` です。

from_date

時間範囲の始まりを指定します。

to_date

時間範囲の終わりを指定します。

例 **例 1** `display` アクションの簡単な説明、構文、使用量情報を表示します。

```
sp_spaceusage 'help', 'display'
```

```
Display the space usage information for an entity in the current database.
```

```
Usage:
```

```
sp_spaceusage 'display', {'table'|'index'}, <name>
```

```
    [,<select_list> [,<where_clause> [,<order_by> [,<command>]]]]
```

```
sp_spaceusage 'display summary', {'table'|'index'}, <name>
```

```
    [,<where_clause> [,<order_by> [,<command>]]]
```

```
sp_spaceusage 'display', 'tranlog' [,{'syslogs'|NULL}]
```



```
[,<select_list> [,<where_clause> [,<order_by>]]]
```

For more information, use:

```
sp_spaceusage 'help', 'display', 'table'
sp_spaceusage 'help', 'display', 'index'
sp_spaceusage 'help', 'display', 'tranlog'
```

例 2 titles テーブルの領域の使用量の概要を表示します。

```
sp_spaceusage 'display summary', 'table', 'titles'
```

All the page counts in the result set are in the unit 'KB'.

OwnerName	TableName	Type	UsedPages	RsvdPages	ExpRsvdPages	PctBloatRsvdPages
dbo	titles	DATA	6.0	30.0	16.0	87.50
dbo	titles	INDEX	8.0	64.0	32.0	50.00

例 3 titles テーブルの領域の使用量情報を表示します。

```
sp_spaceusage 'display', 'table', 'titles'
```

All the page counts in the result set are in the unit 'KB'.

OwnerName	TableName	IndId	NumRows	UsedPages	RsvdPages
ExtentUtil	ExpRsvdPages	PctBloatUsePages	PctBloatRsvdPages		
dbo	titles	0	18.0	6.0	30.0
20.00	16.0	0.0	87.50		
dbo	titles	1	NULL	4.0	32.0
12.50	16.0	0.00	100.00		
dbo	titles	2	NULL	4.0	32.0
12.50	16.0	0.00	100.00		

例 4 title テーブルの、名前が **titles** で始まるすべてのインデックスの領域の使用量に関する情報をメガバイト単位で表示します。

```
sp_spaceusage 'display using unit-MB', 'index', 'titles.title%'
```

All the page counts in the result set are in the unit 'MB'.

OwnerName	TableName	IndId	IndexName	UsedPages	RsvdPages
ExtentUtil	ExpRsvdPages	PctBloatUsedPages	PctBloatRsvdPages		
dbo	titles	0	titles	.005859375	.029296875
20.00	.015625	0.00	87.50		
dbo	titles	1	titleidind	.00390625	.03125
12.50	.015625	0.00	100.00		
dbo	titles	2	titleind	.00390625	.03125
12.50	.015625	0.00	100.00		

(1 row affected)


```

50.00
dbo      titles      titles      0      6.0      30.0      16.0
46.67

```

例 8 authors テーブルに対して `update table statistics` を実行し、その領域の使用量に関する情報の概要を単位 *pages* で表示します。

```

sp_spaceusage 'display summary using unit=pages', 'table', 'authors', null,
null, null, 'update table statistics'

All the page counts in the result set are in the unit 'pages'.
OwnerName TableName Type UsedPages RsvdPages ExpRsvdPages PctBloatRsvdPages
-----
dbo          authors  DATA   2.0          16.0          8.0          100.00
dbo          authors  INDEX   4.0          32.0          16.0          50.00

```

例 9 現在のデータベース (*pubs2*) のトランザクション・ログの領域の使用量に関する情報を表示します。

```

sp_spaceusage 'display', 'tranlog'

TableName TotalPages UsedPages CLRPagesFreePages PctUsedPages PctFreePages
-----
syslogs   4096.0      18.0 0.0  1482.0          0.43      36.18

```

例 10 現在のデータベース内の *authors* テーブルの領域の使用量に関する情報を、デフォルト・テーブル (テーブルとインデックスの *spaceusage_object*) にアーカイブします。

```

sp_spaceusage 'archive', 'table', 'authors'

Data was successfully archived into table 'pubs2.dbo.spaceusage_object'.

```

例 11 *authors* テーブルの領域の使用量に関する情報を、*pubs3* データベース内のデフォルト・テーブル (テーブルとインデックスの *spaceusage_object*) にアーカイブします。

```

sp_spaceusage 'archive using dbname = pubs3', 'table', 'authors'

Data was successfully archived into table 'pubs3.dbo.spaceusage_object'.

```

例 12 *authors* テーブルに対して `update table statistics` を実行し、プレフィックス *monday_* (*monday_spaceusage_object* など) を付けて、現在のデータベース内のテーブルに *authors* テーブルの領域の使用量情報をアーカイブします。

```

1> sp_spaceusage 'archive using dbname = pubs2, prefix=monday_',
'table','authors', null, 'update table statistics'

```

例 13 現在のデータベースのトランザクション・ログの領域の使用量に関する情報を、pubs3 データベース内のデフォルト・テーブル(トランザクション・ログの spaceusage_tranlog) にアーカイブします。

```
sp_spaceusage 'archive using dbname=pubs3', 'tranlog'
Data was successfully archived into table 'pubs3.dbo.spaceusage_tranlog'.
```

例 14 現在のデータベース内のデフォルト・テーブル(テーブルまたはインデックスの spaceusage_object) に最後にアーカイブされた、authors テーブルの領域の使用量に関する詳しい情報のレポートを生成します。

```
sp_spaceusage 'report', 'table', 'authors'
All the page counts in the result set are in the unit 'KB'.
All the data in the result set are dated 'Jun 15 2007 11:50PM'.
OwnerName  TableName  IndId  NumRows  UsedPages  RsvdPages  ExtentUtil
  ExpRsvdPages  PctBloatUsedPages  PctBloatRsvdPages
-----
-----
-----
dbo         authors    0       23.0    4.0        32.0       12.50
  16.0      0.00
dbo         authors    1       NULL    4.0        32.0       12.50
  16.0      0.00
dbo         authors    2       NULL    4.0        32.0       12.50
  16.0      0.00
(1 row affected)
(return status = 0)
```

例 15 pubs3 データベース内のデフォルト・テーブルに最後にアーカイブされた、authors テーブルの領域の使用量情報の概要レポートを生成します。

```
sp_spaceusage 'report summary using dbname=pubs3', 'table', 'authors'
All the page counts in the result set are in the unit 'KB'.
All the data in the result set are dated 'Jan 17 2007 11:29AM'.
OwnerName  TableName  Type  UsedPages  RsvdPages  ExpRsvdPages  PctBloatRsvdPages
-----
-----
-----
dbo         authors    DATA  4.0        32.0       16.0          100.00
dbo         authors    INDEX  8.0        64.0       32.0          50.00
```

例 16 現在のデータベース内の monday_spaceusage_object テーブルに最後にアーカイブされた、authors テーブルの領域の使用量情報の概要レポートをメガバイト単位で生成します。

```
sp_spaceusage 'report summary using prefix=monday_, unit=MB', 'table',
'authors'
```

```
All the page counts in the result set are in the unit 'MB'.
All the data in the result set are dated 'Jan 17 2007 11:29AM'.
OwnerName TableName Type UsedPages RsvdPages ExpRsvdPages PctBloatRsvdPages
-----
dbo         authors  DATA  .00390625  .03125  .015625  100.00
dbo         authors  INDEX  .0078125   .0625   .03125   50.00
```

例 17 2007 年 6 月 9 日以降に現在のデータベース内のデフォルト・テーブルにアーカイブされた、**authors** テーブルのすべてのインデックスの領域の使用量情報のレポートを生成します。

```
sp_spaceusage 'report', 'index', 'authors.%', null, null, null, 'Jun 9 2007'
```

```
All the page counts in the result set are in the unit 'KB'.
ArchiveDateTime      OwnerName TableName IndId IndexName UsedPages
RsvdPages ExtentUtil ExpRsvdPages PctBloatUsedPages PctBloatRsvdPag
es
-----
Jun  9 2007 12:06AM  dbo         authors      0  authors      4.0
 32.0      12.50      16.0         0.00 100.00
Jun 10 2007 12:05AM  dbo         authors      0  authors      4.0
 32.0      12.50      16.0         0.00 100.00
Jun 11 2007 11:35PM  dbo         authors      0  authors      4.0
 32.0      12.50      16.0         0.00 100.00
Jun  9 2007 12:06AM  dbo         authors      1  auidind      4.0
 32.0      12.50      16.0         0.00 100.00
Jun 10 2007 12:05AM  dbo         authors      1  auidind      4.0
 32.0      12.50      16.0         0.00 100.00
Jun 11 2007 11:35PM  dbo         authors      1  auidind      4.0
 32.0      12.50      16.0         0.00 100.
Jun  9 2007 12:06AM  dbo         authors      2  aunmind      4.0
 32.0      12.50      16.0         0.00 100.00
Jun 10 2007 12:05AM  dbo         authors      2  aunmind      4.0
 32.0      12.50      16.0         0.00 100.00
Jun 11 2007 11:35PM  dbo         authors      2  aunmind      4.0
 32.0      12.50      16.0         0.00 100.00
(1 row affected)
(return status = 0)
```

例 18 2007 年 6 月 10 日から 2007 年 6 月 15 日の間に現在のデータベース内のデフォルト・テーブルにアーカイブされた、**authors** テーブルの領域使用量情報のレポートを生成します。

```
sp_spaceusage 'report', 'table', 'authors', null, null, null, 'Jun 10 2007',
'Jun 15 2007'
```

```
All the page counts in the result set are in the unit 'KB'.
```

sp_spaceusage

ArchiveDateTime	OwnerName	TableName	IndId	NumRows	UsedPages
RsvdPages	ExtentUtil	ExpRsvdPages	PctBloatUsedPages	PctBloatRsvdPages	
Jun 10 2007 12:05AM	dbo	authors	0	23.0	4.0
32.0	12.50	16.0	0.00		100.00
Jun 11 2007 11:35PM	dbo	authors	0	23.0	4.0
32.0	12.50	16.0	0.00		100.00
Jun 13 2007 11:46PM	dbo	authors	0	23.0	4.0
32.0	12.50	16.0	0.00		100.00
Jun 14 2007 11:46PM	dbo	authors	0	23.0	4.0
32.0	12.50	16.0	0.00		100.00
Jun 14 2007 11:46PM	dbo	authors	0	23.0	4.0
32.0	12.50	16.0	0.00		100.00
Jun 10 2007 12:05AM	dbo	authors	1	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 11 2007 11:35PM	dbo	authors	1	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 13 2007 11:46PM	dbo	authors	1	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 14 2007 11:46PM	dbo	authors	1	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 14 2007 11:46PM	dbo	authors	1	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 10 2007 12:05AM	dbo	authors	2	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 11 2007 11:35PM	dbo	authors	2	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 13 2007 11:46PM	dbo	authors	2	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 14 2007 11:46PM	dbo	authors	2	NULL	4.0
32.0	12.50	16.0	0.00		100.00
Jun 14 2007 11:46PM	dbo	authors	2	NULL	4.0
32.0	12.50	16.0	0.00		100.00

(1 row affected)

(return status = 0)

使用方法

- `sp_spaceusage` は、テーブルとインデックスの領域の使用量に関する情報、および現在のデータベースのトランザクション・ログを表示します。

- `sp_spaceusage` による出力で表示されるカラムのセットは、アクションとエンティティのタイプによって異なります。デフォルトではカラムの標準セットが表示されます。ただし、`select_list` パラメータを使用して他のカラムを表示したり、`select` リスト内に * ワイルドカードを含めることで、すべてのカラムを表示したりできます。表 1-31 と表 1-32 に、各エンティティ・タイプ `table`、`index`、`tranlog` のすべての出力カラム名とその説明を記載します。`select_list`、`where_clause`、`orderby_clause` パラメータでは、これらの表に記載されているカラム・セットに属するカラム名を指定してください。

表 1-31: テーブル・タイプまたはインデックス・エンティティ・タイプの出力カラム

カラム名	説明
ArchiveDateTime	データのタイムスタンプ
ServerName	サーバ名
MaxPageSize	論理ページ・サイズ (変数 @@maxpagesize で設定)
DBName	オブジェクトのデータベース名
OwnerName	オブジェクトの所有者名
TableName	テーブル名
Id	テーブルの ID
IndId	インデックスの ID
IndexName	インデックス名
PtnId	パーティションの ID
PtnName	パーティション名
DataPtnID	インデックスによってカバーされているデータが配置されているデータ・パーティションの ID
RowSize	<ul style="list-style-type: none"> • テーブルの場合 - データ・ローの平均サイズ。 • インデックスの場合 - ノンクラスタード・インデックスおよびクラスタード・インデックス (データオンリーロック・テーブル) のリーフ・ローの平均サイズ。
RowCount_ts	sysstatstabs テーブルに格納されている、パーティションのロー数
NumFwdRows	パーティション内の転送されたローの数
NumDelRows	パーティション内の削除されたローの数
NumIRows	パーティションのロー数
PctFwdRows	パーティション内の転送されたローのパーセンテージ
NonLeafRowSize	パーティション内のリーフ・ロー以外のローの平均サイズ
FF	パーティション内のフィル・ファクタ
MRPP	パーティション内のページあたりのローの最大数
ERS	パーティション内の予期されるロー・サイズ

カラム名	説明
RPG	パーティション内のページ・ギャップの予約値
IndexHeight	パーティション内のインデックス・ツリーの高さ
OAMAPageCount	OAM および AP ページの数 (ページ単位)
Extent0PageCount	エクステンツ 0 内のページの数 (ページ単位)
Status	sysindexes テーブルに格納されている status
Sysstat	sysobjects テーブルに格納されている sysstat
Sysstat2	sysobjects テーブルに格納されている sysstat2
LockScheme	テーブルのロック・スキーム
NumVarCols	テーブルに含まれている可変長カラムの数
HasAPLCl	テーブルに APL Cl が含まれているかどうかを示す
SpUtil	パーティションについて領域使用率から得られた統計
DPCR	パーティションについてデータ・ページ・クラスタ率から得られた統計
DRCR	パーティションについてデータ・ロー・クラスタ率から得られた統計
IPCR	パーティションについてインデックス・ページ・クラスタ率から得られた統計
LGIO	パーティションについて大容量 I/O 効率から得られた統計
ExtentUtil	パーティションのエクステンツ使用率
EmptyPages	パーティションにある空白ページの数 (ユニット単位)
DataPages	パーティションにあるデータ・ページの数 (ユニット単位)
UsedPages	パーティション内の使用中のページ数 (ユニット単位)
RsvdPages	パーティション内の予約済みページの数 (ユニット単位)
LeafPages	パーティション内のリーフ・ページの数 (ユニット単位)
ExpDataPages	データが圧縮された場合のパーティション内のデータ・ページの予測数 (ユニット単位)
ExpUsedPages	データが圧縮された場合のパーティション内の使用中のページの予測数 (ユニット単位)
ExpRsvdPages	データが圧縮された場合のパーティション内の予約済みページの予測数 (ユニット単位)
ExpLeafPages	データが圧縮された場合のパーティション内のインデックス・ツリーの予測の高さ
PctBloatUsedPages	パーティション内の使用中のページの膨張率
PctBloatRsvdPages	パーティション内の予約済みページの膨張率
PctBloatLeafPages	パーティション内のリーフ・ページの膨張率
PctEmptyPages	パーティション内の空白になっていたデータ・ページのパーセンテージ

表 1-32: tanlong エンティティ・タイプに使用可能なパラメータ

カラム名	説明
ArchiveDateTime	データのタイムスタンプ
ServerName	サーバ名

カラム名	説明
MaxPageSize	@@maxpagesize で設定されている論理ページ・サイズ
DBName	オブジェクトのデータベース名
OwnerName	オブジェクトの所有者名
TableName	syslogs などのトランザクション・ログ名
Id	syslogs テーブルの ID
IsMLD	「ログとデータが混在した」トランザクション・ログかどうか
IsLogFull	トランザクション・ログが満杯かどうか
LCTPages	ログの「ラスト・チャンス・スレッシュホールド」値 (ユニット単位)
TotalPages	ログ・ページの総数 (ユニット単位)
UsedPages	すでに使用されているログ・ページの数 (ユニット単位)
CLRPages	ロールバックについて予約されているログ・ページの数 (ユニット単位)
FreePages	まだ使用されていないログ・ページの数 (ユニット単位)
PctUsedPages	すでに使用されているログ・ページのパーセンテージ
PctFreePages	空いているログ・ページのパーセンテージ

- PctBloatUsedPages および PctBloatReservedPages カラムには、ページ数が、テーブルで使用および予約されている最小数よりもどのくらい多いのか、その予測数を表示します。これらの予測値を調べることで、テーブルに対して reorg rebuild を実行する場合の効率性を確認できます。

ExtentUtil は、実際に使用されているページの数と、オブジェクトについて予約されているページの数との比率です。値が 100 に近い場合、オブジェクトについて予約されている、エクステント内のページのほとんどが現在、使用されていることを示しています。表 1-33 に測定値の概要を記載します。

表 1-33: PctBloatUsedPages および PctBloatReservedPages 値の解釈

PctBloatUsed – Pages 値	PctBloatReserved – Pages 値	解釈
0 に近い数値、 小さい数値	0 に近い数値、小 さい数値	テーブルの圧縮状態が良好で、割り付けられているページおよびアロケーション・ユニットがすべて完全に使用されていることを示します。ExtentUtil は 1.0 に近い数値である必要があります。
0 に近い数値、 小さい数値	0 から遠い数値、 大きい数値	使用されているページの圧縮状態は良好であるが、テーブルのエクステントの利用率が低く、削除ページまたは空白ページが多いため、ページ間に大規模な断片化が発生していることを示しています。systabstats の unusedpgcnt も大きい可能性があります。PctBloatReservedPages の値が大きい場合、ExtentUtil は 1.0 よりもずっと小さいことを示唆しています。reorg rebuild コマンドを実行すると、ほとんどの問題を解決できる場合があります。

PctBloatUsed - Pages 値	PctBloatReserved - Pages 値	解釈
0 から遠い数値、大きい数値	0 に近い数値、小さい数値	<p>ページ内に大規模な断片化が発生しているが、ページ間の断片化の規模は小さいことを示しています。エクステンツ使用率は高い可能性があるため、ExtentUtil 値は 1.0 に近い数値である必要があります。</p> <p>reorg compact を実行すると、これらの問題の解決に役立つ可能性があります。</p>
0 から遠い数値、大きい数値	0 から遠い数値、大きい数値	<p>PctBloatUsedPages の値が大きい場合、ページ内断片化の規模が大きい、つまり、使用されているページのデータ・ローの圧縮が完全でない (使用されているページ内のほとんどが空き領域である) ことを示します。ページ間とページ内で発生している断片化が原因で PctBloatReservedPages の値が大きくなっているため、Extent_Util の値は 1.0 よりも小さいままである可能性があります。reorg compact および reorg rebuild コマンドを実行すると、これらの問題を解決できる場合があります。</p>

- 領域使用量データがアーカイブされるデータベースでは、`sp_dboption ... select into` を有効にする必要があります。
- アーカイブ時にアーカイブ・テーブルが存在していない場合は作成されます。それ以外の場合、結果は現在のテーブルに追加されます。そのため、`sp_spaceusage` を実行するすべてのユーザには、アーカイブ・データベースに対する `create table` パーミッションが必要になります。
- データをアーカイブまたはレポート生成している間、`sp_spaceusage` を実行しているユーザが所有しているテーブルだけが、アーカイブ・テーブルとして考慮されます。名前は同じであっても所有者が異なるテーブルは無視されます。デフォルトでは、テーブルまたはインデックスの `spaceusage_object` テーブルとトランザクション・ログの `spaceusage_tranlog` テーブルに結果がアーカイブされ、これらのテーブルからレポートが生成されます。

- `from_date` 引数と `to_date` 引数は、アーカイブされているデータからレポートを生成するときに `report` アクションでのみ使用できません。Adaptive Server では、レポートを生成する場合、特定の時間範囲内にある、アーカイブ・テーブル内のデータのみが使用されます。`from_date` または `NULL` を指定しない場合、Adaptive Server では、`to_date` の前にアーカイブされているすべてのデータが使用されます。`to_date` または `NULL` を指定しない場合、Adaptive Server では、`to_date` の値として現在の日付が使用されます。`from_date` または `to_date` のいずれかを指定しない場合、Adaptive Server では、アーカイブ・テーブル内の最新のデータを使用してレポートが生成されます。
- `sp_spaceusage` の結果は、統計データに基づいて予測されます。これらの予測値は、提供されている統計値と同じ精度があります。`update statistics` を実行すると、結果の精度を向上させることができます。

パーミッション

どのユーザも `sp_spaceusage` を実行して領域使用率を表示できます。ただし、表示するパーミッションを持っていないテーブルについては、一部の情報を表示できない場合があります。

sp_spaceused

説明 ロー数、データ・ページ数、インデックスのサイズ、指定のテーブルが使用する領域、現在のデータベースが使用する領域の概算を表示します。

構文 sp_spaceused [*objname* [,1]]

パラメータ *objname*

情報を表示するテーブルの名前です。テーブル名を省略すると、現在のデータベースで使用している領域の概要が表示されます。

1

テーブルのインデックスに関する情報と、**text** と **image** の記憶領域に関する情報を分けて出力します。

例 **例 1 titles** テーブルに割り付けられた (予約された) 領域の量、データで使用している量、インデックス (1 つまたは複数) で使用している量、使用可能な (未使用の) 領域をレポートします。

```
sp_spaceused titles
```

name	rowtotal	reserved	data	index_size	unused
titles	18	46 KB	6 KB	4 KB	36 KB

例 2 titles テーブルの情報の他に、テーブルの各インデックスの情報を出力します。

```
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

例 3 テーブルが使用している領域とは別に、**text/image** ページの記憶領域が使用している領域を表示します。**text/image** 記憶領域のオブジェクト名は、テーブル名に “i” を加えたものです。

```
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

```

例4 現在のデータベースで使用している領域の概要を出力します。

```
sp_spaceused
```

```

database_name  database_size
-----
master          5 MB
reserved      data      index_size  unused
-----
2176 KB      1374 KB      72 KB      730 KB

```

例5 トランザクション・ログの、予約領域の量と使用可能領域の量をレポートします。

```
sp_spaceused syslogs
```

```

name      rowtotal reserved      data      index_size  unused
-----
syslogs   Not avail. 32 KB      32 KB      0 KB      0 KB

```

使用法

- `sp_spaceused` は、データ・ページの数、指定したテーブルまたは現在のデータベースにあるすべてのテーブルで使用している領域、テーブル内のロー数の概算を表示します。`sp_spaceused` は `rowcnt` 組み込み関数を使用して、`rowtotal` の値を計算します。この関数は、そのオブジェクトのアロケーション・ページにある値をもとにして、データ・ページごとのローの平均値を使用します。この方法は非常に高速ですが、結果は概算であり、更新や挿入アクティビティによって実際の値が変動します。`update statistics` コマンド、`dbcc checktable`、`dbcc checkdb` はページごとのローの推定値(概算)を更新します。したがって、これらのコマンドを実行した後に、`rowtotal` は最も正確な値になります。ローの正確なカウントが必要な場合は、`select count(*)` を使用してください。
- `sp_spaceused` は、テーブル、クラスタード・インデックス、ノンクラスタード・インデックスによって影響される領域の量をレポートします。
- `sp_spaceused` は、データ、インデックス・サイズ、使用可能な(未使用の)領域の合計値を、割り付けられた(予約された)領域の量としてレポートします。
- `text` および `image` カラムは、別々のデータベース・オブジェクトとして保管されますが、これらのカラムが使用する領域は、`index_size` カラムに分けてレポートされ、テーブルの合計行に加算されます。`index_size` カラムの `text/image` 記憶領域のオブジェクト名は、“t” にテーブル名を加えたものです。
- `syslogs` で使用すると、`sp_spaceused` は、`rowtotal` を “Not available” としてレポートします。例 5 を参照してください。

パーミッション

`sp_spaceused` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照

カタログ・ストアド・プロシージャ [sp_statistics](#)

コマンド [create index](#)、[create table](#)、[drop index](#)、[drop table](#)

システム・プロシージャ [sp_helpindex](#)

sp_ssladmin

説明	Adaptive Server に対するサーバ認証のリストを追加、削除、表示します。
構文	<pre>sp_ssladmin {[addcert, <i>certificate_path</i> [, <i>password</i> NULL]] [dropcert, <i>certificate_path</i>] [lscert] [help]} [lsciphers] [setciphers, {"FIPS" "Strong" "Weak" "All" <i>quoted_list_of_ciphersuites</i>}]</pre>
パラメータ	<p>addcert 認証ファイルにローカル・サーバ用の証明書を追加します。</p> <p><i>certificate_path</i> ローカル・サーバの認証ファイルへの絶対パスを指定します。</p> <p><i>password</i> 新しいサーバ証明書を認証ファイルに追加するときにプライベート・キーの暗号化に使用されるパスワードです。</p> <p>NULL この値を指定すると、コマンド・ラインからの起動中にパスワードが要求され、Adaptive Server を手動で起動する必要があります。</p> <p>dropcert 証明書を認証ファイルから削除します。</p> <p>lscert 認証ファイル内の証明書をリストします。</p> <p>help sp_ssladmin のオンライン・ヘルプを表示します。</p> <p>lsciphers 暗号スイートの優先度セットの値を表示します。</p>

setciphers, {"FIPS" | "Strong" | "Weak" | "All" | *quoted_list_of_ciphersuites*}

特定の暗号スイートの優先度を設定します。次のいずれかのオプションを選択します。

- “FIPS” – FIPS に準拠した暗号化、ハッシュ、キー交換アルゴリズムのセット。このリストに含まれるアルゴリズムは AES、3DES、DES、SHA1 です。
- “Strong” – 64 ビットより長いキーを使用する暗号化アルゴリズムのセット。
- “Weak” – サポート対象のすべての暗号スイートのセットの中で強力セットのカテゴリに含まれない暗号化アルゴリズムのセット。
- “All” – デフォルトの暗号スイートのセット。
- *quoted_list_of_ciphersuites* – 暗号スイートのセットを、優先度順にカンマで区切ったリストで指定します。引用符 (“ ”) でリストの先頭と最後をマークします。引用符で囲んだリストに、個々の暗号スイート名のほか、定義済みの任意のセットを含めることができます。未知の暗号スイート名を指定するとエラーが報告され、優先度は変更されません。定義済みのセットに含まれている暗号スイートのリストについては、『システム管理ガイド』の「第 19 章 データの機密保持」を参照してください。

例

例 1 絶対パス `¥sybase¥ASE-15_0¥certificates` (Windows の場合は `x:¥sybase¥ASE-15_0¥certificates` への絶対パス) の認証ファイルに、ローカル・サーバのエントリ `Server1.crt` を追加します。プライベート・キーは、パスワード “mypassword” を使用して暗号化されています。パスワードは、プライベート・キーを作成時に指定したパスワードになります。

```
sp_ssladmin addcert, "¥sybase¥ASE-15_0¥certificates¥Server1.crt",
"mypassword"
```

例 2 証明書 `Server1.crt` を、`¥sybase ¥ASE-15_0¥certificates` (Windows の場合は `x: ¥sybase¥ASE-15_0 ¥certificates`) に格納されている認証ファイルから削除します。

```
sp_ssladmin dropcert , "¥sybase¥ASE-15_0¥certificates¥Server1.crt"
```

例 3 ローカル・サーバに関するすべてのサーバ証明書をリストします。

```
sp_ssladmin lscert
go

certificate_path
-----
/sybase/ASE-15_0/certificates/Server1.crt
```

例 4 最初に開始されるときは、まだ暗号スイートの優先度が設定されていないので、sp_ssladmin lscipher は優先度を表示しません。

```
1sp_ssladmin lscert
2> go

Cipher Suite Name Preference
-----
(0 rows affected)
(return status = 0)
```

この例では、**FIPS** アルゴリズムを使用する暗号スイートのセットを指定しています。

```
1> sp_ssladmin setcipher, 'FIPS'
2> go
```

優先度 0 (ゼロ) の sp_ssladmin 出力は、暗号スイートが **Adaptive Server** で使用されないことを示します。他のゼロ以外の値は、**SSL** ハンドシェイクの間に **Adaptive Server** がアルゴリズムを使用する優先度の順序を示します。**SSL** ハンドシェイクのクライアント側はこれらの暗号スイートから、受け付ける暗号スイートのリストに一致するものを選びます。

例 5 引用符で囲んだ暗号スイートのリストで、**Adaptive Server** に優先度を設定しています。

```
1> sp_ssladmin setcipher, 'TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA'
2> go
```

使用法

- **Adaptive Server** リスナは、クライアントに証明書を提出しなければなりません。証明書内の共通名は、認証ファイル内でクライアントが使用している共通名と一致する必要があります。両方の共通名が一致しない場合は、サーバ認証とログインが実行されません。
- **NULL** がパスワードとして指定されている場合は、**-y** フラグを指定して **dataserver** を起動してください。このフラグは、管理者に、コマンド・ラインでプライベート・キーのパスワードを入力するように要求します。
- **NULL** をパスワードとして使用するのには、**SSL** の初期設定中にパスワードを保護するためです。**SSL** の暗号化セッションは初期設定後に開始されます。

SSL 接続が確立された状態で Adaptive Server を再起動した後、実際のパスワードを使用して `sp_ssladmin` を再実行します。このパスワードは暗号化されて保管されます。その後、コマンド・ラインから Adaptive Server を起動すると、この暗号化されたパスワードが使用されます。したがって、起動中にコマンド・ラインでパスワードを指定する必要はありません。

- `interfaces` ファイル (Windows では `sql.ini`) 内の `hostname` として “localhost” を指定すると、クライアントはリモート接続できなくなります。これによりローカル接続だけが確立されるため、パスワードがネットワーク間の接続で送信されることはありません。

Isciphers と setciphers の使用による暗号スイートの設定

Isciphers オプションと setciphers オプションを使用すると、Adaptive Server の使う暗号スイートのセットを制限でき、システム・セキュリティ担当者はサーバに対するクライアント接続や Adaptive Server からのアウトバウンド接続で使われる暗号化アルゴリズムの種類をコントロールすることができます。デフォルトでは、Adaptive Server は、暗号スイートに対して内部的に定義された一連の設定を使用します。詳細については、『システム管理ガイド』の「第 19 章 データの機密保持」を参照してください。

`sp_ssladmin setciphers` は、指定された順序リストに暗号スイートの優先度を設定します。これは使用可能な SSL 暗号スイートを、“FIPS”、“Strong”、“Weak”、“All”、または引用符で囲まれた暗号スイート・リストのセットに制限します。これが有効になるのは次のリスナが開始されたときで、Adaptive Server を再起動してすべてのリスナが新しい設定を使うようにする必要があります。

設定されている任意の暗号スイートの優先度を、`sp_ssladmin isciphers` で表示することができます。優先度が設定されていない場合、`sp_ssladmin isciphers` は 0 個のローを返します。これは優先度が設定されておらず、Adaptive Server がデフォルトの (内部) 優先度を使用することを意味します。

パーミッション

`sp_ssladmin` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage security configuration</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>ssso_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 マニュアル『システム管理ガイド』の「第 19 章 データの機密保持」

sp_syntax

説明	<p>使用しているサーバ上の製品とそれに対応する <code>sp_syntax</code> スクリプトに応じて、Transact-SQL 文、システム・プロシージャ、ユーティリティ、Adaptive Server のその他のルーチンの構文を表示します。</p>
構文	<pre>sp_syntax word [, mod][, language]</pre>
パラメータ	<p>word</p> <p>コマンドまたはルーチンの名前あるいは部分名です。たとえば、“help” はヘルプを提供するすべてのシステム・プロシージャをリストします。スペースまたは Transact-SQL の予約語を使用する場合は、その語を引用符で囲みます。</p> <p>mod</p> <p>“Transact-SQL” や “Utility” などのモジュールの名前または部分名です。それぞれの <code>sp_syntax</code> インストール・スクリプトは、別のモジュールを追加します。使用しているサーバ上に存在するモジュールを確認するには、パラメータなしで <code>sp_syntax</code> を使用します。</p> <p>language</p> <p>検索する構文を記述する言語です。<i>language</i> には、<code>syslanguages</code> テーブルに存在する有効な言語名を指定してください。</p>
例	<p>例 1 使用しているサーバで利用可能なすべての <code>sp_syntax</code> モジュールを表示します。</p> <pre>sp_syntax sp_syntax provides syntax help for Sybase products. These modules are installed on this Server: Module ----- Transact-SQL UNIX Utility System Procedure Usage:sp_syntax command [, module [, language]]</pre> <p>例 2 “disk” という語句またはそのフラグメントを含む、すべてのルーチンについての構文と機能の記述を表示します。disk は Transact-SQL の予約語なので、引用符で囲んでください。</p> <pre>sp_syntax "disk"</pre>

使用法

- `sp_syntax` のテキストは、`sybsyntax` データベース内にあります。使用しているプラットフォームの設定ガイドで説明されているインストール・スクリプトを使用して、`sp_syntax` と `sybsyntax` データベースを `Adaptive Server` にロードします。`sp_syntax` にアクセスできない場合は、使用しているサーバに `sp_syntax` をインストールする方法を、システム管理者に問い合わせてください。
- 検索するコマンド名にワイルドカード文字を使用できます。ただし、リテラルの “_” を含むコマンドまたは関数を検索する場合は、予想しない結果が発生することがあります。これは、アンダースコア・ワイルドカード文字が任意の 1 文字を表すためです。

パーミッション

`sp_syntax` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

使用するテーブル

`sybsyntax..sybsyntax`

参照

システム・プロシージャ [sp_helpdb](#)

sp_sysmon

説明

パフォーマンス情報を表示します。

構文

```
sp_sysmon begin_sample
sp_sysmon { end_sample | interval }[, section[, applmon] ]
          [, 'cache wizard' [, top_N[, filter] ]]
```

パラメータ

begin_sample

サンプリングを開始します。begin_sample を指定するときは、セクションを指定できません。

section

sp_sysmon が出力する 1 つのセクションの省略名です。レポート・セクションの値と対応する名前は次のとおりです。

パラメータ	レポート・セクション
apppmgmt	Application Management (アプリケーション管理)
dcache	Data Cache Management (データ・キャッシュ管理)
diskio	Disk I/O Management (ディスク I/O 管理)
esp	ESP Management (ESP 管理)
indexmgmt	Index Management (インデックス管理)
kernel	Kernel Utilization (カーネル利用率)
locks	Lock Management (ロック管理)
memory	Memory Management (メモリ管理)
mdcache	Metadata Cache Management (メタデータ・キャッシュ管理)
monaccess	Monitor Access to Executing SQL (実行 SQL へのモニタ・アクセス)
netio	Network I/O Management (ネットワーク I/O 管理)
parallel	Parallel Query Management (並列クエリ管理)
pcache	Procedure Cache Management (プロシージャ・キャッシュ管理)
recovery	Recovery Management (リカバリ管理)
taskmgmt	Task Management (タスク管理)
xactmgmt	Transaction Management (トランザクション管理)
xactsum	Transaction Profile (トランザクション・プロファイル)
wpm	Worker Process Management (ワーカー・プロセス管理)

applmon

アプリケーションの詳細を出力するか、アプリケーションとログインの詳細を出力するか、またはアプリケーションの詳細を省略するかどうかを指定します。デフォルトでは、アプリケーションの詳細は省略されます。有効な値とレポートされる情報は次のとおりです。

- `appl_only` – CPU、I/O、優先順位の変更、リソース制限違反をアプリケーション名とログイン名で表示する。
- `appl_and_login` – CPU、I/O、優先順位の変更、リソース制限違反をアプリケーション名とログイン名で表示する。
- `no_appl` – レポートのアプリケーション別またはログイン別のセクションを省略する。これがデフォルト値です。

このパラメータは、全体のレポートを出力する場合と、*section* の `apppgmt` を指定した場合だけに有効です。

end_sample

サンプリングを終了し、レポートを出力します。

interval

サンプリングの時間を指定します。これは `HH:MM:SS` のフォーマットにしてください。たとえば “00:20:00” のように指定します。

'cache wizard'

最適なパフォーマンスを得るためにデータ・キャッシュをモニタし、設定します。

top_N

`varchar` データ型であり、指定された間隔における論理読み込み回数 (LR/sec カラムの表示) のランク付け基準に基づいて、オブジェクト・セクションに出力されるオブジェクトのリストを限定します。

ランク付けの順序は、正の整数を指定すると昇順、負の整数を指定すると降順になります。“0” (ゼロ) の値を指定すると、間隔の終了時にキャッシュを占有していた全オブジェクトのリストを取得できます。デフォルト値は 10 です。デフォルト値は 10 です。

filter

`varchar` データ型であり、レポートに含めるキャッシュのパターンを指定します。

たとえば、`default data cache` と指定した場合、デフォルト・データ・キャッシュの情報だけがレポートされます。`emp%` と指定すると、すべてのキャッシュで名前がこのパターンに一致するキャッシュの情報のみが出力されます。

値を指定しないと、すべてのキャッシュの情報が出力されます。最初に `default data cache` のキャッシュが表示された後、それ以外のキャッシュの情報がアルファベット順に表示されます。

例

例 1 10 分後にモニタ情報を出力します。

```
sp_sysmon "00:10:00"
```

例 2 5 分後に、`sp_sysmon` レポートの “Disk Management” セクションだけを出力します。

```
sp_sysmon "00:05:00", diskio
```

例 3 サンプルングを開始し、ストアド・プロシージャを実行します。また、クエリを選択し、サンプルングを終了して、レポートの “Data Cache” セクションだけを出力します。

```
sp_sysmon begin_sample
go
execute proc1
go
execute proc2
go
select sum(total_sales) from titles
go
sp_sysmon end_sample, dcache
go
```

例 4 完全なレポートを出力します。また、ログインごとのアプリケーションやログインについての詳細情報も一緒に表示します。

```
sp_sysmon "00:05:00", @applmon = appl_and_login
```

例 5 カウンタをクリアしないで、カーネルの使用状況を報告します。

```
sp_sysmon "00:01:00", kernel, noclear
```

次の方法も使用できます。

```
sp_sysmon "00:01:00", noclear
```

注意 noclear パラメータは、サンプル・インターバルを sp_sysmon で指定するときだけ使用できます。begin_sample または end_sample を指定するときは、noclear は使用できません。

例 6 キャッシュ・ウィザードを使用してレポートを出力します。

```
sp_sysmon '00:00:30', 'cache wizard'
```

```
=====
Cache Wizard
=====

-----
default data cache
-----

Run Size      : 100.00 Mb   Usage%        :      2.86
LR/sec       : 41.10     PR/sec        : 22.57   Hit%: 45.09
Cache Partitions: 4       Spinlock Contention%: 0.00
Buffer Pool Information
-----

IO Size Wash Size Run Size APF% LR/sec PR/sec Hit% APF-Eff% Usage%
-----
4 Kb 3276 Kb 16.00 Mb 10.00 0.47 0.13 71.43 n/a 0.20
2 Kb 17200 Kb 84.00 Mb 10.00 40.63 22.43 44.79 n/a 3.37

(1 row affected)

Object Statistics
-----
Object LR/sec PR/sec Hit% Obj_Cached% Cache_Occp%
-----
empdb.dbo.t1 0.57 0.30 47.06 56.25 0.02
empdb.dbo.t2 0.30 0.30 0.00 56.25 0.02
empdb.dbo.t3 0.30 0.30 0.00 56.25 0.02
empdb.dbo.t4 0.30 0.30 0.00 56.25 0.02
empdb.dbo.t5 0.30 0.30 0.00 56.25 0.02
empdb.dbo.t6 0.30 0.30 0.00 56.25 0.02
empdb.dbo.t8 0.30 0.30 0.00 56.25 0.02
empdb.dbo.t7 0.57 0.20 64.71 62.50 0.02
tempdb.dbo.tempcachedobjstats 3.63 0.00 100.00 50.00 0.01
tempdb.dbo.tempobjstats 0.47 0.00 100.00 25.00 0.00

Object Obj Size Size in Cache
-----
```

```
empdb.dbo.t1          32 Kb      18 Kb
empdb.dbo.t2          32 Kb      18 Kb
empdb.dbo.t3          32 Kb      18 Kb
empdb.dbo.t4          32 Kb      18 Kb
empdb.dbo.t5          32 Kb      18 Kb
empdb.dbo.t6          32 Kb      18 Kb
empdb.dbo.t8          32 Kb      18 Kb
empdb.dbo.t7          32 Kb      20 Kb
tempdb.dbo.tempcachedobjstats 16 Kb      8 Kb
tempdb.dbo.tempobjstats 16 Kb      4 Kb
```

company_cache

```
Run Size      :      1.00 Mb  Usage%      :      0.39
LR/sec       :      0.07    PR/sec      :      0.07  Hit%:    0.00
Cache Partitions:      1    Spinlock Contention%: 0.00
```

Buffer Pool Information

```
-----
IO Size Wash Size Run Size  APF%  LR/sec  PR/sec  Hit%  APF-Eff% Usage%
-----
2 Kb      204 Kb      1.00 Mb 10.00  0.07   0.07   0.00  n/a     0.39
Object Statistics
-----
```

```
-----
Object          LR/sec  PR/sec  Hit%  Obj_Cached%  Cache_Occp%
-----
empdb.dbo.history  0.07   0.07   0.00   25.00        0.39
```

```
-----
Object          Obj Size  Size in Cache
-----
empdb.dbo.history  16 Kb    4 Kb
```

companydb_cache

```
Run Size      :      5.00 Mb  Usage%      :     100.00
LR/sec       :    380.97  PR/sec      :     56.67  Hit%:    85.13
Cache Partitions:      1    Spinlock Contention%: 0.00
```

Buffer Pool Information

```
-----
IO Size Wash Size Run Size  APF%  LR/sec  PR/sec  Hit%  APF-Eff% Usage%
-----
2 Kb      1024 Kb      5.00 Mb 10.00 380.97  56.67  85.13  98.42  100.00
```

Object Statistics

Object	LR/sec	PR/sec	Hit%	Obj_Cached%	Cache_Occp%
company_db.dbo.emp_projects	41.07	22.80	44.48	19.64	9.45
company_db.dbo.dept_det	93.03	20.67	77.79	99.08	54.53
company_db.dbo.emp_perf	116.70	2.63	97.74	97.77	34.18
company_db.dbo.dept_locs	0.43	0.17	61.54	50.00	0.16

Object	Obj Size	Size in Cache
company_db.dbo.emp_projects	2464 Kb	484 Kb
company_db.dbo.dept_det	2818 Kb	2792 Kb
company_db.dbo.emp_perf	1790 Kb	1750 Kb
company_db.dbo.dept_locs	16 Kb	8 Kb

TUNING RECOMMENDATIONS

Usage% for 'default data cache' is low (< 5%)
 Usage% for 4k buffer pool in cache:default data cache is low (< 5%)
 Usage% for 2k buffer pool in cache:default data cache is low (< 5%)

Usage% for 'company_cache' is low (< 5%)
 Usage% for 2k buffer pool in cache:company_cache is low (< 5%)
 Consider adding a large I/O pool for 'companydb_cache'

使用法

- Adaptive Server バージョン 15.0.1 以降、sp_sysmon のデフォルトの動作ではモニタ・カウンタはクリアされません。

モニタ・カウンタをクリアする必要がある場合は、clear オプションを指定して sp_sysmon を使用してください。互換性のため、Adaptive Server は noclear オプションを有効なパラメータとして受け入れますが、sp_sysmon の動作には影響ありません。

ただし、begin_sample オプションと end_sample オプションを使用して sp_sysmon を実行し、サンプル時間を開始および終了する場合、sp_sysmon では常にモニタ・カウンタがクリアされます。begin_sample または end_sample および noclear オプションを指定して sp_sysmon を実行すると、Adaptive Server ではエラー・メッセージが発行されます。

- sp_sysmon は、Adaptive Server のパフォーマンスに関する情報を表示します。内部カウンタを 0 に設定し、指定された時間だけ、サーバのアクティビティによってカウンタが増えるのを待ちます。指定された時間が経過すると、sp_sysmon はカウンタの値から情報を出力します。詳細については、『パフォーマンス & チューニング・ガイド』を参照してください。
- レポートのセクションを 1 つだけ出力するには、sp_sysmon applmon の有効な値を使用します。

- バッチ・モードで `begin_sample` や `end_sample` を使用して `sp_sysmon` を実行する場合は、2 つのコマンドの実行の間隔を 1 秒以上に設定してください。バッチの実行時間を長くするには、`waitfor delay "00:00:01"` を使用します。
- サンプル・インターバル中、結果は符号付き整数値で格納されま
す。特に多数の CPU を使用するアクティビティの高いシステムで
は、これらのカウンタはオーバーフローする可能性があります。
`sp_sysmon` 出力に負の結果が表示された場合は、サンプル時間を
短縮してください。

パーミッション

`sp_sysmon` を実行するには、`execute` パーミッションが付与されたユーザであることが必要です。このパーミッションは、`sybsystemprocs` のデータベース所有者が他のユーザに付与できます。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

マニュアル『パフォーマンス&チューニング・ガイド』の「`sp_sysmon` を利用したパフォーマンスのモニタリング」

sp_tab_suspectptn

説明	suspect (疑わしい)パーティションを含んでいるテーブルをリストします。文字ベースのパーティション・キーを使用する範囲分割されたテーブルは、ソート順を変更すると suspect になる可能性があり、ハッシュ分割されたテーブルは、プラットフォーム間のダンプとロードを行うと suspect になる可能性があります。
構文	sp_tab_suspectptn [<i>table_name</i>]
パラメータ	<i>table_name</i> suspect パーティションを含んでいるテーブルの名前です。
使用法	次に従います。 <ul style="list-style-type: none">• テーブル名を指定する場合 – Adaptive Server は <i>table_name</i> で指定されたテーブルのみをチェックします。• テーブル名を指定しない場合 – Adaptive Server は、現在のデータベースのすべてのテーブルをチェックします。
パーミッション	sp_tab_suspectptn は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。
参照	ストアド・プロシージャ sp_indsuspect

sp_tempdb

説明

sp_tempdb を使用すると、次の処理が行えます。

- テンポラリ・データベース・グループを作成および管理する。
- default または他のテンポラリ・データベース・グループ、あるいは特定のローカル・テンポラリ・データベースに、ユーザまたはアプリケーションをバインドする。
- ローカル・テンポラリ・データベースおよびテンポラリ・データベース・グループへのバインドを管理する。

これらのバインドは、master データベース内の sysattributes テーブルに格納されます。

sp_tempdb は、複数のテンポラリ・データベースに関連するバインドを sysattributes で管理するためのバインド・インタフェースを提供します。

構文

```
sp_tempdb [
  [ { "create" | "drop" }, "groupname" ] |
  [ { "add" | "remove" }, "tempdbname", "groupname" ] |
  [ { "bind", "objtype", "objname", "bindtype", "bindobj"
    [, "scope", "hardness" ] } |
  { "unbind", "objtype", "objname" [, "scope" ] "instance_name" } ] |
  [ "unbindall_db", "tempdbname" ] |
  [ show [, "all" | "gr" | "db" | "login" | "app" [, "name" ] ] |
  [ who, "dbname" ]
  [ help ] ]
```

パラメータ

create

テンポラリ・データベース・グループを作成します。

drop

テンポラリ・データベース・グループを削除します。

groupname

テンポラリ・データベース・グループの名前です。

add

テンポラリ・データベースをテンポラリ・データベース・グループに追加します。

remove

テンポラリ・データベース・グループからテンポラリ・データベースを削除します。

tempdbname

追加または削除するテンポラリ・データベースの名前です。Cluster Edition では、*tempdbname* は、ローカル・ユーザ・テンポラリ・データベースである必要があります。

bind

テンポラリ・データベースまたはテンポラリ・データベース・グループに、ログインまたはアプリケーションをバインドします。

unbind

テンポラリ・データベースまたはテンポラリ・データベース・グループから、ログインまたはアプリケーションをバインド解除します。

objtype

オブジェクト・タイプです。有効な値は次のとおりです。

- login_name (または LG)
- application_name (または AP)

これらの値では、大文字と小文字を区別しません。

objname

バインドまたはバインド解除するオブジェクトの名前です。

bindtype

バインド型です。有効な値は次のとおりです。

- group (または GR)
- database (または DB)

これらの値では、大文字と小文字を区別しません。

bindobj

バインドするオブジェクトの名前です。*bindtype* に応じてグループまたはデータベースを指定します。

scope

NULL。

instance_name

クラスター環境の場合 - バインドを解除するローカル・テンポラリ・データベースを所有するインスタンスの名前です。このオプションは Cluster Edition のみで使用できます。

hardness

hardness — *hard*、*soft*、または *NULL* を指定します。デフォルトは *soft* です。

hardness の値を *hard* に設定すると、バインドに従ってテンポラリー・データベースを割り当てできなければ、ログインできません。

値を *soft* に設定した場合、同様の割り当てに失敗すると、デフォルト・グループのテンポラリー・データベースまたはローカル・システムのテンポラリー・データベースが割り当てられます。

unbindall_db

指定されたテンポラリー・データベースから、すべてのログインとアプリケーションをバインド解除します。この解除によってデータベースとグループ・メンバシップの関係が削除されることはありません。このオプションを使用する場合は、*tempdbname* 変数を指定する必要があります。

この操作によって、アクティブなセッションに対する既存の割り当てが影響を受けることはありません。

show

既存のグループ、グループ・メンバ、ログインおよびアプリケーションのバインド、指定されたデータベースに割り当てられているアクティブなセッションなど、**sysattributes** テーブルに格納されている情報を表示します。値は次のとおりです。

- **all** または引数なし — テンポラリ・データベース・グループ、すべてのデータベースとグループ・メンバシップの関係、およびすべてのログインとアプリケーションのバインドを表示します。
- **gr** — **default** テンポラリ・データベース・グループを表示します。**sp_tempdb show** は、**name** オプションに “**default**” を指定したかどうかに関係なく、**default** テンポラリ・データベース・グループにバインドされたすべてのテンポラリ・データベースを表示します。
- **db** — すべてのデータベースおよびテンポラリ・データベースとグループ・メンバシップの関係を表示します。**name** を指定した場合、データベース **name** のデータベースとグループ・メンバシップの関係だけが表示されます。
- **login** — **NULL** 以外のログインのすべてのバインドを表示します。**name** を指定した場合、ログイン **name** のバインドだけが表示されます。
- **app** — **NULL** 以外のアプリケーションのすべてのバインドを表示します。**name** を指定した場合、アプリケーション **name** のバインドが表示されます。

注意 tempdb は、常に default データベース・グループに属しています。

who

指定されたテンポラリ・データベースに割り当てられたすべてのアクティブなセッションを表示します。**who** パラメータを使用する場合は、次の変数を指定してください。

- **dbname** — テンポラリ・データベースの名前です。**dbname** にテンポラリではないデータベースを指定した場合、**sp_tempdb who** は動作しますが、そのデータベースにバインドされたアクティブなセッションは表示されません。

system_view が **cluster** に設定されている場合、クラスタのすべてのアクティブなセッションが検査されます。**system_view** が **instance** に設定されている場合、現在のインスタンスでアクティブなセッションが検査されます。

このコマンドは、クラスタのどのインスタンスからも実行できます。

help

使用方法を表示します。コマンドを指定しないで **sp_tempdb** を実行すると、**sp_tempdb "help"** を実行した場合と同じように動作します。

例

例 1 **mytempdb1** を **default** グループに追加します。

```
sp_tempdb "add", "mytempdb1", "default"
```

例 2 **mytempdb1** を **default** グループから削除します。

```
sp_tempdb "remove", "mytempdb1", "default"
```

例 3 ログイン “sa” を **default** グループにバインドします。

```
sp_tempdb "bind", "lg", "sa", "GR", "default"
```

この例では、**objtype** の値は **login_name** です。**login_name** の代わりに **lg** や **LG** を指定できます。

この例では、**bindtype** の値は **group** です。**group** の代わりに **gr** や **GR** を指定できます。

例 4 ログイン “sa” のバインド先を **default** グループから **mytempdb1** に変更します。

```
sp_tempdb "bind", "lg", "sa", "DB", "mytempdb1"
```

この例では、**bindtype** の値は **database** です。**database** の代わりに **db** や **DB** を指定できます。

例 5 **isql** を **mytempdb1** にバインドします。

```
sp_tempdb "bind", "ap", "isql", "DB", "mytempdb1"
```

この例では、objtype の値は application_name です。application_name の代わりに ap や AP を指定できます。

例 6 isql のバインド先を mytempdb1 から default グループに変更します。

```
sp_tempdb "bind", "ap", "isql", "GR", "default"
```

例 7 ログイン “sa” とアプリケーション “isql” のバインドを削除します。

```
sp_tempdb "unbind", "lg", "sa"
sp_tempdb "unbind", "ap", "isql"
```

例 8 mytempdb1 データベースから、すべてのログインとアプリケーションをバインド解除します。

```
sp_tempdb "unbindall_db", "mytempdb1"
```

例 9 ここからは、sp_temp show コマンドの使用例を示します。さまざまなパラメータの指定方法とその出力を簡単に示します。

```
sp_tempdb show

Temporary Database Groups
-----
default

Database                                     GroupName
-----
tempdb                                       default
mytempdb                                    default
mytempdb1                                   default
mytempdb2                                   default
mytempdb3                                   default

Login    Application  Group    Database    Hardness
-----
NULL     isql         default  NULL        SOFT
sa       NULL         NULL     mytempdb3   HARD
```

例 10 テンポラリ・データベース・グループを表示します。

```
sp_tempdb show, "gr"

Temporary Database Groups
-----
default
```

例 11 default グループにバインドされているすべてのテンポラリ・データベース・グループ名を表示します。

```

sp_tempdb show, "gr", "default"

Member Databases
-----
tempdb
mytempdb
mytempdb1
mytempdb2
mytempdb3

```

例 12 すべてのデータベースとグループ・メンバシップの関係を表示します。

```

sp_tempdb show, "db"

Database                Group
-----
tempdb                  default
mytempdb                default
mytempdb1               default
mytempdb2               default
mytempdb3               default

```

例 13 mytempdb1 データベースのすべてのデータベースとグループ・メンバシップの関係を表示します。

```

sp_tempdb show, "db", "mytempdb1"

Database                Group
-----
mytempdb1               default

```

例 14 NULL 以外のログインのすべてのバインドを表示します。

```

sp_tempdb show, "login"

Login  Application  Group  Database  Hardness
-----
sa      NULL           NULL   mytempdb3  HARD

```

例 15 システムの tempdb に割り当てられているすべてのアクティブなセッションを表示します。

```

sp_tempdb who, "tempdb"

spid  loginame
-----
2      NULL

```

```

3      NULL
4      NULL
5      NULL
6      NULL
7      NULL
8      NULL

```

例 16 ユーザが作成した mytempdb3 テンポラリ・データベースに割り当てられたすべてのアクティブなセッションを表示します。

```

sp_tempdb who, "mytempdb3"

spid  loginame
-----
17    sa

```

例 17 使用方法を表示します。

```
sp_tempdb help
```

Usage:

```

sp_tempdb 'help'
sp_tempdb 'create', <groupname>
sp_tempdb 'drop', <groupname>
sp_tempdb 'add', <tempdbname>, <groupname>
sp_tempdb 'remove', <tempdbname>, <groupname>
sp_tempdb 'bind', <objtype>, <objname>, <bindtype>, <bindobj>, <scope>,
<hardness>
sp_tempdb 'unbind', <objtype>, <objname>, <scope>
sp_tempdb 'unbindall_db', <tempdbname>
sp_tempdb 'show', <command>, <name>
sp_tempdb 'who', <dbname>

```

```
<objtype> = ['LG' ('login_name') | 'AP' ('application_name')];
```

```
<bindtype> = ['GR' ('group') | 'DB' ('database')]
```

```
<hardness> = ['hard' | 'soft']
```

```
<command> = ['all' | 'gr' | 'db' | 'login' | 'app']
```

例 18 すべてのテンポラリ・データベースと、テンポラリ・データベースが属するグループの名前を表示します。

```

create temporary database mytempdb
-----
CREATE DATABASE: allocating 1536 logical pages (3.0
megabytes) on disk 'master'.|

create temporary database mytempdb1

```

```

-----
CREATE DATABASE : allocating 1536 logical pages (3.0
megabytes) on disk 'master'.

sp_tempdb 'add', mytempdb, 'default'
-----
(return status = 0

sp_tempdb show, db
-----
Database Group
-----
tempdb default
mytempdb default
mytempdb1
(3 rows affected)
(return status = 0)

```

例 19 指定されたテンポラリ・データベースに割り当てられたすべてのアクティブ・セッションのログイン名とアプリケーション名を表示します。

```

sp_addlogin anunay, anunay
-----
sp_tempdb "bind", lg, sa, DB, mytempdb3
-----
(return status = 0)

sp_tempdb "bind", lg, anunay, DB, mytempdb3
-----
(return status = 0)

starting sessions
-----
${ISQL} -J -U anunay -P anunay -I${SYBASE}/interfaces -w200

sp_tempdb who, mytempdb3
-----
spid loginame application
-----
11 sa isql
13 anunay isql

(2 rows affected)
(return status = 0)

```

使用法

show および who

テンポラリ・データベース全体のユーザ分散を表示するには、**show** オプションと **who** オプションの両方を使用します。

- すべてのテンポラリ・データベースの名前を取得するには、次を実行します。

• `sp_tempdb 'show'`

- 各テンポラリ・データベース名を渡します。

• `sp_tempdb 'who', tempdbname`

Adaptive Server バージョン 15.0 以降では、**sp_who** を実行して同じ出力を取得できます。

create および drop

sp_tempdb create ストアド・プロシージャを使用する場合、*groupname* 変数は次の条件を満たす必要があります。

- 有効な識別子であること
- 既存のものではないこと

default グループはシステムによって生成されたグループであり、**tempdb** は常にこのグループに属しています。以下の操作を実行していれば、この **default** グループは生成されています。

- この機能を含む Adaptive Server を使用したアップグレード、または
- 新しいマスタ・デバイスの作成

default グループが存在しない場合、次のコマンドを実行して作成できます。

```
sp_tempdb create, "default"
```

既存の **default** グループを作成しようとすると、エラー・メッセージが表示されます。

add および remove

テンポラリ・データベース・グループにテンポラリ・データベースを追加するには、テンポラリ・データベースとそのグループ名があらかじめ存在している必要があります。sp_tempdb add を使用して、default テンポラリ・データベース・グループに属するデータベースの集合に tempdbname を追加すると、tempdbname はそのグループ内からラウンドロビン方式で割り当てられるようになります。

注意 tempdbname が、Adaptive Server の使用可能なデータベースのグローバル・リストに含まれていない場合、sp_tempdb add は失敗します。

ユーザが作成したテンポラリ・データベースは、テンポラリ・データベース・グループに属している必要はありません。システムの tempdb は、暗黙で default グループに属しています。

テンポラリ・データベース・グループにすでに属しているテンポラリ・データベースをグループに追加しようとすると、エラー・メッセージが表示され、sysattributes の内容は何も変更されません。

パーミッション

sp_tempdb のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、manage server 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sa_role が付与されたユーザであることが必要。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> Roles – 現在のアクティブな役割 Keywords or options – NULL Previous value – NULL Current value – NULL Other information – すべての入力パラメータ Proxy information – set proxy が有効な場合は元のログイン名

sp_tempdb_markdrop

説明	(クラスタ環境の場合) ローカル・システム・テンポラリ・データベースを削除ステータスにします。
構文	<code>sp_tempdb_markdrop database_name [, {'mark' 'unmark'}]</code>
パラメータ	<i>database_name</i> 削除するローカル・システム・テンポラリ・データベースの名前です。 mark 指定されたデータベースに削除のマークを付けます。 unmark データベースのマークをクリアします。
例	例 1 削除されるローカル・システム・テンポラリ・データベース“old_cluster_tempdb1”にマークを付けます。 <pre>sp_tempdb_markdrop 'old_cluster_tempdb1', 'mark'</pre> 例 2 ローカル・システム・テンポラリ・データベース“old_cluster_tempdb1”からマークを削除します。 <pre>sp_tempdb_markdrop 'old_cluster_tempdb1', 'unmark'</pre>
使用法	最後のローカル・テンポラリ・データベースを削除するには、次の手順に従います。 <ol style="list-style-type: none">1 <code>sp_tempdb_markdrop</code> を使用して、ローカル・システム・テンポラリ・データベースを削除ステータスにします。2 最後のローカル・テンポラリ・データベースを所有するインスタンスを停止して再起動します。<hr/>注意 削除するローカル・システム・テンポラリ・データベースにマークを付けた後、他のアクティブなインスタンスがない場合は、所有者インスタンスが再起動されます。このインスタンスは、起動時にマークの付いているローカル・システム・テンポラリ・データベースを使用しません。<hr/>3 <code>drop database</code> を使用して、最後のローカル・システム・テンポラリ・データベースを削除します。

パーミッション	sp_tempdb_markup のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、指定したデータベースに対する own database 権限または manage cluster 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、データベース所有者か sa_role が付与されたユーザであることが必要。

sp_thresholdaction

説明 ログ・セグメント上の空きページの数ラストチャンス・スレッシュホルドを下回る場合、このスレッシュホルドが別のプロシージャと対応していないかぎり、自動的に実行されます。Sybaseはこのプロシージャを提供していません。

構文 スレッシュホルドを下回ると、Adaptive Server は次のパラメータをスレッシュホルド・プロシージャに渡します。

```
sp_thresholdaction @dbname,  
                  @segment_name,  
                  @space_left,  
                  @status
```

パラメータ

@dbname

スレッシュホルドに達したデータベースの名前です。

@segment_name

スレッシュホルドに達したセグメントの名前です。

@space_left

論理ページ単位のスレッシュホルドのサイズです。

@status

ラストチャンス・スレッシュホルドには1を指定し、その他のスレッシュホルドには0を指定します。

例 ラストチャンス・スレッシュホルドに対して、トランザクション・ログをテープ・デバイスにダンプするスレッシュホルド・プロシージャを作成します。

```
create procedure sp_thresholdaction  
    @dbname varchar(30),  
    @segmentname varchar(30),  
    @space_left int,  
    @status int  
as  
    dump transaction @dbname to tapedump1
```

使用法

- sp_thresholdaction の作成は、(ユーザ・データベースでは)データベース所有者、(sybsystemprocs データベースでは)システム管理者、または create procedure パーミッションを付与されているユーザが行う必要があります。
- スレッシュホルドの追加やスレッシュホルド・プロシージャの作成は、データベース内のすべてのセグメントで実行できます。

- ラストチャンス・スレッシュホールドを下回ると、Adaptive Server はスレッシュホールド・イベントが発生したデータベース内で、`sp_thresholdaction` プロシージャを検索します。このプロシージャがデータベース内に存在しなければ、Adaptive Server は、`sybsystemprocs` 内を検索します。`sybsystemprocs` にも存在しない場合は `master` が検索されません。Adaptive Server は、プロシージャを検索できなければ、エラーログにエラー・メッセージを送信します。
- `sp_thresholdaction` には、トランザクション・ログをトランケートする `dump transaction` コマンドを組み込むようにしてください。
- ラストチャンス・スレッシュホールドを使用すると、十分な空き領域に `dump transaction` コマンドを記録できます。データベースには、追加のユーザ・トランザクションを記録する領域が十分でなくなる場合があります。その場合は、トランザクション・ログに記録されないコマンド (`select`、高速 `bcp`、`readtext`、`writetext`) と、さらに追加のログ領域を解放するために必要なコマンド (`dump transaction`、`dump database`、`alter database`) だけを実行できます。デフォルトでは、他のコマンドの実行は中断され、メッセージがエラー・ログに送信されます。コマンドを中断ではなくアポートするには `sp_dboption` の `abort tran on log full` オプションを使用してから、`checkpoint` コマンドを実行してください。

中断されたプロセスの再起動

- `dump transaction` コマンドが十分なログ領域を解放したら、中断されたプロセスは自動的に再開され、終了します。
- 最後のバックアップ以降、高速 `bcp`、`writetext`、または `select into` を実行しても、データベースへの変更がログに記録されなかった場合、ラストチャンス・スレッシュホールド・プロシージャは `dump transaction` コマンドを実行できません。この場合は、`dump database` コマンドでデータベースをコピーし、`dump transaction` コマンドでトランザクション・ログをトランケートします。
- この方法で中断したプロセスを再起動するための十分な領域が解放されない場合は、トランザクション・ログのサイズを大きくする必要があります。`alter database` コマンドの `log on` オプションを使用して、追加のログ領域を割り付けてください。
- 最後の手段として、システム管理者が `sp_who` を実行し、どのプロセスが中断されたかを確認してから、`kill` コマンドを使用して、中断したプロセスを強制終了することができます。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [create procedure](#)、[dump transaction](#)

システム・プロシージャ [sp_addthreshold](#)、[sp_dboption](#)、[sp_droptreshold](#)、[sp_helpsegment](#)、[sp_helpthreshold](#)、[sp_modifythreshold](#)、[sp_who](#)

sp_tran_dumpable_status

説明 データベース上にトランザクション・ダンプを作成できない場合、`sp_tran_dumpable_status` はダンプができない理由を表示する。

構文 `sp_tran_dumpable_status [database_name]`

パラメータ `database_name`
調査の対象となるデータベースの名前です。

例 現在、`sybsystemprocs` でトランザクションをダンプできない理由を示します。

```

sp_tran_dumpable_status sybsystemprocs
bit                description
-----
                2          Log is not on its own device
                8          Trunc log on ckpt is set
               32          Dump tran with truncate_only
               64          Database is new or upgraded

```

使用法 このシステム・プロシージャは、単に `tran_dumpable_status` 組み込み関数を読み出します。

パーミッション `tran_dumpable_status` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

このプロシージャは、すべてのユーザが実行できます。

sp_transactions

説明 アクティブなトランザクションについての情報をレポートします。

構文 sp_transactions ["xid", *xid_value*] |
 ["state", {"heuristic_commit" | "heuristic_abort"
 | "prepared" | "indoubt"} [, "xactname"]] |
 ["gtrid", *gtrid_value*]

パラメータ *xid_value*
 master.dbo.systransactions の xactname カラムに格納されているトランザクション名です。

gtrid_value
 Adaptive Server によって調整されたトランザクションのグローバル・トランザクション ID 名です。

例 **例 1** すべてのアクティブなトランザクションについての一般的な情報を表示します。

```
sp_transactions
```

xactkey	connection	dbid	spid	loid	failover	coordinator	srvname	namelen	starttime	state
0x00000b1700040000dd6821390001	Attached	1	1	2	Resident	Local	None	17	Jun 1 1999 3:47PM	Begun
\$user_transaction										
0x00000b1700040000dd6821390001	NA	0	8	0	Resident	Remote	ASTC	108	Jun 1 1999 3:47PM	Begun

```
00000b1700040000dd6821390001-aa01f04ebb9a-00000b1700040000dd6821390001-aa01f04ebb9a-caserv1-caserv1-0002
```

例 2 指定したトランザクションの詳細な情報を表示します。

```
sp_transactions "xid",  
"00000b1700040000dd6821390001-aa01f04ebb9a-00000b1700040000dd6821390001-aa01f04ebb9a-caserv1-caserv1-0002"
```

xactkey	connection	dbid	spid	loid	failover	coordinator	srvname	namelen	starttime	state
0x00000b2500080000dd6821960001	Attached	1	8	139	Resident	External	ASTC	108	Jun 1 1999 3:47PM	Begun

```
00000b1700040000dd6821390001-aa01f04ebb9a-00000b1700040000dd6821390001-aa0
```



```
1f04ebb9a-caserv1-caserv1-0002
```

```
caserv1          caserv1
00000b1700040000dd6821390001-aa01f04ebb9a
```

例 3 「準備」ステータスのトランザクションに関する一般的な情報を表示します。

```
sp_transactions "state", "prepared"
```

例 4 「準備」ステータスのトランザクションのトランザクション名だけを表示します。

```
sp_transactions "state", "prepared", "xactname"
```

例 5 指定したグローバル・トランザクション ID を割り当てられているトランザクションに関するステータス情報を表示します。

```
sp_transactions "gtrid", "00000b1700040000dd6821390001-aa01f04ebb9a"
```

```
xactkey          type      coordinator starttime      state
connection dbid spid loid failover srvname namelen  xactname
commit_node
parent_node
-----
-----
-----
0x00000b1700040000dd6821390001 Local   None      Jun 1 1999 3:47PM Begun
Attached        1      1      2 Resident Tx      NULL      17 $user_transaction
caserv1
caserv1
```

使用法

- `sp_transactions` は、`systransactions` テーブルのデータを変換して、アクティブなトランザクションに関する情報を表示します。`systransactions` には、アクティブなトランザクションに関するメモリ内情報の他に、`syscoordinations` テーブル内のデータも含まれます。
- キーワードを付けずに `sp_transactions` を使用すると、すべてのアクティブ・トランザクションに関する情報が表示されます。
- `xid` キーワードを付けて `sp_transactions` を使用すると、指定したトランザクションの `gtrid`、`commit_node`、および `parent_node` カラムだけが表示されます。
- `state` キーワードを付けて `sp_transactions` を使用すると、指定した状態のアクティブ・トランザクションに関する情報だけが表示されます。

xid および xactname を付けて `sp_transactions` を使用すると、指定した状態のトランザクションのトランザクション名だけが表示されます。

- `gtrid` キーワードを付けて `sp_transactions` を使用すると、指定したグローバル・トランザクション ID を持つトランザクションに関する情報だけが表示されます。
- `sp_transactions` は、XA-Library および XA-Server 製品の `sp_xa_scan_xact` プロシージャに置き換わるものです。
- 詳細については、『Adaptive Server 分散トランザクション管理機能の使用』を参照してください。

`sp_transactions` 出力のカラムの説明

- `xactkey` カラムには、Adaptive Server がトランザクションを識別するために使用する内部トランザクション・キーが表示されます。
- `type` カラムには、トランザクションのタイプが表示されます。
 - “Local” とは、トランザクションが `begin transaction` 文によって、ローカル Adaptive Server 上で明示的に開始されたことを示します。
 - “Remote” とは、リモート Adaptive Server 上で実行されるトランザクションを示します。
 - “External” とは、トランザクションに対応する外部コーディネータがあることを示します。たとえば、リモート Adaptive Server、MSDTC、または X/Open XA トランザクション・マネージャによって調整されるトランザクションには、“External” のフラグが付けられます。
 - “Dtx_State” は、Adaptive Server がコーディネーションする分散トランザクションの特殊な状態を示します。これは、ローカル・サーバ上のトランザクションがコミットまたはアポートされたが、Adaptive Server が、リモート・パティシパント上のそのトランザクションの分岐を解決できないことを示します。この事態は、Adaptive Server がコーディネーションするサーバとの接触を失った場合に発生することがあります。
- `coordinator` カラムは、分散トランザクションの管理に使用されるメソッドまたはプロトコルを示します。

sp_transactions “coordinator” の値	意味
None	トランザクションが分散トランザクションではなく、調整プロトコルを必要としない。
ASTC	トランザクションが、Adaptive Server のトランザクション調整サービスによって調整される。
XA	トランザクションが、Adaptive Server XA-Library インタフェースを介して、X/Open XA に準拠するトランザクション・マネージャによって調整される。このようなトランザクション・マネージャには、Encina、CICS、および Tuxedo がある。
DTC	トランザクションが MSDTC によって調整される。
SYB2PC	トランザクションが Sybase の 2 フェーズ・コミット・プロトコルによって調整される。

- **starttime** カラムには、トランザクションの開始時刻が示されます。
- **state** カラムには、sp_transactions を実行したときのトランザクションのステータスが示されます。

sp_transactions “state” の値	意味
Begun	トランザクションが開始したが、更新が実行されていない。
Done Command	トランザクションが更新コマンドを完了した。
Done	X/Open XA トランザクションがデータの変更を終了した。
Prepared	トランザクションが正常に準備された。
In Command	トランザクションが現在データを変更している。
In Abort Cmd	トランザクションでの現在のコマンドの実行がアボートされた。
Committed	トランザクションが正常にコミットされて、コミット・ログ・レコードが書き込まれた。
In Post Commit	トランザクションが正常にコミットされたが、現在トランザクション・リソースの割り付けを解除している。
In Abort Tran	トランザクションがアボートされている。この事態は、明示的コマンドの結果、またはシステム障害が原因で発生する可能性がある。
In Abort Savept	トランザクションがセーブポイントまでロールバックされている。
Begun-Detached	トランザクションが開始したが、現在付加されているスレッドがない。
Done Cmd-Detached	トランザクションがデータの修正を終了したが、現在付加されているスレッドがない。
Done-Detached	トランザクションがこれ以上データを修正せず、現在付加されているスレッドがない。
Prepared-Detached	トランザクションが正常に準備されて、現在付加されているスレッドがない。
Heur Committed	トランザクションが、dbcc complete_xact コマンドを使用してヒューリスティックにコミットされた。
Heur Rolledback	トランザクションが、dbcc complete_xact コマンドを使用して、ヒューリスティックにロールバックされた。

- **connection** カラムは、トランザクションがスレッドと現在対応しているかどうかを示します。
 - “Attached” は、トランザクションに対応する制御スレッドがあることを示します。
 - “Detached” は、現在トランザクションに対応するスレッドがないことを示します。CICS や TUXEDO など、一部の外部トランザクション・マネージャは、X/Open XA の “suspend” および “join” セマンティックを使用して、異なる複数のスレッドを 1 つのトランザクションに対応付けます。
- **dbid** カラムは、トランザクションが開始されたデータベースのデータベース ID を示します。
- **spid** カラムは、トランザクションに対応するサーバ・プロセス ID を示します。トランザクションが “Detached” の場合、“spid” 値は 0 です。
- **loid** カラムは、master.dbo.systransactions のユニークなロック所有者 ID を示します。
- **failover** カラムは、トランザクションのフェールオーバー・ステータスを示します。
 - “Resident Tx” はトランザクションが開始して、同じサーバ上で実行されていることを示します。“Resident Tx” は、通常の動作状態で、Adaptive Server 高可用性機能を利用しないシステムで表示されます。
 - **Failed-over Tx** は、セカンダリ・コンパニオン・サーバへのフェールオーバーが発生した後に表示されます。**Failed-over Tx** は、トランザクションが最初はプライマリ・サーバで開始され、準備ステータスに達したが、プライマリ・サーバでのシステム障害などのために、自動的にセカンダリ・コンパニオン・サーバにマイグレートされたことを意味します。準備済みトランザクションのマイグレーションは、外部コーディネーティング・サービスに対して透過的に発生します。
 - “Tx by Failover-Conn” は、指定サーバ上でトランザクションを開始しようとしたが、代わりにセカンダリ・コンパニオン・サーバで開始されたことを示します。この事態は、元のサーバにフェールオーバー条件がある場合に発生します。

- `srvname` カラムは、トランザクションが実行されているリモート・サーバの名前を示します。このカラムは、リモート・トランザクションでのみ有効です。ローカルおよび外部トランザクションの場合、`srvname` は `null` になります。
- `namelen` カラムは、`xactname` 値の合計長を示します。
- `xactname` はトランザクション名です。ローカル・トランザクションでは、トランザクション名は `begin transaction` コマンドの一部として定義できます。外部トランザクション・マネージャは、ユニークなトランザクション名をさまざまなフォーマットで指定します。たとえば、`X/Open XA` に準拠するトランザクション・マネージャは、`xactname` に格納されたグローバル・トランザクション識別子と分岐識別子からなるトランザクション ID (`xid`) を指定します。
- Adaptive Server によってコーディネートされたトランザクションについては、`gtrid` カラムに、グローバル・トランザクション ID が表示されます。同じ分散トランザクションの一部であるトランザクション分岐は、同じ `gtrid` を共有します。特定の `gtrid` を `sp_transactions gtrid` キーワードで使用すると、同じ分散トランザクション内にある別のトランザクション分岐の状態を判別できます。

`sp_transactions` は、外部コーディネータを持つトランザクションの `gtrid` を表示できません。`X/Open XA` に準拠するトランザクション・マネージャ、`MSDTC`、または `SYB2PC` によってコーディネートされるトランザクションの場合、`gtrid` カラムには、外部コーディネータが指定する完全なトランザクション名が表示されます。

- Adaptive Server によって調整されるトランザクションの場合、`commit_node` カラムには、分散トランザクションの最も外側にあるブロックを実行するサーバが表示されます。この最も外側にあるブロックは、すべての従属トランザクションのコミット・ステータスを最終的に決定します。

Adaptive Server によって調整されないトランザクションの場合、`commit_node` には、表 1-34 で説明されている値のいずれかが表示されます。

表 1-34: `commit_node` と `parent_node` の値

値	意味
<code>server_name</code>	コミットまたは親ノードは、指定の <code>server_name</code> を持つ Adaptive Server。
XATM	コミットまたは親ノードは、 <code>X/Open XA</code> に準拠するトランザクション・マネージャ。
MSDTC	コミットまたは親ノードは <code>MSDTC</code> 。
SYB2PCTM	トランザクションは <code>SYB2PC</code> プロトコルを使用して調整される。

- Adaptive Server によって調整されるトランザクションの場合、`parent_node` カラムには、ローカル・サーバ上の外部トランザクションを調整するサーバが示されます。

Adaptive Server によって調整されないトランザクションの場合、`parent_node` には、表 1-34 で説明されている値のいずれかが表示されます。

注意 `commit_node` と `parent_node` の値は、分散トランザクションの階層のレベルによって異なる可能性があります。

パーミッション

`sp_transactions` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> – 現在のアクティブな役割 <i>Keywords or options</i> – NULL <i>Previous value</i> – NULL <i>Current value</i> – NULL <i>Other information</i> – すべての入力パラメータ <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ [sp_lock](#)、[sp_who](#)

sp_unbindcache

説明	データ・キャッシュから、データベース、テーブル、インデックス、text オブジェクト、または image オブジェクトのバインドを解除します。
構文	<pre>sp_unbindcache dbname [, [owner.]tablename [, indexname "text only"]]</pre>
パラメータ	<p>dbname バインドを解除するデータベースの名前、またはバインドを解除するオブジェクトが格納されているデータベースの名前です。</p> <p>owner テーブル所有者の名前です。テーブルがデータベース所有者に所有されている場合、所有者名は省略可能です。</p> <p>tablename キャッシュからバインドを解除するテーブルの名前です。または、キャッシュからバインドを解除するインデックス、text オブジェクト、または image オブジェクトを格納しているテーブルの名前です。</p> <p>indexname キャッシュからバインドを解除するインデックスの名前です。</p> <p>text only キャッシュから text オブジェクトまたは image オブジェクトのバインドを解除します。</p>
例	<p>例 1 titles テーブルがバインドされているキャッシュから、そのバインドを解除します。</p> <pre>sp_unbindcache pubs2, titles</pre> <p>例 2 titleidind インデックスがバインドされているキャッシュから、そのバインドを解除します。</p> <pre>sp_unbindcache pubs2, titles, titleidind</pre> <p>例 3 au_pix テーブルに格納されている text オブジェクトまたは image オブジェクトがバインドされているキャッシュから、そのバインドを解除します。</p> <pre>sp_unbindcache pubs2, au_pix, "text only"</pre> <p>例 4 トランザクション・ログ syslogs がバインドされているキャッシュから、そのバインドを解除します。</p> <pre>sp_unbindcache pubs2, syslogs</pre>

使用法

- キャッシュからデータベースまたはデータベース・オブジェクトのバインドを解除すると、その後のキャッシュの I/O は、すべてデフォルト・データ・キャッシュ内で実行されます。キャッシュ内のバインドを解除されたデータ・ページはすべてディスクに書き込まれ、クリーン・ページはキャッシュからすべてクリアされます。詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。
- `sp_unbindcache` を使用して、使用中のデータベースのバインドを解除しようとする、`Adaptive Server` はエラー番号 857 を返します。
- システムの `tempdb` を除いて、キャッシュのバインド解除はただちに有効になります。したがって、サーバを再起動する必要はありません。
- このバージョンでも、システムの `tempdb` に対して `sp_unbindcache` を実行することはできますが、システムの `tempdb` に関連するバインドは動的には行われなくなりました。サーバを再起動するまでは、次のようになります。
 - 変更が有効にならない。
 - `sp_helpcache` で、保留を表すステータス “P” がレポートされる。ただし、システム `tempdb` をデフォルトのデータ・キャッシュに明示的にバインドした場合は、有効を表すステータス “V” がレポートされる。これは、デフォルトでシステム `tempdb` はデフォルトのデータ・キャッシュにすでにバインドされているため。
- データベース、テーブル、インデックスを削除すると、そのキャッシュのバインドは自動的に削除されます。
- データベースのバインドを解除するには、`master` データベースを使用してください。テーブル、インデックス、`text` オブジェクト、または `image` オブジェクトのバインドを解除するには、そのオブジェクトが格納されているデータベースを使用してください。
- データベース内のシステム・テーブルのバインドを解除するには、そのデータベースを使用してください。また、データベースはシングルユーザ・モードにしてください。次のコマンドを使用します。

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

詳細については [sp_dboption](#) を参照してください。

- 次の各プロシージャでは、それぞれ対応するオブジェクトのバインド情報が出力されます。 `sp_helppdb` ではデータベースのバインド、 `sp_help` ではテーブルのバインド、 `sp_helpindex` ではインデックスのバインドに関する情報が出力されます。
- `sp_helpcache` は、キャッシュにバインドされているオブジェクトの名前を出力します。
- `sp_unbindcache` は、テーブルまたはそのインデックスをキャッシュからバインド解除するときに、排他テーブル・ロックを取得する必要があります。バインドの解除が実行されている間、ページを読み込みことはできません。ユーザがテーブル上でロックを保持しているときに、そのオブジェクト上で `sp_unbindcache` を発行すると、 `sp_unbindcache` タスクはロックが解放されるまでスリープします。
- `sp_bindcache` あるいは `sp_unbindcache` を使用してオブジェクトとキャッシュのバインドを変更すると、そのオブジェクトを参照するストアド・プロシージャは、次に実行されたときに再コンパイルされます。データベースに対応するバインドを変更すると、データベース内のオブジェクトを参照するストアド・プロシージャが、次に実行されるときに再コンパイルされます。
- キャッシュからすべてのオブジェクトをバインド解除するには、 `sp_unbindcache_all` システム・プロシージャを使用します。

パーミッション

`sp_unbindcache` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage data cache</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査

`sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照

システム・プロシージャ `sp_bindcache`、`sp_dboption`、`sp_help`、`sp_helppdb`、`sp_helpcache`、`sp_helpindex`、`sp_unbindcache_all`

sp_unbindcache_all

説明	キャッシュにバインドされているすべてのオブジェクトのバインドを解除します。
構文	<code>sp_unbindcache_all cache_name</code>
パラメータ	<code>cache_name</code> オブジェクトのバインドを解除するデータ・キャッシュの名前です。
例	<code>pub_cache</code> にバインドされているすべてのデータベース、テーブル、インデックス、 <code>text</code> オブジェクト、 <code>image</code> オブジェクトのバインドを解除します。 <code>sp_unbindcache_all pub_cache</code>
使用法	<ul style="list-style-type: none"> • キャッシュからエンティティのバインドを解除すると、キャッシュに対応する後続の I/O はすべてデフォルト・キャッシュ内で実行されます。 • キャッシュからオブジェクトのバインドを個別に解除するには、sp_unbindcache システム・プロシージャを使用します。 • システムの <code>tempdb</code> が <code>pub_cache</code> にバインドされている場合は、<code>sp_unbindcache_all</code> を使用できません。使用した場合はエラー・メッセージが表示され、<code>sp_unbindcache_all</code> はどのオブジェクトのバインドも解除しません。 <ul style="list-style-type: none"> • システムの <code>tempdb</code> のバインドを解除するには、先に <code>sp_unbindcache</code> を使用してください。 • キャッシュのバインドを解除する方法の詳細については、sp_unbindcache を参照してください。
パーミッション	<code>sp_unbindcache_all</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage data cache</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_bindcache](#)、[sp_helpcache](#)、[sp_unbindcache](#)

sp_unbindefault

説明	作成済みのデフォルト値を、カラムまたはユーザ定義データ型からバインド解除します。
構文	<code>sp_unbindefault objname [, futureonly]</code>
パラメータ	<p>objname デフォルト値のバインドを解除するテーブル／カラムまたはユーザ定義データ型の名前です。このパラメータが <code>table.column</code> の形式になっていない場合は、ユーザ定義データ型であると判断されます。ユーザ定義データ型からデフォルト値のバインドを解除すると、ユーザ定義データ型と同じデフォルト値を割り当てられているデータ型のカラムのバインドもすべて解除されます。デフォルト値がすでに変更されているカラムは変更されません。</p> <p>futureonly 指定したユーザ定義データ型の既存カラムのデフォルト値が失なわれないようにします。カラムからデフォルト値のバインドを解除する場合は無視されます。</p>
例	<p>例 1 <code>employees</code> テーブルの <code>startdate</code> カラムから、デフォルト値のバインドを解除します。</p> <pre>sp_unbindefault "employees.startdate"</pre> <p>例 2 ユーザ定義データ型である <code>ssn</code> とそのデータ型のすべてのカラムから、デフォルト値のバインドを解除します。</p> <pre>sp_unbindefault ssn</pre> <p>例 3 <code>ssn</code> ユーザ定義データ型からデフォルト値のバインドを解除します。この型の既存のカラムへの影響はありません。</p> <pre>sp_unbindefault ssn, futureonly</pre>
使用法	<ul style="list-style-type: none">• <code>sp_bindefault</code> によって作成されたデフォルトを削除するには、<code>sp_unbindefault</code> を使用します。<code>create table</code> 文または <code>alter table</code> 文によって宣言されたデフォルト値を削除するには、<code>alter table</code> を使用します。• ユーザ定義データ型のカラムは現在のデフォルト値を失います。ただし、デフォルト値が変更されている場合、またはオプションのもう 1 つのパラメータ値が <code>futureonly</code> である場合を除きます。• デフォルト値のテキストを表示するには、デフォルト値の名前をパラメータとして <code>sp_helptext</code> を実行します。

パーミッション `sp_unbindefault` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効 | 細密なパーミッションが有効の場合、データベース所有者またはユーザ・データ型の所有者であることが必要。

細密なパーミッションが無効 | 細密なパーミッションが無効の場合、オブジェクトの所有者であることが必要。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名
67	<code>unbind</code>	<code>sp_unbindefault</code>	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – NULL • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 [コマンド `create default`、`drop default`](#)
[システム・プロシージャ `sp_bindefault`、`sp_helptext`](#)

sp_unbindexclass

説明	指定したスコープのクライアント・アプリケーション、ログイン、ストアド・プロシージャ、デフォルトの実行クラスにこれまで対応していた実行クラス属性を削除します。
構文	<code>sp_unbindexclass object_name, object_type, scope</code>
パラメータ	<p>object_name 実行クラスとの対応関係を削除するアプリケーション、ログイン、ストアド・プロシージャの名前です。object_typeがDFの場合、object_nameはNULLである必要があります。</p> <p>object_type object_name型を指定します。アプリケーションにはAPを、ログインにはLGを、ストアド・プロシージャにはPRを、デフォルトの実行クラスにはDFを指定します。</p> <p>scope バインドを解除するアプリケーションまたはログインの名前です。ストアド・プロシージャの場合、ストアド・プロシージャの所有者の名前(ユーザ名)を指定します。オブジェクトの型がDFの場合はNULLです。</p>
例	<p>アプリケーション <code>isql</code> にスコープされていた “sa” ログインと実行クラス間の対応関係を解除します。その他のバインド仕様、優先度、およびスコープ・ルールに応じて、“sa” は自動的に他の実行クラスにバインドされます。他に適用できるバインドがない場合、オブジェクトはデフォルトの実行クラス <code>EC2</code> にバインドされます。</p> <pre>sp_unbindexclass 'sa', 'lg', 'isql'</pre>
使用法	<ul style="list-style-type: none">指定するパラメータは、<code>sysattributes</code> システム・テーブル内にある既存のエントリと一致させてください。スコープに <code>null</code> 値を指定したときに、スコープが <code>null</code> 値のオブジェクトがあると、Adaptive Server はこのオブジェクトのバインドを解除します。スコープに <code>null</code> 値を指定しても、バインドされたすべてのオブジェクトのバインドが解除されるわけではありません。ストアド・プロシージャを実行クラスからバインド解除するときは、<code>scope</code> パラメータにストアド・プロシージャの所有者の名前(ユーザ名)を使用してください。

- ストアド・プロシージャをユーザ定義のデフォルトの実行クラスからバインド解除する場合は、ユーザ定義のデフォルトの実行クラス属性で実行しているすべてのタスクが、システム定義のデフォルトの実行クラス EC2 の属性で実行されます。
- バインドの解除の前後に、ストアド・プロシージャを削除することができます。
- データベース内の実行クラスにバインドされているストアド・プロシージャを所有しているユーザは、データベースから削除できません。
- PR 型のオブジェクトは、バインドを解除した後でデータベースから削除します。
- 対応するエンジン・グループにオンラインのエンジンがなく、対応している実行クラスにアクティブなプロセスがバインドされている場合は、バインドの解除を正常に実行できません。
- 優先度とスコープ・ルールに応じて、バインドを解除する実行クラスは、*object_name* に対して有効であったり無効であったりします。その他のバインド仕様、優先度、およびスコープ・ルールに応じて、オブジェクトは自動的に別の実行クラスにバインドされます。他に適用できるバインドがない場合、オブジェクトはデフォルトの実行クラスにバインドされます。ユーザ定義のデフォルトの実行クラスがない場合、オブジェクトは EC2 クラスにバインドされます。

パーミッション

`sp_unbindexclass` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>manage any execution class</code> 権限を持つユーザであることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sa_role</code> が付与されたユーザであることが必要。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addexclass](#)、[sp_bindexclass](#)、[sp_dropexclass](#)、[sp_showexclass](#)

ユーティリティ isql

sp_unbindmsg

説明	制約からユーザ定義メッセージのバインドを解除します。
構文	<code>sp_unbindmsg <i>constrname</i></code>
パラメータ	<i>constrname</i> メッセージのバインドを解除する制約の名前です。
例	<code>positive_balance</code> 制約からユーザ定義メッセージのバインドを解除します。 <code>sp_unbindmsg positive_balance</code>
使用法	<ul style="list-style-type: none"> 制約にバインドできるメッセージ数は 1 つだけです。制約にバインドされているメッセージを変更するには、sp_bindmsg を使用します。新しいメッセージ番号によって、既存のバインド・メッセージが置き換えられます。最初に <code>sp_unbindmsg</code> を使用する必要はありません。 <code>sysusermessages</code> テーブルからメッセージ・テキストを検索するには、sp_getmessage を実行します。
パーミッション	<code>sp_unbindmsg</code> を実行するには、制約の所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なります。
監査	<code>sysaudits</code> テーブルの <code>event</code> カラムと <code>extrainfo</code> カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — すべての入力パラメータ <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名
69	unbind	sp_unbindmsg	<ul style="list-style-type: none"> <i>Roles</i> — 現在のアクティブな役割 <i>Keywords or options</i> — NULL <i>Previous value</i> — NULL <i>Current value</i> — NULL <i>Other information</i> — NULL <i>Proxy information</i> — <code>set proxy</code> が有効な場合は元のログイン名

参照 システム・プロシージャ [sp_addmessage](#)、[sp_bindmsg](#)、[sp_getmessage](#)

sp_unbindrule

説明	カラムまたはユーザ定義データ型からルールของ バインドを解除します。
構文	<code>sp_unbindrule objname [, futureonly [, "accessrule" "all"]]</code>
パラメータ	<p>objname ルールのバインドを解除するテーブル、カラム、およびユーザ定義データ型の名前です。このパラメータが <code>table.column</code> の形式になっていない場合は、ユーザ定義データ型であると判断されます。ユーザ定義データ型からルールのバインドを解除すると、同じデータ型のカラムからもルールのバインドが解除されます。ただし、すでに別のルールにバインドされているカラムには影響がありません。</p> <p>futureonly 指定したユーザ定義データ型の既存カラムのルールが失なわれないようにします。カラムからルールのバインドを解除するときは無視されます。</p> <p>accessrule objname にバインドされたアクセス・ルールのバインドを解除することを指定します。</p> <p>all objname にバインドされたすべてのルールのバインドを解除することを指定します。</p>
例	<p>例 1 <code>employees</code> テーブルの <code>startdate</code> カラムから、ルールのバインドを解除します。</p> <pre>sp_unbindrule "employees.startdate"</pre> <p>例 2 <code>def_ssn</code> という名前のユーザ定義データ型と、その型のすべてのカラムから、ルールのバインドを解除します。</p> <pre>sp_unbindrule def_ssn</pre> <p>例 3 <code>ssn</code> ユーザ定義データ型にはルールが存在しなくなりますが、既存の <code>ssn</code> カラムへの影響はありません。</p> <pre>sp_unbindrule ssn, futureonly</pre> <p>例 4 <code>all</code> パラメータを使用すると、アクセス・ルールとドメイン・ルールの両方をバインド解除できます。たとえば、<code>publishers</code> テーブル上のすべてのアクセス・ルールとドメイン・ルールをバインド解除するには、次を発行します。</p> <pre>sp_unbindrule publishers, null, "all"</pre>

ユーザ定義データ型の以降の使用について、ユーザ定義データ型からアクセス・ルールをバインド解除するには、次のコマンドを発行します。

```
sp_unbindrule def_ssn, futureonly, "accessrule"
```

このデータ型の以降の使用について、アクセス・ルールとドメイン・ルールをバインド解除するには、次のコマンドを発行します。

```
sp_unbindrule def_ssn, futureonly, "all"
```

例 5 次のアクセス・ルールは `publishers` テーブルにバインドされます。

```
sp_bindrule empl_access, "publishers.pub_id"
```

このルールをバインド解除するには、次を発行します。

```
sp_unbindrule "empl_access", NULL, "accessrule"
```

使用法

- `sp_unbindrule` を実行すると、現在のデータベース内にあるカラムまたはユーザ定義データ型からルールが削除されます。既存の `objname` カラムからルールのバインドを解除したくない場合は、2つ目の `futureonly` パラメータを使用します。
- `sp_unbindrule` を使用して、検査制約のバインドを解除することはできません。検査制約を削除するには、`alter table` を使用します。
- テーブルのカラムからルールのバインドを解除するには、“`table.column`” の形式で `objname` 引数を指定します。
- ルールは、ユーザ定義データ型の既存のカラムすべてからバインド解除されます。ただし、ルールが変更されている場合、またはオプションの2つ目のパラメータ `futureonly` が指定されている場合を除きます。
- ルールのテキストを表示するには、ルール名をパラメータとして `sp_helptext` を実行します。

パーミッション

`sp_unbindrule` を実行するには、テーブル所有者またはデータ型の所有者であることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – set proxy が有効な場合は元のログイン名

参照 コマンド [create rule](#)、[drop rule](#)

システム・プロシージャ [sp_bindrule](#)、[sp_helptext](#)

sp_version

説明 インストール・スクリプト (*installmaster*、*installdbccdb* など) の最後の実行時のバージョン情報と、実行が正常に終了したかどうかを返します。

構文 `sp_version [script_file, [all]]`

パラメータ *script_file*
 インストール・スクリプトの名前 (デフォルト値は NULL) です。

all
 インストール・スクリプトについての詳細な情報 (実行された日付や実行に要した時間など) をレポートします。

例 1 実行されたすべてのインストール・スクリプトのスクリプト名、バージョン、およびステータスを返します。

```
sp_version
Script          Version
Status
-----
installmaster  15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu
Sep 23 22:12:12 2004
Complete
installmaster  15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu
Sep 23 22:12:12 2004
Complete
installmodel   15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1861/32-
bit/OPT/Mon Sep 27 23:40:02 2004
Complete
```

例 2 *installmaster* インストール・スクリプトに関する情報を返します。

```
sp_version      installmaster
-----
installmaster  15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu
Sep 23 22:12:12 2004
Complete
```

例 3 実行されたすべてのインストール・スクリプトのスクリプト・ファイル名、日付、時刻、バージョン、およびステータスを返します。

```
sp_version null, 'all'
Script
Version          Status
Start/End Date
```

```

-----
installdbccdb      15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1861/32-
bit/OPT/Mon Sep 27 23:40:02 2004
Complete [Started=Sep 29 2004  4:41PM]-[Completed=Sep 29 2004  4:42PM]
installmaster
15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu Sep 23 22:12:
12 2004
Complete [Started=Sep 29 2004  3:49PM]-[Completed=Sep 29 2004  3:58PM]
installmodel
15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1861/32-bit/OPT/Mon Sep 27 23:
40:02 2004
Complete [Started=Sep 29 2004  4:51PM]-[Completed=Sep 29 2004  4:51PM]

```

例 4 名前が *install%* で始まるすべてのインストール・スクリプトのファイル名、バージョン、インストールのステータスを返します。

```

sp_version 'install%'
Script
Version          Status
-----
installdbccdb
15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1861/32-bit/OPT/Mon Sep 27
23:40:02 2004      Complete
installmaster
15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu Sep 23 22:12:
12 2004            Complete
installmodel
15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1861/32-bit/OPT/Mon Sep 27 23:
40:02 2004        Complete

```

例 5 ワイルド・カード “*install%*” に一致するインストール・スクリプトのすべての詳細情報を返します。

```

sp_version 'install%', 'all'
Script
Version          Status
Start/End Date
-----
installmaster
15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu Sep 23 22:12:
12 2004
Complete [Started=Sep 29 2004  3:49PM]-[Completed=Sep 29 2004  3:58PM]

```

例 6 *installmaster* インストール・スクリプトについてのすべての詳細情報を返します。

```

sp_version 'installmaster', 'all'
Script

```

```

Version          Status
Start/End Date
-----
installmaster
15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu Sep 23 22:12:
12 2004
Complete [Started=Sep 29 2004  3:49PM]-[Completed=Sep 29 2004  3:58PM]

```

使用法 `sp_version` を使用して、Adaptive Server にインストールされているスクリプト (`installmaster`、`installdbccdb` など) の現在のバージョン、それらのスクリプトが正常に実行されたかどうか、および、実行に要した時間を確認できます。

パーミッション `sp_version` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

sp_volchanged

説明	ダンプまたはロード中に、要求されたボリューム処理をオペレータが実行したことを Backup Server に通知します。
構文	<code>sp_volchanged session_id, devname, action [, fname [, vname]]</code>
パラメータ	<p>session_id ボリュームの変更を要求した Backup Server セッションを示します。Backup Server のボリュームの変更要求で指定された <code>@session_id</code> パラメータを使用します。</p> <p>devname 新しいボリュームがマウントされるデバイスの名前です。Backup Server のボリューム変更要求で指定された <code>@devname</code> パラメータを使用します。Backup Server が Adaptive Server と同じマシン上に存在しない場合は、次の構文を使用します。</p> <pre>device at backup_server_name</pre> <p>action Backup Server がダンプまたはロードをアボートする (<code>abort</code>) のか、続行する (<code>proceed</code>) のか、または再試行する (<code>retry</code>) のかを示します。</p> <p>fname ロードするファイル名です。sp_volchanged によってファイル名を指定しなければ、Backup Server は、ロード用コマンドの <code>file = filename</code> パラメータをロードします。sp_volchanged とロード・コマンドのどちらにもロードするファイルが指定されていない場合は、テープの先頭のファイルがロードされます。</p>

vname

ANSI テープ・ラベルに表示されるボリューム名です。Backup Server がボリューム名を ANSI テープ・ラベルに書き込むのは、既存のダンプを上書きする場合、メーカー品の新しいテープにダンプする場合、または内容を識別できないテープにダンプを行う場合です。sp_volchanged で vname が指定されていない場合、Backup Server は、ダンプ用コマンド内で指定されている dumpvolume 値を使用します。ボリューム名を sp_volchanged でも、ダンプ用コマンドでも指定しなければ、Backup Server は ANSI テープ・ラベルの名前フィールドをブランクのままにします。

ロードを実行するとき、Backup Server は vname を使用して、正しいテープがマウントされていることを確認します。sp_volchanged で vname が指定されなければ、Backup Server は、ロード用コマンドで指定されている dumpvolume を使用します。ボリューム名を sp_volchanged でも、ロード用コマンドでも指定しなければ、Backup Server は、ダンプをロードする前に ANSI テープ・ラベルの名前フィールドを確認しません。

例

オペレータが、テープを変更した後、次のコマンドを実行します。

```
sp_volchanged 8, "/dev/nrmt4", RETRY
```

Backup Server からの次のメッセージは、マウントされているテープの有効期限がまだ切れていないことを示します。

```
Backup Server は次の作業を行います。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 は次の作業を行います。4.78.1.1:EXECUTE sp_volchanged
```

```
@session_id = 8,
@devname = '/auto/remote/pubs3/SERV/Masters/testdump',
@action = { 'PROCEED' | 'RETRY' | 'ABORT' }
```

使用法

- Backup Server は、現在マウントされているボリュームの問題を検出すると、次のようにボリューム変更を要求します。
 - (UNIX システムの場合) Backup Server は、ダンプまたはロード要求を開始したクライアントにメッセージを送信します。ダンプ用またはロード用コマンドの with notify = operator_console オプションを使用して、Backup Server が開始された端末にメッセージを送信します。

- 他のボリュームをマウントした後、オペレータは、ダンプまたはロードを実行している Backup Server と通信できる Adaptive Server から sp_volchanged を実行します。オペレータは、ダンプまたはロードの開始元である Adaptive Server にログインする必要はありません。
- (UNIX システムの場合) テープ容量がいっぱいになると、Backup Server がボリュームの変更を要求します。オペレータは、別のテープをマウントし、sp_volchanged を実行します。表 1-35 にこの手順例を示します。

表 1-35: UNIX システムでのテープ・ボリュームの変更

手順	オペレータ側 (isql を使用)	Adaptive Server	Backup Server
1	<ul style="list-style-type: none"> • dump database コマンドを発行する。 		
2		<ul style="list-style-type: none"> • Backup Server にダンプ要求を送信する。 	
3			<ul style="list-style-type: none"> • Adaptive Server からダンプ要求メッセージを受信する。 • テープ・マウントのメッセージをオペレータに送信する。 • オペレータからの応答を待つ。
4	<ul style="list-style-type: none"> • Backup Server からボリューム変更要求を受信する。 • テープをマウントする。 • sp_volchanged を実行する。 		
5			<ul style="list-style-type: none"> • テープを調べる。 • テープが使用可能であれば、ダンプを開始する。 • テープに空き領域がない場合は、ボリューム交換要求をオペレータに送信する。
6	<ul style="list-style-type: none"> • Backup Server からボリューム変更要求を受信する。 • テープをマウントする。 • sp_volchanged を実行する。 		
7			<ul style="list-style-type: none"> • ダンプを続行する。 • ダンプが完了したら、オペレータと Adaptive Server にメッセージを送信する。

手順	オペレータ側 (isql を使用)	Adaptive Server	Backup Server
8	<ul style="list-style-type: none"> • ダンプ完了のメッセージを受信する。 • テープを取り出してラベルを付ける。 	<ul style="list-style-type: none"> • ダンプ完了のメッセージを受信する。 • ロックを解除する。 • <code>dump database</code> コマンドを完了する。 	

パーミッション `sp_volchanged` は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査 `sysaudits` テーブルの `event` カラムと `extrainfo` カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	<code>exec_procedure</code>	プロシージャの実行	<ul style="list-style-type: none"> • <i>Roles</i> – 現在のアクティブな役割 • <i>Keywords or options</i> – NULL • <i>Previous value</i> – NULL • <i>Current value</i> – NULL • <i>Other information</i> – すべての入力パラメータ • <i>Proxy information</i> – <code>set proxy</code> が有効な場合は元のログイン名

参照 コマンド `dump database`、`dump transaction`、`load database`、`load transaction`
ユーティリティ `isql`

sp_webservices

説明 Adaptive Server Web Services Engine で使用されるプロキシ・テーブルの作成と管理を行います。

構文 プロキシ・テーブルを作成する場合：

```
sp_webservices 'add', 'wsdl_uri' [, sds_name]
                [, 'method_name=proxy_table
                [,method_name=proxy_table '* ' ]
```

sp_webservices の使用方法を表示する場合：

```
sp_webservices help [, 'option']
```

WSDL ファイルにマップされるプロキシ・テーブルをリストする場合：

```
sp_webservices 'list' [, 'wsdl_uri'] [, sds_name]
```

タイムアウト設定を変更する場合：

```
sp_webservices 'modify', 'wsdl_uri', 'timeout=time'
```

WSDL ファイルにマップされるプロキシ・テーブルを削除する場合：

```
sp_webservices 'remove', 'wsdl_uri' [, sds_name]
```

ユーザ定義 Web サービスのオプション

ユーザ定義 Web サービスのデータベース・エイリアスを作成する場合：

```
sp_webservices 'addalias' alias_name , database_name
```

ユーザ定義 Web サービスを展開する場合：

```
sp_webservices 'deploy', ['all' | 'service_name']
```

ユーザ定義 Web サービスのデータベース・エイリアスを削除する場合：

```
sp_webservices 'dropalias' alias_name
```

ユーザ定義 Web サービスの WSDL ファイルにマップされるプロキシ・テーブルをリストする場合：

```
sp_webservices 'listudws' [, 'service_name']
```

ユーザ定義 Web サービスのデータベース・エイリアスをリストする場合：

```
sp_webservices 'listalias'
```

ユーザ定義 Web サービスの展開を解除する場合：

```
sp_webservices 'undeploy', ['all' | 'service_name']
```

パラメータ

```
'add', 'wsdl_uri' [, sds_name] [, 'method_name=proxy_table[,  
method_name=proxy_table]* ' ]
```

WSDL ファイルで指定された Web メソッドのプロキシ・テーブルを作成するために使用します。add オプションが正常に実行されると、list オプションが自動的に呼び出され、新しいプロキシ・テーブルのスキーマが示されます。

- *wsdl_uri* – 新しいプロキシ・テーブルにマップされる WSDL ファイルのロケーションです。このパラメータを指定すると、Web Services は、syswsdl テーブルに URI が存在することを確認します。
- *sds_name* – *interfaces* または *sql.ini* ファイルで ASE Web Services Engine 用に指定された名前です。デフォルト値は *ws* です。sysattributes テーブルにエントリがない場合は、エラーが発生します。
- *method_name* – プロキシ・テーブルにマップされる Web メソッドの名前です。指定する *method_name* は、関連する WSDL ファイルで指定された Web メソッドの名前にします。
- *proxy_table* – *method_name* で指定された Web メソッドがマップされるプロキシ・テーブルの名前です。

```
'addalias' alias_name , database_name
```

ユーザ定義 Web サービスでデータベース名を表すエイリアスの作成に使用します。各パラメータの意味は、次のとおりです。

- *alias_name* – 指定したデータベースのエイリアスです。このパラメータは必須です。
- *database_name* – エイリアスを指定するデータベースの名前です。このパラメータは必須です。

エイリアスは、データベース名を表す URL の部分を指定する場合に役に立ち、create service コマンドの userpath オプションと共に使用することで、エイリアスはユーザ定義 Web サービスへのアクセスに使用される URL を完全に制御することができます。

'deploy', ['all' | 'service_name']

ユーザ定義 Web サービスを展開し、HTTP または HTTPS を介して ASE Web Services Engine からアクセス可能にするために使用されます。各パラメータの意味は、次のとおりです。

- **all** — すべてのユーザ定義 Web サービスを現在のデータベース用に展開することを指定します。
- **service_name** — 展開するユーザ定義 Web サービスの名前です。

deploy および **undeploy** オプションは、ユーザ定義 Web サービスを利用可能にするかどうかを制御するために使用します。このオプションを使用するには、システム役割 **webservices_role** 権限が必要です。

all パラメータが指定されると、ASE Web Services Engine はユーザ定義 Web サービスの内部キャッシュを削除し、ユーザ定義 Web サービスに関するすべてのメタデータを Adaptive Server Enterprise から再度読み込みます。

現在展開中のユーザ定義 Web サービスの削除または名前変更はできません。

'dropalias' *alias_name*

データベース名を表すエイリアスの削除に使用します。*alias_name* は削除するエイリアスです。

展開したユーザ定義 Web サービスによって参照されている場合、エイリアスは削除できません。エイリアスを削除するには、そのエイリアスを参照しているユーザ定義 Web サービスの展開を解除します。

help[, 'option']

sp_webservices ストアド・プロシージャの使用法と使用例を示します。*'option'* の有効な値は、**add**、**list**、**remove**、**modify** です。

option の値を指定せずに **help** オプションを使用すると、**add**、**addalias**、**deploy**、**dropalias**、**list**、**listalias**、**listudws**、**modify**、**remove**、**undeploy** オプションに関する構文の簡単な説明が出力されます。

'list' [, 'wsdl_uri'] [, sds_name]

WSDL ファイルに記述される Web メソッドを示します。各パラメータの意味は、次のとおりです。

- *wsdl_uri* — マップされる WSDL ファイルの URI です。*wsdl_uri* の値を指定しない場合、*list* オプションは、プロキシ・テーブルにマップされたすべての Web メソッドに関する情報を表示します。
- *sds_name* — *interfaces* または *sql.ini* ファイルで ASE Web Services Engine 用に指定された SDS サーバの名前です。デフォルト値は *ws* です。*sysattributes* テーブルにエントリがない場合は、エラーが発生します。

wsdl_uri パラメータと *sds_name* パラメータの両方を指定しない場合は、*sysattributes* テーブル内のすべてのエントリが *wsdlid* の順にリストされます。

WSDL ファイルに記述された Web メソッドが、

- すでにプロキシ・テーブルにマップされている場合 — *list* オプションは各プロキシ・テーブルに関する情報を出力します。
- プロキシ・テーブルにマップされていない場合 — *list* オプションはプロキシ・テーブルの作成に使用できる SQL を出力します。

'listalias'

ユーザ定義 Web サービスにおけるすべてのエイリアスをリストするために使用します。

'listudws' [, 'service_name']

現在のデータベースのユーザ定義 Web サービスをリストするために使用します。*service_name* は、リストするユーザ定義 Web サービスの名前です。

service_name パラメータを指定しないと、すべてのユーザ定義 Web サービスがリストされます。

'modify', 'wsdl_uri', 'timeout=*time*'

WSDL ファイルの属性情報を変更するために使用します。各パラメータの意味は、次のとおりです。

- *wsdl_uri* — 属性情報を変更する WSDL ファイルの URI です。
- *time* — オペレーションがアボートされる前に Web メソッドが応答する秒単位の時間間隔です。

'remove', 'wsdl_uri' [, sds_name]

Web メソッドのプロキシ・テーブル・マッピングを削除するために使用します。各パラメータの意味は、次のとおりです。

- *wsdl_uri* — プロキシ・テーブルを削除する WSDL ファイルの URI です。
- *sds_name* — *interfaces* または *sql.ini* ファイルで ASE Web Services Engine 用に指定された SDS サーバの名前です。デフォルト値は *ws* です。

注意 *sysattributes* テーブルにエントリがない場合は、エラーが発生します。

'undeploy', [all | 'service_name']

ユーザ定義 Web サービスが HTTP または HTTPS を介して Adaptive Server Enterprise Web Services からアクセスできないようにするために使用します。各パラメータの意味は、次のとおりです。

- *all* — すべてのユーザ定義 Web サービスを現在のデータベースから展開解除することを指定します。
- *service_name* — 展開解除するユーザ定義 Web サービスの名前です。

ユーザ定義 Web サービスを利用可能にするかどうかを制御するには、*deploy* および *undeploy* オプションを使用します。このオプションを使用するには、システム役割 *webservices_role* 権限が必要です。

例

例 1 RPC/encoded Web メソッドを呼び出して 2 つの通貨間の為替レートを表示します。

- 1 *sp_webservices* の *add* オプションを使用して、プロキシ・テーブルに Web メソッドをマップします。

```
1> sp_webservices 'add',
'http://www.xmethods.net/sd/2001/CurrencyExchangeService.wsdl'
2> go
```

getRate Web メソッドは、同じ名前のプロキシ・テーブルにマップされます。

- 2 プロキシ・テーブルからこの Web メソッドを選択して呼び出します。

```
1> select * from getRate where _country1 = 'usa' and _country2 = 'india'
2> go
```


この `select` から返された結果には、パラメータで指定した為替レートが次のように示されます。

```
Result          _country1      _country2
43.000000      usa            india
(1 row affected)
```

例 2 Web メソッドを呼び出して XML ドキュメント内で株式情報を表示します。

- 1 `sp_webservices` の `add` オプションを使用して、プロキシ・テーブルに Web メソッドをマップします。

```
1> sp_webservices "add" , "http://www.webserviceX.net/stockquote.asmx?WSDL"
2> go
```

`GetQuote` Web メソッドは、同じ名前のプロキシ・テーブルにマップされます。

- 2 `GetQuote` プロキシ・テーブルの `outxml` カラムを選択して、この Web メソッドを呼び出します。

```
1> select outxml from GetQuote where _inxml = '<?xml version="1.0"
encoding="utf-8"?>
2>     <GetQuote xmlns="http://www.webserviceX.NET/">
3>         <symbol>SY</symbol>
4>     </GetQuote>'
5> go
```

この `select` の結果、株価情報が XML ドキュメントに次のように示されます。

```
outxml

<?xml version="1.0" encoding="UTF-8" ?><GetQuoteResponse
xmlns="http://www.webserviceX.NET/"><GetQuoteResult><StockQuotes><Stock>
<Symbol>SY</Symbol><Last>21.48</Last><Date>7/21/2005</Date><Time>4:01pm
</Time><Change>+1.72</Change><Open>20.00</Open><High>21.60</High>
<Low>19.91</Low><Volume>2420100</Volume><MktCap>1.927B</MktCap>
<PreviousClose>19.76</PreviousClose><PercentageChange>+8.70%
</PercentageChange><AnnRange>12.75 - 20.44</AnnRange><Earns>0.706</Earns>
<P-E>27.99</P-E><Name>SYBASE INC</Name></Stock></StockQuotes>
</GetQuoteResult></GetQuoteResponse>
```

(1 row affected)

例 3 前の例でプロキシ・テーブルにマップされた **GetQuote Web** メソッドをビューを介して呼び出し、株式情報を表示します。

- 1 この **Web** サービスを使用するには、株式を示す記号を保持するテーブルを作成してください。

```
1> create table stocksymbol(symbol varchar(100))
2> go
```

- 2 **stocksymbol** テーブルにデータを挿入します。

```
1> insert stocksymbol values("SY")
2> insert stocksymbol values("ORCL")
3> go
```

- 3 **GetQuote Web** メソッドを呼び出すビューを作成します。

```
1> CREATE VIEW getstockvw as
2> select Symbol = xmlextract('//Stock/Symbol/text()',outxml returns varchar(5)),
3>    Name = xmlextract('//Stock/Name/text()',outxml returns varchar(20)),
4>    Time = xmlextract('//Stock/Time/text()',outxml returns varchar(10)),
5>    Date = xmlextract('//Stock/Date/text()',outxml returns date),
6>    High = xmlextract('//Stock/High/text()',outxml returns decimal(15,2)),
7>    Low = xmlextract('//Stock/Low/text()',outxml returns decimal(15,2))
8> FROM GetQuote ,stocksymbol
9> WHERE _inxml = '<GetQuote
xmlns="http://www.webserviceX.NET/"><symbol>'+symbol+'</symbol></GetQuote>'
10> go
```

- 4 **getstockvw** ビューから選択して、**GetQuotes** メソッドからの出力を表示します。

```
1> select * from getstockvw
2> go
```

この **select** の結果、株価情報がビュー定義で指定されたパラメータにより次のように示されます。

Symbol	Name	Time	Date	High	Low
SY	SYBASE INC	4:01pm	Jul 21 2005	21.60	19.91
ORCL	ORACLE CORP	4:00pm	Jul 21 2005	14.05	13.54
MSFT	MICROSOFT CP	4:00pm	Jul 21 2005	26.48	26.19

(3 rows affected)

例 4 ユーザ “bob” によって **pubs2** データベースに入力された次のコマンドに対する監査テーブル・エントリを示します。

```
sp_webservices 'deploy', 'all'
```

対応する監査テーブル・エントリには、110、bob、pubs2 が、event、loginname、dbname カラムの値としてそれぞれリストされます。extrainfo カラムの内容は次のようになります。

```
webservices_role; deploy_all; ; ; ; bob/ase;
```

例 5 ユーザ “bob” によって pubs2 データベースに入力された次のコマンドに対する監査テーブル・エントリを示します。

```
sp_webservices 'deploy', 'rawservice'
```

対応する監査テーブル・エントリには、110、bob、pubs2 が、event、loginname、dbname カラムの値としてそれぞれリストされます。extrainfo カラムの内容は次のようになります。

```
webservices_role; deploy; ; ; ; bob/ase;
```

例 6 ユーザ “bob” によって pubs2 データベースに入力された次のコマンドに対する監査テーブル・エントリを示します。

```
sp_webservices 'undeploy', 'all'
```

対応する監査テーブル・エントリには、111、bob、pubs2 が、event、loginname、dbname カラムの値としてそれぞれリストされます。extrainfo カラムの内容は次のようになります。

```
webservices_role; undeploy_all; ; ; ; bob/ase;
```

例 7 ユーザ “bob” によって pubs2 データベースに入力された次のコマンドに対する監査テーブル・エントリを示します。

```
sp_webservices 'undeploy', 'rawservice'
```

対応する監査テーブル・エントリには、111、bob、pubs2 が、event、loginname、dbname カラムの値としてそれぞれリストされます。extrainfo カラムの内容は次のようになります。

```
webservices_role; deploy; ; ; ; bob/ase;
```

sysaudits テーブルのカラムの詳細な説明については、Adaptive Server Enterprise の『システム管理ガイド』を参照してください。

使用法

```
sp_webservices add
```

Web メソッドの *method_name* 値と *proxy_table* 値を指定しない場合、その Web メソッド用に生成されるプロキシ・テーブルは、デフォルトで、WSDL ファイルに指定した Web メソッドの名前になります。この Web メソッドの名前を持つプロキシ・テーブルがすでにある場合は、次のような名前での新しいプロキシ・テーブルが生成されます。

```
method_nameN
```

上記のパラメータの意味は、次のとおりです。

- *method_name* – デフォルトのプロキシ・テーブル名です。
- *N* – Web メソッドの一連の各マッピングを示す 1～9 の数字です。プロキシ・テーブルの複製は 99 個まで可能です。

Web メソッドの *method_name* 値と *proxy_table* 値を指定する場合、プロキシ・テーブル名は新しい名前にします。*proxy_table* で指定した名前を持つプロキシ・テーブルがすでにある場合はエラーが発生し、**add** オプションで指定した Web メソッドはどれもプロキシ・テーブルにマップされません。

add オプションからの出力には、プロキシ・テーブルに正常にマップされたメソッドとマップされなかったメソッドがリストされます。マップされなかった Web メソッドのプロキシ・テーブルの名前は、**add** オプションの出力で **NULL** として示されます。

注意 入力と出力に使用されるカラムは、RPC/encoded Web メソッド用に生成されたプロキシ・テーブルと document/literal Web メソッド用に生成されたプロキシ・テーブルでは異なります。RPC/encoded Web メソッドを表すプロキシ・テーブルには、各入力パラメータと出力パラメータのカラムが含まれています。document/literal Web メソッドを表すプロキシ・テーブルには、*_inxml* と *outxml* の 2 つのカラムが含まれています。

ユーザ定義 Web サービスのセキュリティ

システム役割 *webservices_role* は、*sp_webservices* の **deploy** および **undeploy** オプションを使用するために必要になります。ユーザ定義 Web サービスを実行するには、対応するストアド・プロシージャを実行するための有効なログインと権限が必要です。

ユーザ定義 Web サービスの作成、削除、実行を行うには、Adaptive Server Enterprise でストアド・プロシージャを作成、削除、実行するために必要な権限と同じ権限が必要です。**grant** および **revoke** コマンドを使用して適切な権限を設定する方法の詳細については、Adaptive Server Enterprise の『システム管理ガイド』を参照してください。

パーミッション

sp_webservices を実行するには、(展開および展開解除のための) *webservices_role* が付与されたユーザであることが必要です。パーミッション・チェックは、細密なパーミッションの設定によって異なりません。

監査

- 監査イベントの番号 110 は、sp_webservices の deploy オプションに対応しています。
- 監査イベントの番号 111 は、sp_webservices の undeploy オプションに対応しています。

ユーザ定義 Web サービスは、Adaptive Server Enterprise のストアード・プロシージャをモデルにしています。ユーザ定義 Web サービスの操作において、Adaptive Server Enterprise は既存のストアード・プロシージャ向けの監査範囲を使用して、次のイベントを生成します。

- ユーザ定義 Web サービスの作成 – イベント 11 「プロシージャの作成」を生成
- ユーザ定義 Web サービスの削除 – イベント 28 「プロシージャの削除」を生成
- ユーザ定義 Web サービスの実行 – イベント 38 「ストアード・プロシージャの実行」を生成

既存の監査機能の詳細については、『システム管理ガイド』を参照してください。

既存の監査機能に加えて、Adaptive Server Enterprise では sp_webservices の deploy オプションおよび undeploy オプション用に 2 つの監査イベントが用意されています。

監査レコードは sybsecurity システム・テーブルに保管されます。Web サービスの監査を有効にするには、次のコマンドを実行します。

```
sp_audit "security", "all", "all", "on"
```

参照

コマンド create service

マニュアル 『Web Services ユーザーズ・ガイド』

sp_who

説明 現在のすべての Adaptive Server ユーザとプロセス、または特定のユーザとプロセスの情報についてレポートします。Adaptive Server がタスクを実行するために使用するスレッド・プールを記述する `thread_pool` カラムを含む。

プロセス・モードの考慮事項 `sp_who` は、`threadpool` カラムを含みません。

構文 `sp_who [loginame | "spid"]`

パラメータ `loginame`
レポートを要求するユーザの Adaptive Server ログイン名です。

`spid`
レポートを要求するプロセスの ID 番号です。プロセスの ID 番号は引用符で囲みます (Adaptive Server は `char` 型を予期します)。

例 **例 1** Adaptive Server で実行中のプロセスについてレポートします。`sp_who` 以外に実行中のユーザ・プロセスがなくても、サーバは作業内容を表示します。アイドル・サイクル中に、ハウスキーピング・ウォッシュ・タスクはダーティ・バッファをバッファ・ウォッシュ領域に移動し、ハウスキーピング・チャオ・タスクは他の管理タスクを実行します。ハウスキーピング・ガーベジ・コレクション・タスクは、論理的に削除されたデータをクリーン・アップして、テーブルの領域が解放されるようにローをリセットします。この動作は一般ユーザの優先順位で行われます。

`sp_who`

fid	spid	status	loginame	origname	hostname	blk_spid	dbname
	tempdbname	cmd		block_xloid	threadpool		
0	2	sleeping	NULL	NULL	NULL	0	master
	tempdb	DEADLOCK TUNE		0	syb_default_pool		
0	3	sleeping	NULL	NULL	NULL	0	master
	tempdb	ASTC HANDLER		0	syb_default_pool		
0	4	sleeping	NULL	NULL	NULL	0	master
	tempdb	CHECKPOINT SLEEP		0	syb_default_pool		
0	5	sleeping	NULL	NULL	NULL	0	master
	tempdb	HK WASH		0	syb_default_pool		
0	6	sleeping	NULL	NULL	NULL	0	master
	tempdb	HK GC		0	syb_default_pool		
0	7	sleeping	NULL	NULL	NULL	0	master
	tempdb	HK CHORES		0	syb_default_pool		
0	8	sleeping	NULL	NULL	NULL	0	master
	tempdb	PORT MANAGER		0	syb_default_pool		
0	9	sleeping	NULL	NULL	NULL	0	master

```

tempdb          NETWORK HANDLER          0  syb_default_pool
0  10  sleeping  NULL  NULL  0  NULL  0  master
tempdb          LICENSE HEARTBEAT        0  syb_default_pool
0  13  sleeping  NULL  NULL  0  NULL  0  master
tempdb          NETWORK HANDLER          0  syb_default_pool
0  14  sleeping  NULL  NULL  0  NULL  0  master
tempdb          NETWORK HANDLER          0  syb_default_pool
0  17  sleeping  NULL  NULL  0  NULL  0  master
tempdb          NETWORK HANDLER          0  syb_default_pool
0  20  sleeping  NULL  NULL  0  NULL  0  master
tempdb          NETWORK HANDLER          0  syb_default_pool
0  26  running   sa      sa  tiger.sybase.com  0  master
tempdb          INSERT                    0  syb_default_pool

```

例 2 Adaptive Server で実行中のプロセスについてレポートします。プロセス 11 (テーブル上の `select into`) は、プロセス 8 (同じテーブル上の `begin transaction` とその後の `insert`) によってブロックされています。プロセス 8 では、現在の `loginame` は “robert” になっています。元の `loginame` は “sa” です。ログイン “sa” は `set proxy` コマンドを実行して、ユーザ “robert” と同一化しました。

```
sp_who
```

```

fid  spid  status      loginame  origname  hostname      blk_spid  dbname
----  ----  -
tempdbname  cmd      block_xloid  threadpool
-----  -
0    1  recv sleep      bird      bird      jazzy        0         master
tempdb  AWAITING COMMAND  0         syb_default_pool
0    2  sleeping  NULL      NULL      NULL        0         master
tempdb  NETWORK HANDLER  0         syb_default_pool
0    3  sleeping  NULL      NULL      NULL        0         master
tempdb  MIRROR HANDLER  0         syb_default_pool
0    4  sleeping  NULL      NULL      NULL        0         master
tempdb  AUDIT PROCESS   0         syb_default_pool
0    5  sleeping  NULL      NULL      NULL        0         master
tempdb  CHECKPOINT SLEEP  0         syb_default_pool
0    6  recv sleep      rose      rose petal  0         master
tempdb  AWAITING COMMAND  0         syb_default_pool
0    7  sleeping  NULL      NULL      actor       0         sybssystemdb
tempdb  ASTC HANDLER    0         syb_default_pool
0    8  running  robert     sa        helos       0         master
tempdb  SELECT          0         syb_default_pool
0    9  send sleep      daisy     daisy     chain       0         pubs2
tempdb  SELECT          0         syb_default_pool
0   10  alarm sleep      lily      lily      pond        0         master
tempdb  WAITFOR        0         syb_default_pool
0   11  lock sleep      viola     viola     cello       8         pubs2
tempdb  INSERT          0         syb_default_pool

```

例 3 ユーザ “joe” が実行しているプロセスをレポートします。

```
sp_who joe
fid spid status  loginame  origname  hostname          blk_spid  dbname
  tempdbname  cmd                block_xloid  threadpool
-----
0   28  recv sleep      joe       joe  tiger.sybase.com    0          pubs2
      tempdb          SELECT                0          syb_default_pool
```

例 4 Adaptive Server のプロセス番号 17 が実行している内容がレポートされます。

```
sp_who "17"
fid spid status  loginame  origname  hostname          blk_spid  dbname
  tempdbname  cmd                block_xloid  threadpool
-----
0   17  sleeping  NULL      NULL      NULL              0          pubs2
      tempdb  NETWORK HANDLER      0          syb_default_pool
```

例 5 システムが呼び出したトランザクションまたはコマンドのロールバックをレポートします。

```
sp_who
fid spid status  loginame  origname  hostname          blk_spid  dbname
  tempdbname  cmd                block_xloid  threadpool
-----
0   28  running  joe       joe  tiger.sybase.com    0          pubs2
      tempdb  rollback                0          syb_default_pool
```

使用法

- `sp_who` は、指定のユーザまたは Adaptive Server のプロセスに関する情報をレポートします。
- パラメータを指定しなければ、`sp_who` は、すべてのデータベース内でどのユーザがどんな処理をしているかをレポートします。
- `sp_who` によって返されるカラムは、次のとおりです。

カラム	説明
fid	ロックが属するファミリ (その他のコーディネーティング・プロセスとそのワーカー・プロセス) を示します。詳細については、 sp_familylock を参照してください。
spid	プロセスの番号を示します。システム管理者は、Transact-SQL の <code>kill</code> コマンドでこの番号を指定して、そのプロセスを停止することができます。
status	プロセスが実行中であるか、スリープしているかどうかを示します。
loginame	プロセスを開始したユーザのログイン名またはエイリアスです。すべてのシステム・プロセスでは、loginame は NULL です。

カラム	説明
origname	loginame がエイリアスの場合、origname は実際のログイン名を表示します。そうでない場合、origname は loginame と同じ情報を表示します。
hostname	データベースが常駐するサーバの名前です。
blk_spid	ブロッキング・プロセスがあれば、そのプロセスの ID 番号が含まれます。ブロッキング・プロセスとは、他のプロセスが必要とするリソースを保持しているプロセスで、何らかの影響を受けているか、排他的なロックが割り当てられています。
dbname	プロセスが実行されているデータベースの名前です。
tempdb	セッションに割り当てられるテンポラリ・データベース。
cmd	現在実行されているコマンドまたはプロセスを識別します。if または while のループなどの条件文が評価されると、cond を返す。
block_xloid	ブロック・トランザクションのユニークなロック所有者 ID を識別します。
threadpool	タスクが使用するスレッド・プール。

- シングルエンジン・サーバ上で sp_who を実行すると、現在実行中の sp_who プロセスが表示されます。また、実行可能なプロセスか、いずれかのスリープ・ステータスにあるプロセスも表示されます。マルチエンジン・サーバでは、エンジンごとに「実行中」のプロセスが存在する可能性があります。
- ミラー・ディスクまたはリモート・プロシージャ・コールを有効にすると、sp_who が提供するレポートに、ミラー・ハンドラとサイト・ハンドラも表示されます。

パーミッション

sp_who は、すべてのユーザが実行できます。パーミッション・チェックは、細密なパーミッションの設定によって異なります。

監査

sysaudits テーブルの event カラムと extrainfo カラムの値は次のとおりです。

イベント	監査オプション	監査対象のコマンドまたはアクセス	extrainfo の情報
38	exec_procedure	プロシージャの実行	<ul style="list-style-type: none"> • Roles — 現在のアクティブな役割 • Keywords or options — NULL • Previous value — NULL • Current value — NULL • Other information — すべての入力パラメータ • Proxy information — set proxy が有効な場合は元のログイン名

参照

コマンド [kill](#)
 システム・プロシージャ [sp_familylock](#)、[sp_lock](#)

sp_xmlschema

説明	Adaptive Server データベースで <code>spt_xmlcatalog</code> ユーザ・テーブルを作成および管理します。 <code>spt_xmlcatalog</code> は、 <code>xmlvalidate</code> 関数が XML ドキュメントを検証するために使用するスキーマの定義を保存します。
構文	構文、例、および使用方法については、 <i>XML Services</i> 説明書の「 <code>sp_xmlschema</code> 」を参照してください。
使用法	

この章では、カタログ・ストアド・プロシージャについて説明します。カタログ・ストアド・プロシージャは、表形式のシステム・テーブルから情報を検索します。

この章では、次の項目について説明します。

トピック名	ページ
概要	857
オプション・パラメータの指定	859
パターン一致	860
システム・プロシージャ・テーブル	860
ODBC のデータ型	860

概要

表 2-1 は、この章で説明するカタログ・ストアド・プロシージャを示します。

表 2-1: カatalog・ストアド・プロシージャ

プロシージャ	説明
sp_column_privileges	テーブルまたはビュー内の 1 つまたは複数のカラムのパーミッション情報を返す。
sp_columns	1 つまたは複数のカラムに保管できるデータ型の情報を返す。
sp_databases	Adaptive Server にあるデータベースのリストを返す。
sp_datatype_info	特定の ODBC データ型またはすべての ODBC データ型の情報を返す。
sp_fkeys	現在のデータベース内の <code>create table</code> コマンドまたは <code>alter table</code> コマンドで作成した、外部キー制約に関する情報を返す。
sp_pkeys	1 つのテーブルに対する <code>create table</code> コマンドまたは <code>alter table</code> コマンドで作成した、主キー制約 (プライマリ・キー制約) の情報を返す。
sp_server_info	Adaptive Server の属性名と現在の値のリストを返す。
sp_special_columns	テーブルまたはビュー内のローをユニークに識別する最適カラム・セットを返す。また、トランザクションによってロー内の値が更新されると自動的に値が更新されるカラムのリストも返す。

プロシージャ	説明
<code>sp_sproc_columns</code>	ストアド・プロシージャの入力パラメータとリターン・パラメータについての情報を返す。
<code>sp_statistics</code>	1つのテーブルのインデックス・リストを返す。
<code>sp_stored_procedures</code>	1つまたは複数のストアド・プロシージャの情報を返す。
<code>sp_table_privileges</code>	テーブルまたはビューのすべてのカラムの権限情報を返す。
<code>sp_tables</code>	<code>from</code> 句内に表示できるオブジェクトのリストを返す。

カタログ・ストアド・プロシージャは、システム・テーブルから表形式で情報を取り出します。

各カタログ・ストアド・プロシージャは、インストール時に `installmaster` によって作成され、`sybsystemprocs` データベース内に配置されます。各カタログ・ストアド・プロシージャはシステム管理者が所有します。

ほとんどのプロシージャは、どのデータベースからでも実行できます。カタログ・ストアド・プロシージャを `sybsystemprocs` 以外のデータベースから実行すると、カタログ・ストアド・プロシージャは、実行したデータベースの中にあるシステム・テーブルの情報を検索します。

すべてのカタログ・ストアド・プロシージャは、独立性レベル 1 で実行されます。

すべてのカタログ・ストアド・プロシージャは、リターン・ステータスをレポートします。たとえば、次のレポートは、プロシージャが正しく実行されたことを表します。このマニュアルでは、リターン・ステータスの例を省略します。

```
return status = 0
```

オプション・パラメータの指定

カタログ・ストアド・プロシージャのパラメータ値内で句読表記、埋め込みブランク、予約語を使用する場合は、それを一重引用符または二重引用符で囲んでください。パラメータがデータベース名または所有者名で修飾されているオブジェクト名である場合は、その名前全体を一重引用符か二重引用符で囲みます。

注意 カタログ・ストアド・プロシージャのパラメータには区切り識別子を使用しないでください。予期しない結果が生じる可能性があります。

ほとんどの場合、次の形式でカタログ・ストアド・プロシージャにパラメータを指定します。

```
@parametername = value
```

このように指定した方が、すべてのパラメータを指定するよりも便利です。構文内のパラメータ名は、プロシージャによって定義されるパラメータ名に対応します。

たとえば、`sp_columns` の構文は次のようになります。

```
sp_columns table_name [, table_owner]
           [, table_qualifier] [, column_name]
```

`sp_columns` を使用して、特定の列に関する情報を検索するには、次の構文を使用します。

```
sp_columns publishers, @column_name = "pub_id"
```

この構文を使用すると、すべてのパラメータを指定したコマンドと同じ情報が得られます。

```
sp_columns publishers, "dbo", "pubs2", "pub_id"
```

また、プレースホルダとして “null” も使用できます。

```
sp_columns publishers, null, null, "pub_id"
```

指定したパラメータの数が、システム・プロシージャが予期しているパラメータ数を超えると、Adaptive Server は超過したパラメータを無視します。

パターン一致

ASE は、ワイルドカードとして `_`、`%`、`[]`、および `[^]` をサポートしています。ただし、最大の相互運用性を確保するには、ANSI SQL 規格で規定されたワイルドカード文字である `%` と `_` のみを使用してください。

システム・プロシージャ・テーブル

カタログ・ストアド・プロシージャ `sp_columns`、`sp_datatype_info`、`sp_special_columns`、`sp_sproc_columns` は、`sybsystemprocs` データベース内のカタログ・ストアド・プロシージャ・テーブル `spt_datatype_info`、`spt_datatype_info_ext`、`spt_server_info` を使用して、ステータス・ビットなどの内部システム値を人間が判読できるフォーマットに変換します。

カタログ・ストアド・プロシージャ `sp_column_privileges` と `sp_table_privileges` は、テンポラリ・テーブルを作成してから削除します。

ODBC のデータ型

表 2-2 と表 2-3 は、`sp_columns` および `sp_sproc_columns` が `data_type` カラム内に返すデータ型のコード番号と、それに対応するデータ型名を示します。このリストは、ODBC (Open Database Connectivity) の API (アプリケーション・プログラミング・インタフェース) から引用したものです。

表 2-2: ODBC データ型のコード番号

データ型	コード番号
char	1
decimal	3
double precision	8
float	6
integer	4
numeric	2
real	7
smallint	5
varchar	12
wchar	-8
wvarchar	-9
wlongvarchar	-10

表 2-3: 拡張データ型のコード番号

データ型	コード番号
bigint	-5
binary (ビット・データ型)	-2
bit	-7
date	9
java.lang.Object	1111
long univarchar	-10
long varbinary	-4
long varchar	-1
time	10
timestamp	11
tinyint	-6
unichar	-8
univarchar	-9
varbinary (ビット可変データ型)	-3

sp_column_privileges

説明	テーブルまたはビュー内の 1 つまたは複数のカラムのパーミッション情報を返します。
構文	<code>sp_column_privileges table_name [, table_owner [, table_qualifier [, column_name]]]</code>
パラメータ	<p>table_name テーブルの名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。</p> <p>table_owner テーブル所有者の名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。テーブル所有者を指定しなければ、<code>sp_column_privileges</code> は、現在のユーザが所有しているテーブルを検索してから、データベース所有者が所有しているテーブルを検索します。</p> <p>table_qualifier データベースの名前を指定します。値は、現在のデータベースの名前または <code>null</code> です。</p> <p>column_name パーミッションを表示するカラムの名前を指定します。複数のカラムの情報を要求する場合は、ワイルドカード文字を使用します。カラム名を指定しなければ、指定したテーブル内のすべてのカラムに関するパーミッション情報が返されます。</p>

例

```
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

使用法

- `sp_column_privileges` の結果セットは次のようになります。

カラム	データ型	説明
<code>table_qualifier</code>	<code>varchar(32)</code>	<code>table_name</code> パラメータ用に指定されたテーブルが格納されるデータベースの名前。
<code>table_owner</code>	<code>varchar(32)</code>	テーブル所有者。 <code>table_owner</code> パラメータに値が指定されていない場合、この値は現在の所有者またはデータベース所有者になる。
<code>table_name</code>	<code>varchar(32)</code>	<code>table_name</code> パラメータに指定された名前。この値に <code>NULL</code> を指定することはできない。
<code>column_name</code>	<code>varchar(32)</code>	指定されたカラム名。文にカラム名が指定されていない場合、結果には指定したテーブル内にあるすべての列が含まれる。
<code>grantor</code>	<code>varchar(32)</code>	<code>column_name</code> に対するパーミッションを <code>grantee</code> に付与したデータベース・ユーザの名前。この値に <code>NULL</code> を指定することはできない。
<code>grantee</code>	<code>varchar(32)</code>	<code>column_name</code> に対するパーミッションを <code>grantor</code> から付与されたデータベース・ユーザの名前。この値に <code>NULL</code> を指定することはできない。
<code>privilege</code>	<code>varchar(32)</code>	カラムの権限を識別する。権限は、次のうちのいずれか1つ。 <ul style="list-style-type: none"> • <code>SELECT</code> - 付与されたユーザはカラムのデータを検索できる。 • <code>UPDATE</code> - 付与されたユーザはカラム内のデータを更新できる。 • <code>REFERENCE</code> - 付与されたユーザは参照制約のみを使用できる。
<code>is_grantable</code>	<code>varchar(3)</code>	付与されたユーザが、他のユーザに権限を付与できるかどうかを示す。値は <code>YES</code> 、 <code>NO</code> 、 <code>NULL</code> 。

パーミッション `sp_column_privileges` は、すべてのユーザが実行できます。

sp_columns

説明 1つまたは複数のカラムに保管できるデータ型の情報を返します。

構文 `sp_columns table_name [, table_owner] [, table_qualifier] [, column_name]`

パラメータ

table_name

テーブルまたはビューの名前を指定します。複数のテーブルの情報を要求するには、ワイルドカード文字を使用します。

table_owner

テーブルまたはビューの所有者名を指定します。複数のユーザが所有しているテーブルの情報を要求するには、ワイルドカード文字を使用します。テーブル所有者を指定しなければ、`sp_columns` は、現在のユーザが所有しているテーブルを検索してから、データベース所有者が所有しているテーブルを検索します。

table_qualifier

データベースの名前を指定します。現在のデータベースまたは NULL のどちらかを指定できます。

column_name

情報を要求するカラムの名前を指定します。複数のカラムに関する情報を要求するには、ワイルドカード文字を使用します。

例 **例 1** `publishers` テーブル内にある、“p”で始まる名前のカラムの情報をすべて表示します。

```
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
```

例 2 “s”で始まる名前のテーブル内にある、“st”が冒頭に付いているカラムの情報をすべて表示します。

```
sp_columns "s%", null, null, "st%"
```

使用法

- `sp_columns` の結果セットは次のようになります。

カラム	データ型	説明
<code>table_qualifier</code>	<code>varchar(32)</code>	<code>table_name</code> パラメータ用に指定されたテーブルが格納されるデータベースの名前。
<code>table_owner</code>	<code>varchar(32)</code>	テーブル所有者。 <code>table_owner</code> パラメータに値が指定されていない場合、この値は現在の所有者またはデータベース所有者になる。
<code>table_name</code>	<code>varchar(32)</code>	NOT NULL。
<code>column_name</code>	<code>varchar(32)</code>	NOT NULL。
<code>data_type</code>	<code>smallint</code>	ODBC データ型の整数コード。ODBC 型にマップできないデータ型の場合は NULL。
<code>type_name</code>	<code>varchar(30)</code>	データ型を表す文字列。基本となる DBMS は、このデータ型名を示す。
<code>precision</code>	<code>int</code>	有効桁数。
<code>length</code>	<code>int</code>	データ型のバイト長。
<code>scale</code>	<code>smallint</code>	小数点以下の桁数。
<code>radix</code>	<code>smallint</code>	数値データ型の基底データ型。
<code>nullable</code>	<code>smallint</code>	値 1 は NULL が可能であることを示し、値 0 は NOT NULL が可能であることを示す。
<code>remarks</code>	<code>varchar(254)</code>	
<code>ss_data_type</code>	<code>smallint</code>	Adaptive Server のデータ型。
<code>colid</code>	<code>tinyint</code>	結果セットに追加されるカラム。
<code>column_def</code>	<code>varchar(255)</code>	NULL。
<code>sql_data_type</code>	<code>smallint</code>	Adaptive Server のデータ型。
<code>sql_datetime_sub</code>	<code>smallint</code>	NULL。
<code>char_octet_length</code>	<code>int</code>	<p><code>char_octet_length</code> の値は、<code>char_octet_length</code> が次のデータ型のときに <code>precision</code> カラムの値と同じです。</p> <ul style="list-style-type: none"> • <code>binary</code> • <code>char</code> • <code>image</code> • <code>nchar</code> • <code>nvarchar</code> • <code>sysname</code> • <code>text</code> • <code>timestamp</code> • <code>varbinary</code> • <code>varchar</code> <p>これ以外の場合、<code>char_octet_length</code> の値は 0 になる。</p>

カラム	データ型	説明
ordinal_position	int	テーブル内のカラムの位置を序数で表したもの。テーブルの最初のカラムは1になる。
is_nullable	varchar(3)	カラムまたはパラメータに値として NULL を使用できるかどうかを示す。syscolumns から取得。

- sp_columns は、double precision に定義されたカラムに対して、type_name を float 値としてレポートし、data_type を 6 としてレポートします。Adaptive Server の double precision データ型は、ODBC 仕様で指定されている値の範囲をサポートする float 実装データ型です。

パーミッション

sp_columns は、すべてのユーザが実行できます。

sp_databases

説明 Adaptive Server 内にあるデータベースのリストを返します。

構文 sp_databases

パラメータ なし

例

```

sp_databases
database_name      database_size      remarks
-----
master             5120              NULL
model              2048              NULL
mydb               2048              NULL
pubs2              2048              NULL
sybsecurity        5120              NULL
sybssystemprocs    16384             NULL
tempdb             2048              NULL

```

使用法 sp_databases の結果セットは次のようになります。

カラム	データ型	説明
database_name	char(32)	データベース名は NOT NULL。
database_size	bigint	データベースのサイズ(キロバイト単位)。
remarks	varchar(254)	Adaptive Server は、常に NULL を返す。

パーミッション sp_databases は、すべてのユーザが実行できます。

sp_datatype_info

説明 特定の ODBC データ型またはすべての ODBC データ型の情報を返します。

構文 sp_datatype_info [data_type]

パラメータ data_type
 情報を要求する ODBC データ型のコード番号を指定します。データ型のコード番号については、表 2-2 (861 ページ) と表 2-3 (861 ページ) を参照してください。

使用法 sp_datatype_info の結果セットは次のようになります。

カラム	データ型	説明
type_name	varchar(30)	DBMS に依存するデータ型名 (sp_columns 結果セット内の type_name カラムと同じ)。
data_type	smallint	このデータ型の全カラムがすべてマップされる、マッピング先 ODBC データ型のコード。
precision	int	データ・ソースのデータ型に使用される最大精度。精度が適用できないデータ型の場合は 0 が返される。
literal_prefix	varchar(32)	リテラルの前に付ける文字 (プレフィクス)。たとえば、文字型では一重引用符 (')、バイナリでは 0x を付ける。
literal_suffix	varchar(32)	リテラルの終了に使用する文字。たとえば、文字型では一重引用符 (') を使用する。バイナリでは何も使用しない。
create_params	varchar(32)	このデータ型の作成パラメータに関する記述。
nullable	smallint	1 は、このデータ型を作成すると null 値が許容されることを示す。0 はそうでないことを示す。
case_sensitive	smallint	1 はこの型のカラムの大文字と小文字が (照合用に) すべて区別されることを示す。0 は区別しないことを示す。
searchable	smallint	1 は where 句でこの型のカラムを使用できることを示す。
unsigned_attribute	smallint	1 はデータ型に符号がないことを示す。0 は符号付きのデータ型であることを示す。
money	smallint	1 は通貨データ型であることを示す。0 は通貨データ型でないことを示す。
auto_increment	smallint	1 は自動的に増分されるデータ型であることを示す。0 は増分されないことを示す。
local_type_name	varchar(128)	データ型のデータ・ソース従属名のローカライズ・バージョン。

パーミッション sp_datatype_info は、すべてのユーザが実行できます。

sp_fkeys

説明	<code>create table</code> または <code>alter table</code> コマンドによって作成された、現在のデータベース内の外部キー制約の情報を返します。
構文	<pre>sp_fkeys <i>pktable_name</i> [, <i>pktable_owner</i>] [, <i>pktable_qualifier</i>] [, <i>fktable_name</i>] [, <i>fktable_owner</i>] [, <i>fktable_qualifier</i>]</pre>
パラメータ	<p><i>pktable_name</i></p> <p>プライマリ・キー・テーブルの名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。<i>pktable_name</i>、<i>fktable_name</i>、または両方を指定してください。</p> <p><i>pktable_owner</i></p> <p>プライマリ・キー・テーブル所有者の名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。テーブル所有者を指定しなければ、<code>sp_fkeys</code> は、現在のユーザが所有しているテーブルを検索してから、データベース所有者が所有しているテーブルを検索します。</p> <p><i>pktable_qualifier</i></p> <p>プライマリ・キー・テーブルを格納しているデータベースの名前を指定します。現在のデータベースまたは <code>NULL</code> のどちらかを指定できます。</p> <p><i>fktable_name</i></p> <p>外部キー・テーブルの名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。<i>fktable_name</i>、<i>pktable_name</i>、または両方を指定してください。</p> <p><i>fktable_owner</i></p> <p>外部キー・テーブル所有者の名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。<i>fktable_owner</i> を指定しなければ、<code>sp_fkeys</code> は、現在のユーザが所有しているテーブルを検索してから、データベース所有者が所有しているテーブルを検索します。</p> <p><i>fktable_qualifier</i></p> <p>外部キー・テーブルを格納しているデータベースの名前を指定します。現在のデータベースまたは <code>null</code> のどちらかを指定できます。</p>

使用法

- `sp_fkeys` は、`create table` または `alter table` コマンドによって作成された、現在のデータベース内の外部キー制約の情報を返します。外部キーとは、別のテーブル内の「**プライマリ・キー・カラム**」に論理的に従属するテーブル内のキー・カラムのことです。
- `sp_fkeys` の結果セットは次のようになります。

カラム	データ型	説明
<code>pktable_qualifier</code>	<code>varchar(32)</code>	プライマリ・キー・テーブルを格納しているデータベース。
<code>pktable_owner</code>	<code>varchar(32)</code>	プライマリ・キー・テーブルの所有者。
<code>pktable_name</code>	<code>varchar(32)</code>	NOT NULL。
<code>pkcolumn_name</code>	<code>varchar(32)</code>	NOT NULL。
<code>fktable_qualifier</code>	<code>varchar(32)</code>	外部キー・テーブルを格納しているデータベース。
<code>fktable_owner</code>	<code>varchar(32)</code>	外部キー・テーブルの所有者。
<code>fktable_name</code>	<code>varchar(32)</code>	NOT NULL。
<code>fkcolumn_name</code>	<code>varchar(32)</code>	NOT NULL。
<code>key_seq</code>	<code>smallint</code>	NOT NULL。複数カラムで構成されたプライマリ・キー内のカラムのシーケンス番号。
<code>update_rule</code>	<code>smallint</code>	SQL オペレーションが <code>UPDATE</code> の場合に外部キーに適用される処理。このカラムには 0 が返される。
<code>delete_rule</code>	<code>smallint</code>	SQL オペレーションが <code>DELETE</code> の場合に、外部キーに適用される処理。このカラムには 0 が返される。

- プライマリ・キーと外部キーの両方の宣言は `create table` 文または `alter table` 文で行っておいください。
- プライマリ・キー・テーブルの名前を指定しても、外部キー・テーブル名が `NULL` であれば、`sp_fkeys` は、指定したテーブルに外部キーが含まれているすべてのテーブルを返します。外部キー・テーブル名を指定しても、プライマリ・キー・テーブル名が `NULL` であれば、`sp_fkeys` は、外部キー・テーブル内の外部キーに対するプライマリ・キー/外部キー関係に関連付けられたすべてのテーブルを返します。
- `sp_fkeys` は、[sp_commonkey](#)、[sp_foreignkey](#)、または [sp_primarykey](#) によって宣言されたキーについては情報を返しません。

パーミッション

`sp_fkeys` は、すべてのユーザが実行できます。

sp_pkeys

説明 create table コマンドまたは alter table コマンドによって1つのテーブル用に作成された、プライマリ・キー制約の情報を返します。

構文 sp_pkeys table_name [, table_owner]
[, table_qualifier]

パラメータ

table_name

テーブルの名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。

table_owner

テーブル所有者の名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。

table_owner を指定しなければ、sp_pkeys は、現在のユーザが所有しているテーブルを検索してから、データベース所有者が所有しているテーブルを検索します。

table_qualifier

テーブルを格納しているデータベースの名前を指定します。現在のデータベースまたは NULL のどちらかを指定できます。

使用法

- sp_pkeys の結果セットは次のようになります。

カラム	データ型	説明
table_qualifier	varchar(32)	データベース名。このフィールドが NULL の場合もある。
table_owner	varchar(32)	テーブル所有者。table_owner パラメータに値が指定されていない場合、この値は現在の所有者またはデータベース所有者になる。
table_name	varchar(32)	NOT NULL。
column_name	varchar(32)	NOT NULL。
key_seq	smallint	NOT NULL。複数カラムで構成されたプライマリ・キー内のカラムのシーケンス番号。

- プライマリ・キーの宣言は、sp_primarykey ではなく、create table 文または alter table 文で行っておいください。
- 「**プライマリ・キー**」という用語は、テーブルの論理プライマリ・キーを指します。Adaptive Server は、論理プライマリ・キーごとにユニーク・インデックスが定義されていること、およびこのユニーク・インデックスが sp_statistics で返されることを想定しています。

パーミッション

sp_pkeys は、すべてのユーザが実行できます。

sp_server_info

説明 Adaptive Server の属性名と現在の値のリストを返します。

構文 sp_server_info [attribute_id]

パラメータ attribute_id
サーバ属性の整数 ID を指定します。

例 **例 1**

```
sp_server_info 12
attribute_id attribute_name          attribute_value
-----
12 MAX_OWNER_NAME_LENGTH 0
```

例 2 必須ローに記述されているサーバ属性とその値のリストを返します。

```
sp_server_info
```

使用法 sp_server_info の結果セットは次のようになります。

カラム	データ型	説明
attribute_id	int	NOT NULL。
attribute_name	varchar(60)	NOT NULL。
attribute_value	varchar(255)	

sp_server_info によって返される結果セットの必須ロー

ID	サーバ属性名	説明	値
1	DBMS_NAME	DBMS の名前。	SQL SERVER
2	DBMS_VER	ODBC ドライバがバージョンの互換性を判断するときに使用する値。 注意 DBMS_VER の値は変更しないでください。これは @@version とは異なり、ODBC ドライバが使用する値と一致する必要があります。	実際の値
6	DBE_NAME	未使用。	
10	OWNER_TERM	テーブル所有者を表す Adaptive Server の表現 (名前を構成する 3 要素のうちの第 2 要素)。	owner
11	TABLE_TERM	テーブルを表す Adaptive Server の表現 (名前を構成する 3 要素のうちの第 3 要素)。	table
12	MAX_OWNER_NAME_LENGTH	テーブル所有者の名前 (名前を構成する 3 要素のうちの第 2 要素) の最大長。	30

ID	サーバ属性名	説明	値
13	TABLE_LENGTH	テーブル名の最大文字数。	30
14	MAX_QUAL_LENGTH	テーブル修飾子の名前(名前を構成する3要素のうちの第1要素)の最大長。	30
15	COLUMN_LENGTH	カラム名の最大文字数。	30
16	IDENTIFIER_CASE	データベース内でのユーザ定義名(テーブル名、カラム名、ストアド・プロシージャ名)の大文字/小文字の区別(これらのオブジェクトはこの区別に従ってシステム・カタログ内で定義されている)。	MIXED
18	COLLATION_SEQ	このサーバに対する、文字セットの想定順序。	
19	SAVEPOINT_SUPPORT	基本となる DBMS が名前付きセーブポイントをサポートするかどうか。	Y
20	MULTI_RESULT_SETS	基本となる DBMS またはゲートウェイ自体が複数の結果セットをサポートしているかどうか(ゲートウェイを介して複数の文を送信し、複数の結果セットをクライアントに返すことができるかどうか)。	Y
22	ACCESSIBLE_TABLES	sp_tables で、現在のユーザ(少なくともテーブルの select 権限を持っているユーザ)がアクセスできるテーブル、ビューなどのオブジェクトだけをゲートウェイが返すかどうか。	Y
100	USERID_LENGTH	ユーザ名の最大文字数。	30
101	QUALIFIER_TERM	テーブル修飾子を表す Adaptive Server の表現(名前を構成する3要素のうちの第1要素)。	database
102	NAMED_TRANSACTIONS	基本となる DBMS が名前付きのトランザクションをサポートするかどうか。	Y
103	SPROC_AS_LANGUAGE	ストアド・プロシージャを言語イベントとして実行できるかどうか。	Y
103	REMOTE_SPROC	DB-Library 内のリモート・ストアド・プロシージャ API を使用して、ストアド・プロシージャが実行できるかどうか。	Y
104	ACCESSIBLE_SPROC	sp_stored_procedures で、現在のユーザが実行できるストアド・プロシージャだけをゲートウェイが返すかどうか。	Y
105	MAX_INDEX_COLS	DBMS 用のインデックス内の最大カラム数。	32
106	RENAME_TABLE	テーブルの名前を変更できるかどうか。	Y
107	RENAME_COLUMN	カラムの名前を変更できるかどうか。	Y
108	DROP_COLUMN	カラムを削除できるかどうか。	Y
109	INCREASE_COLUMN_LENGTH	カラムのサイズを増やせるかどうか。	N

sp_server_info

ID	サーバ属性名	説明	値
110	DDL_IN_TRANSACTION	DDL 文をトランザクション内で実行できるかどうか。	Y
111	DESCENDING_INDEXES	降順インデックスがサポートされるかどうか。	Y
112	SP_RENAME	ストアド・プロシージャの名前を変更できるかどうか。	Y
500	SYS_SPROC_VERSION	カタログ・ストアド・プロシージャのバージョンが現在実装されているかどうか。	01.01.282 2

パーミッション

sp_server_info は、すべてのユーザが実行できます。

sp_special_columns

説明 テーブルまたはビュー内のローをユニークに識別する最適カラム・セットを返します。また、トランザクションによってロー内の値が更新されると自動的に値が生成される `timestamp` カラムのリストを返します。

構文 `sp_special_columns table_name [, table_owner] [, table_qualifier] [, col_type]`

パラメータ

table_name

テーブルまたはビューの名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。

table_owner

テーブルまたはビューの所有者の名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。テーブル所有者を指定しなければ、`sp_special_columns` は、現在のユーザが所有しているテーブルを検索してから、データベース所有者が所有しているテーブルを検索します。

table_qualifier

データベースの名前を指定します。現在のデータベースまたは `NULL` のどちらかを指定できます。

col_type

`R` を指定すると、テーブル内のローをユニークに識別する値を持つカラムに関する情報が返されます。また、`V` を指定すると、`timestamp` カラムに関する情報が返されます。このカラムの値は、ローを挿入、更新するたびに `Adaptive Server` が生成します。

例

例 1 `systypes` の最適なカラム・セットを返します。

```
sp_special_columns systypes
scope  column_name  data_type  type_name  precision  length  scale
-----  -
0      name              12         varchar    30         30      NULL
```

例 2 テーブルのローをユニークに識別する値を含む最適なセットを `authors` テーブルから返します。

```
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
```

使用法

- 結果セットは次のようになります。

カラム	データ型	説明
scope	int	NOT NULL。ロー ID の実際のスコープ。
column_name	varchar(30)	NOT NULL。カラム識別子。
data_type	smallint	ODBC データ型用の整数コード。このデータ型を ANSI/ISO 型にマップできない場合、この値は NULL になる。ネイティブのデータ型名は、type_name カラム内に返される。(ODBC データ型 表 2-2 を参照)。
type_name	varchar(13)	データ型の文字列表示。これは基本となる DBMS によって表示されているとおりのデータ型名。
precision	int	有効桁数。
length	int	データ型のバイト長。
scale	smallint	小数点以下の桁数。

パーミッション

sp_special_columns は、すべてのユーザが実行できます。

sp_sproc_columns

説明	ストアド・プロシージャの入力パラメータとリターン・パラメータについての情報を返します。
構文	<pre>sp_sproc_columns procedure_name [, procedure_owner] [, procedure_qualifier] [, column_name]</pre>
パラメータ	<p><i>procedure_name</i> ストアド・プロシージャの名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。</p> <p><i>procedure_owner</i> ストアド・プロシージャの所有者を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。所有者が指定されなければ、<code>sp_sproc_columns</code> がすべてのカラムを返します。</p> <p><i>procedure_qualifier</i> データベースの名前を指定します。現在のデータベースまたは NULL のどちらかを指定できます。</p> <p><i>column_name</i> 情報を要求するパラメータの名前を指定します。パラメータ名を指定しなければ、<code>sp_sproc_columns</code> は、すべての入力情報を返し、ストアド・プロシージャのリターン・パラメータを返します。</p>

使用法

- `sp_sproc_columns` の結果セットは次のようになります。

カラム	データ型	説明
<code>procedure_qualifier</code>	<code>varchar(30)</code>	プロシージャ修飾子名。NULL の場合もある。
<code>procedure_owner</code>	<code>varchar(30)</code>	プロシージャの所有者名。常に値が返される。
<code>procedure_name</code>	<code>varchar(41)</code>	プロシージャ名。常に値が返される。
<code>column_name</code>	<code>varchar(30)</code>	返された <code>table_name</code> の各カラムのカラム名。常に値が返される。
<code>column_type</code>	<code>smallint</code>	
<code>data_type</code>	<code>smallint</code>	ODBC データ型用の整数コード。このデータ型を ANSI/ISO 型にマップできない場合、この値は NULL になる。ネイティブのデータ型名は、 <code>type_name</code> カラム内に返される。
<code>type_name</code>	<code>char(30)</code>	データ型の文字列表示。これは基本となる DBMS によって表示されているとおりのデータ型名。
<code>precision</code>	<code>int</code>	有効桁数。
<code>length</code>	<code>int</code>	データ型のバイト長。
<code>scale</code>	<code>smallint</code>	小数点以下の桁数。
<code>radix</code>	<code>smallint</code>	数値データ型の基底データ型。

カラム	データ型	説明
nullable	smallint	1 は、このデータ型を作成すると null 値が許容されることを示す。0 はそうでないことを示す。
remarks	varchar(254)	プロシージャ・カラムの記述。NULL。
ss_data_type	tinyint	Adaptive Server のデータ型。
colid	tinyint	syscolumns のカラム ID。
column_def	varchar(255)	NULL。
sql_data_type	smallint	Adaptive Server のデータ型。
sql_datetime_sub	smallint	NULL。
char_octet_length	int	char_octet_length の値は、char_octet_length が次のデータ型のときに precision カラムの値と同じです。 <ul style="list-style-type: none"> • binary • char • image • nchar • nvarchar • sysname • text • timestamp • varbinary • varchar これ以外の場合、char_octet_length の値は 0 になる。
ordinal_position	int	パラメータ・リスト内のパラメータの位置を序数で表したものの。リストの最初のパラメータは 1 で、戻り値には序数が含まれる。
is_nullable	varchar(3)	カラムまたはパラメータに値として NULL を使用できるかどうかを示す。syscolumns から取得。
mode	varchar(20)	syscolumns に格納されているパラメータ・モード情報。次の内容を含む。 <ul style="list-style-type: none"> • <i>SQL</i> プロシージャの場合 - in、out、または「戻り値」。 • <i>SQLJ</i> プロシージャ (Java) の場合 - in、out、inout、または「戻り値」。

- sp_sproc_columns は、double precision に定義されたパラメータに対して、type_name を float 値としてレポートし、data_type を 6 としてレポートします。Adaptive Server の double precision データ型は、ODBC 仕様で指定されている値の範囲をサポートする float 実装データ型です。

パーミッション

sp_sproc_columns は、すべてのユーザが実行できます。

sp_statistics

説明	1つのテーブルのインデックス・リストを返します。
構文	<code>sp_statistics table_name [, table_owner] [, table_qualifier] [, index_name] [, is_unique]</code>
パラメータ	<p>table_name テーブルの名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。</p> <p>table_owner テーブルの所有者を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。<i>table_owner</i>を指定しなければ、<code>sp_statistics</code>は、現在のユーザが所有しているテーブルを検索してから、データベース所有者が所有しているテーブルを検索します。</p> <p>table_qualifier データベースの名前を指定します。現在のデータベースまたはNULLのどちらかを指定できます。</p> <p>index_name インデックス名を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。</p> <p>is_unique Yを指定すると、ユニーク・インデックスだけを返します。Nを指定すると、ユニーク・インデックスと非ユニーク・インデックスの両方を返します。</p>

```

例
      sp_statistics publishers

table_qualifier      table_owner
table_name           non_unique
index_qualifier      index_name
type    seq_in_index column_name           collation
cardinality pages
-----
-----
-----
-----
-----
pubs2                dbo
publishers           NULL
NULL                NULL
      0              NULL NULL              NULL
      3              1
  
```

```

pubs2                                dbo
publishers                            0
publishers                            pubind
1                                     1 pub_id                                A
3                                     1
    
```

使用法

- sp_statistics の結果セットは次のようになります。

カラム	データ型	説明
table_qualifier	varchar(32)	データベース名。このフィールドが NULL の場合もある。
table_owner	varchar(32)	
table_name	varchar(32)	NOT NULL。
non_unique	smallint	NOT NULL。0 はユニークであることを示し、1 はユニークでないことを示す。
index_qualifier	varchar(32)	
index_name	varchar(32)	
type	smallint	NOT NULL。0 はクラスタード、2 はハッシュ、3 はその他を示す。
seq_in_index	smallint	NOT NULL。
column_name	varchar(32)	NOT NULL。
collation	char(1)	A は昇順、D は降順を示す。NULL は適用できないことを示す。
cardinality	int	テーブル内のロー数またはインデックス内のユニーク値。
pages	int	インデックスまたはテーブルを保管するページ数。

- 結果セット内のインデックスは、non-unique、type、index_name、seq_in_index カラムの昇順で表示されます。
- hashed インデックス・タイプは、正確な一致検索、または範囲検索を受け入れます。ただし、パターン一致を含む検索はインデックスを使用しません。

パーミッション

sp_statistics は、すべてのユーザが実行できます。

sp_stored_procedures

説明 1つまたは複数のストアド・プロシージャの情報を返します。

構文 sp_stored_procedures [sp_name [, sp_owner [, sp_qualifier]]]

パラメータ

sp_name

ストアド・プロシージャの名前を指定します。複数のストアド・プロシージャの情報を要求するには、ワイルドカード文字を使用します。

sp_owner

ストアド・プロシージャの所有者を指定します。複数のユーザが所有しているプロシージャの情報を要求するには、ワイルドカード文字を使用します。

sp_qualifier

データベースの名前を指定します。現在のデータベースまたはNULLを指定できます。

使用法

- sp_stored_procedures は、現在のデータベース内だけにあるストアド・プロシージャの情報を返します。
- sp_stored_procedures の結果セットは次のようになります。

カラム	データ型	説明
procedure_qualifier	varchar(30)	データベースの名前。
procedure_owner	varchar(30)	
procedure_name	varchar(41)	NOT NULL。
num_input_params	int	NOT NULL。常に -1 を返す。
num_output_params	int	NOT NULL。値 ≥ 0 は、パラメータの数を示す。-1 は、パラメータの数が不確定であることを示す。
num_result_sets	int	NOT NULL。常に -1 を返す。
remarks	varchar(254)	NULL。

- sp_stored_procedures は、現在のユーザが実行パーミッションを持っていないストアド・プロシージャ名を返すことができます。ただし、sp_server_info の結果セットでサーバ属性の accessible_sproc が “Y” になっている場合は、現在のユーザが実行できるストアド・プロシージャだけが返されます。

パーミッション

sp_stored_procedures は、すべてのユーザが実行できます。

sp_table_privileges

説明	テーブルまたはビューのすべてのカラムの権限情報を返します。
構文	<code>sp_table_privileges table_name [, table_owner[, table_qualifier]]</code>
パラメータ	<p><i>table_name</i> テーブルの名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。</p> <p><i>table_owner</i> テーブル所有者の名前を指定します。ワイルドカード文字を使用したパターン一致機能はサポートされていません。テーブル所有者を指定しなければ、<code>sp_table_privileges</code> は、現在のユーザが所有しているテーブルを検索してから、データベース所有者が所有しているテーブルを検索します。</p> <p><i>table_qualifier</i> データベースの名前を指定します。現在のデータベースまたは NULL のどちらかを指定できます。</p>
使用法	<ul style="list-style-type: none"> • <code>sp_table_privileges</code> の結果セットは次のようになります。

カラム	データ型	説明
table_qualifier	varchar(32)	データベースの名前。このフィールドが NULL の場合もある。
table_owner	varchar(32)	
table_name	varchar(32)	NOT NULL。
grantor	varchar(32)	NOT NULL。
grantee	varchar(32)	NOT NULL。
privilege	varchar(32)	テーブルの権限を識別する。権限は、次のうちのいずれか1つ。 <ul style="list-style-type: none"> • SELECT — 付与されたユーザは、テーブルの1つまたは複数のカラムのデータを検索できる。 • INSERT — 付与されたユーザは、データを含むローを挿入できる。 • UPDATE — 付与されたユーザは、テーブルの1つまたは複数のカラムのデータを更新できる。 • DELETE — 付与されたユーザは、テーブルからデータのローを削除できる。 • REFERENCE — 付与されたユーザは、制約の範囲内でテーブルの1つまたは複数のカラムを参照できる。
is_grantable	varchar(3)	付与されたユーザが、他のユーザに権限を付与できるかどうかを示す。値は YES、NO、NULL。

パーミッション `sp_table_privileges` は、すべてのユーザが実行できます。

sp_tables

説明	from 句内に表示できるオブジェクトのリストを返します。
構文	sp_tables [table_name] [, table_owner][, table_qualifier][, table_type]
パラメータ	<p>table_name テーブルの名前を指定します。複数のテーブルの情報を要求するには、ワイルドカード文字を使用します。</p> <p>table_owner テーブル所有者を指定します。複数のテーブルの情報を要求するには、ワイルドカード文字を使用します。</p> <p>table_qualifier データベースの名前を指定します。現在のデータベースの名前または NULL を指定できます。</p> <p>table_type カンマで複数の値を区切ったリストです。これによって、そのようなテーブル・タイプ (1 つまたは複数) のすべてのテーブルに関する情報を表示するように指定できます。次の例のように指定します。</p> <pre>" 'TABLE', 'SYSTEM TABLE', 'VIEW' "</pre>

注意 テーブル・タイプは一重引用符で囲み、パラメータ全体を二重引用符で囲んでください。テーブル・タイプは大文字で入力してください。

例

```
sp_tables @table_type = 'TABLE', 'VIEW'
```

このプロシージャは、現在のデータベースから TABLE と VIEW のタイプを持つすべてのテーブルの情報を返します。システム・テーブルの情報は返しません。

- 使用法
- Adaptive Server は、*table_name* への読み取りおよび書き込みパーミッションを確認しません。テーブルに関する情報を表示できても、そのテーブルへのアクセスは保証されていません。
 - 結果セットには、テーブル、ビュー、シノニム (オブジェクトに対するエイリアス)、および DBMS 製品へのゲートウェイ用のエイリアスが含まれます。
 - サーバ属性 *accessible_tables* が *sp_server_info* の結果セット内で Y に設定されていると、現在のユーザがアクセスできるテーブルだけが返されます。

- sp_tables の結果セットは次のようになります。

カラム	データ型	説明
table_qualifier	varchar(30)	データベース名。このフィールドが NULL の場合もある。
table_owner	varchar(30)	
table_name	varchar(30)	NOT NULL。テーブル名。
table_type	varchar(32)	NOT NULL。'TABLE'、'VIEW'、'SYSTEM TABLE' のいずれか1つ。
remarks	varchar(254)	NULL。

パーミッション

sp_tables は、すべてのユーザが実行できます。

使用するテーブル

master.dbo.sysattributes、master.dbo.sysloginroles、
master.dbo.syssrvroles、sysroles

システム拡張ストアド・プロシージャ

この章では、Sybase が提供する拡張ストアド・プロシージャ (ESP) である、システム拡張ストアド・プロシージャについて説明します。ESP は、installmaster によって、インストール時に作成されます。作成されたシステム・プロシージャは sybssystemprocs データベースに保管され、システム管理者が所有します。このプロシージャは、どのデータベースからでも実行できます。

この章では、次の項目について説明します。

トピック名	ページ
概要	885
システム ESP へのパーミッション	886
システム ESP に対応する DLL	886
システム ESP の使用	886

概要

表 3-1 では、この章で説明するシステム拡張ストアド・プロシージャをリストしています。

表 3-1: システム拡張ストアド・プロシージャ

プロシージャ	説明	プラットフォーム
xp_cmdshell	Adaptive Server を実行するホスト・システムでネイティブなオペレーティング・システムのコマンドを実行する。	サポートされているすべての DLL
xp_deletemail	Adaptive Server のメッセージ・インボックスからメッセージを削除する。	NT のみ
xp_enumgroups	特定の Windows NT ドメインのグループを表示する	NT のみ
xp_findnextmsg	Adaptive Server メッセージ・インボックス内で、次のメッセージの識別子を取得する。	NT のみ

プロシージャ	説明	プラットフォーム
xp_logevent	Windows NT のイベント・ログにユーザ定義イベントをログイングする。	NT のみ
xp_readmail	Adaptive Server のメッセージ・インボックスからメッセージを読み込む。	NT のみ
xp_sendmail	MAPI インタフェースを使って、指定された受信者にメッセージを送信する。	NT のみ
xp_startmail	Adaptive Server のメール・セッションを開始する。	NT のみ
xp_stopmail	Adaptive Server のメール・セッションを終了する。	NT のみ

システム ESP へのパーミッション

パーミッションは、`sybserverprocs` データベース内で設定されます。

`sa_role` によって指定されたユーザには、システム ESP に対するデフォルトの実行パーミッションが割り当てられます。これらのシステム管理者は、ほかのユーザに実行パーミッションを付与できます。

システム ESP に対応する DLL

`sybserverprocs` データベースにある `sp_helpextendedproc` を実行することによって、システム ESP に対応する DLL の名前を取得できます。

システム ESP の使用

システム ESP は、通常のシステム・プロシージャと同じ呼び出しルールに従います。特別な条件は、**Open Server** アプリケーションである **XP Server** が動作していることだけです。Adaptive Server は、初めてシステム ESP が呼び出されたときに **XP Server** を起動します。Adaptive Server を停止させるまで、**XP Server** は動作し続けます。

xp_cmdshell

説明	Adaptive Server を実行するホスト・システム上で、ネイティブなオペレーティング・システムのコマンドを実行します。
構文	<code>xp_cmdshell command[, no_output] [return_status no_wait]</code>
パラメータ	<p>command オペレーティング・システム・コマンドを表す文字列で、最長 8192 バイトです。</p> <p>no_output このパラメータを指定すると、コマンドからの出力が表示されなくなります。</p> <p>return_status このパラメータを指定すると、command パラメータに指定したオペレーティング・システム・コマンドの完了ステータスが返されます。このパラメータを使用しない場合、戻り値は成功時に 0、失敗時に 1 です。</p> <p>no_wait このパラメータを指定すると、xp_cmdshell オペレーションは呼び出し元へすぐに制御を返し、指定されたコマンドはバックグラウンド・プロセスとして実行されます。出力は行われません。返される結果にはバックグラウンド・プロセス自体の成功または失敗は反映されず、プロセスの起動の成功または失敗だけが反映されます。</p>
例	<p>例 1 (Windows の場合) ドライブ C のファイル <code>log</code> を、ドライブ A のファイル <code>log.0102</code> にコピーします。実行中はコマンドからの出力を表示しません。</p> <pre>xp_cmdshell 'copy C:¥log A:¥log.0102', no_output</pre> <p>例 2 (UNIX の場合) オペレーティング・システムの <code>ls</code> コマンドを実行し、ディレクトリの内容を示すリストをデータのローとして返します。</p> <pre>xp_cmdshell 'ls'</pre>
使用法	<ul style="list-style-type: none"> • xp_cmdshell は、単一カラムにあるテキストのローとして、オペレーティング・システムのエラーも含めた出力を返します。 • xp_cmdshell は、XP Server の現在のディレクトリから実行されます。 • 返される出力のカラム幅は、80 文字です。出力はフォーマット化されません。

- xp_cmdshell は、“login”のようなユーザとの対話が必要なコマンドを実行することはできません。
- オペレーティング・システム・コマンドが xp_cmdshell を介して実行されるユーザ・コンテキストは、xp_cmdshell context 設定パラメータの値によって制御されます。このパラメータが 1 (デフォルト) に設定されると、xp_cmdshell は、システム管理者の権限を持つユーザに対してオペレーティング・システム・レベルのパーミッションを制限します。このパラメータが 0 に設定されると、xp_cmdshell は、Adaptive Server が動作しているオペレーティング・システムのアカウントのセキュリティ・コンテキストを使います。したがって、xp_cmdshell context 設定パラメータを 0 に設定して xp_cmdshell を使用すると、どのユーザも、Adaptive Server を実行しているアカウントのパーミッションを使用してオペレーティング・システムのコマンドを実行できます。このアカウントは、ユーザ所有のアカウントよりも制限が少なくなることがあります。
- xp_cmdshell context の値にかかわらず、xp_cmdshell を実行するユーザがシステム管理者ではない (sa_role を割り当てられていない) 場合、システム管理者は、そのユーザに、xp_cmdshell を実行する明示的なパーミッションを付与してください。たとえば、次の文は、xp_cmdshell を実行するパーミッションを joe に付与します。

```
grant execute on xp_cmdshell to joe
```

- xp_cmdshell が外部コマンドを XP Server で正常に生成できたかどうかを調べるには、次のコマンドを入力します。ここで、command は xp_cmdshell を使用して実行するコマンドの名前です。

```
exec @ret = xp_cmdshell command
```

xp_cmdshell が正常に動作していた場合、exec @ret = xp_cmdshell command は 0 を返します。xp_cmdshell が正常に動作しなかった場合、exec @ret = xp_cmdshell command は 1 を返します。

- xp_cmdshell を使用して実行したコマンドそのものが正常に動作したかどうかを調べるには、次の文を入力します。ここで、command は xp_cmdshell を使用して実行するコマンドの名前です。

```
exec @ret = xp_cmdshell command, return_status
```

`exec @ret = xp_cmdshell command, return_status` を実行すると、`xp_cmdshell` はコマンドの実際の終了ステータス・コードを返します。エラーが発生して **XP Server** がコマンドを実行できない場合、`xp_cmdshell` は 1 を返します。コマンドが正常に動作した場合、`xp_cmdshell` は 0 を返します。

コマンドが正常に終了した場合、`exec @ret = xp_cmdshell command` は 0 を返します。正常に終了しなかった場合、`exec @ret = xp_cmdshell command` は 1 を返します。

注意 `exec @ret = xp_cmdshell command` と `exec @ret = xp_cmdshell command, return_status` には下位互換性があります。`return_status` パラメータを使用しない古いストアド・プロシージャは、`exec @ret = xp_cmdshell command, return_status` を `exec @ret = xp_cmdshell command` と同じように処理します。

`return_status` は、`no_output` パラメータと一緒に使用できます。指定する順序は問いません。

- `xp_cmdshell` リモート・プロシージャを使用してプロキシ・テーブルを作成するときは、`cmdstr` カラム名を使用してください。

```
create existing table xpoutput
(
    cmdstr varchar(255) null
)
external procedure at "THIS...xp_cmdshell"

select cmdstr from xpoutput where cmdstr = "date"
```

`cmdstr` を使用しないと、エラー・メッセージが表示されます。

プロキシ・テーブルから返される結果の詳細については、『コンポーネント統合サービス・ユーザズ・ガイド』の「プロキシ・テーブルとしてのリモート・プロシージャ」を参照してください。

パーミッション

デフォルトでは、`xp_cmdshell` を実行できるのはシステム管理者だけです。システム管理者は、実行パーミッションをほかのユーザに付与できます。

参照

`xp_cmdshell context` の詳細については、『システム管理ガイド』を参照してください。

xp_deletemail

説明	(Windows のみ) Adaptive Server のメッセージ・インボックスからメッセージを削除します。
構文	xp_deletemail [<i>msg_id</i>]
パラメータ	<i>msg_id</i> 削除するメール・メッセージのメッセージ識別子です。
例	<p>例 1 Adaptive Server のメッセージ・インボックスから、<i>cur_msg_id</i> 変数で指定したメッセージ識別子を持つメッセージを削除します。</p> <pre>1> declare @cur_msg_id binary(255) 2> exec xp_deletemail @msg_id = @cur_msg_id</pre> <p>例 2 Adaptive Server のメッセージ・インボックスから最初のメッセージを削除します。</p> <pre>xp_deletemail</pre>
使用法	<ul style="list-style-type: none">• xp_findnextmsg を使用して、<i>msg_id</i> パラメータを取得します。• <i>msg_id</i> パラメータを使用しなければ、デフォルトでメッセージ・インボックス内にある最初のメッセージが削除されます。
パーミッション	デフォルトでは、xp_deletemail を実行できるのはシステム管理者だけです。システム管理者は、このパーミッションをほかのユーザに付与できます。

xp_enumgroups

説明	(Windows のみ) 指定の Windows NT ドメインのグループを表示します。
構文	<code>xp_enumgroups [domain_name]</code>
パラメータ	<i>domain_name</i> ユーザ・グループのリスト作成に使用する Windows NT ドメインの名前です。
例	例 1 XP Server を実行する Windows NT コンピュータ上のすべてのユーザ・グループをリストします。 <pre>xp_enumgroups</pre> 例 2 PCS ドメイン内のすべてのユーザ・グループをリストします。 <pre>xp_enumgroups 'PCS'</pre>
使用法	<ul style="list-style-type: none">パラメータが渡されない場合、<code>xp_enumgroups</code> は、すべてのローカル・ユーザ・グループを表示します。「ドメイン」とは、共通ユーザ・アカウント・データベースとセキュリティ方式を共有する、コンピュータの名前付き集合体をいいます。リターン・ステータスが 0 であれば成功、1 であれば失敗を示します。
パーミッション	デフォルトでは、 <code>xp_enumgroups</code> を実行できるのはシステム管理者だけです。システム管理者は、このパーミッションをほかのユーザに付与できます。

xp_findnextmsg

説明	(Windows のみ) Adaptive Server のメッセージ・インボックスから次のメッセージの識別子を取得します。
構文	<code>xp_findnextmsg @msg_id = @msg_id output[, type] [, unread_only = {true false}]</code>
パラメータ	<p>msg_id 入力時、検索するメッセージ識別子のすぐ前にあるメッセージ識別子を指定します。検索したメッセージ識別子は <i>msg_id</i> 出力パラメータ内に設定されますが、これはバイナリ型でなければなりません。</p> <p>type MAPI メール定義を基本にした入力メッセージ・タイプです。サポートされるメッセージ・タイプは CMC:IPM だけです。NULL 値または値なしで指定すると、CMC:IPM がデフォルトになります。</p> <p>unread_only このパラメータを true に設定すると、<code>xp_findnextmsg</code> は未読のメッセージだけを処理します。このパラメータが false に設定されると、<code>xp_findnextmsg</code> は、指定されたメッセージの次にあるメッセージのメッセージ識別子を検索するときに、既読、未読両方のメッセージすべてを対象にします。デフォルトは true です。</p>
例	<p>例 1 <code>@out_msg_id</code> で指定されたメッセージの次にある未読メッセージのメッセージ識別子を、<code>@out_msg_id</code> 出力変数に返します。</p> <pre>xp_findnextmsg @msg_id = @out_msg_id output</pre> <p>例 2 <code>@out_msg_id</code> で指定されたメッセージの次にあるメッセージのメッセージ識別子を、<code>@out_msg_id</code> 出力変数に返します。メッセージは既読の場合もあれば、未読の場合もあります。</p> <pre>xp_findnextmsg @msg_id = @out_msg_id output, NULL, @unread_only = false</pre>
使用法	<ul style="list-style-type: none"> インボックス内でもうメッセージが見つからなければ、<code>xp_findnextmsg</code> は 1 のステータスを返します。 <code>xp_deleteemail</code> と <code>xp_readmail</code> は、<code>xp_findnextmsg</code> から返されたメッセージ識別子を使用します。
パーミッション	デフォルトでは、 <code>xp_findnextmsg</code> を実行できるのはシステム管理者だけです。システム管理者は、このパーミッションをほかのユーザに付与できます。

xp_logevent

説明	(Windows のみ) Adaptive Server 内から Windows NT のイベント・ログにユーザ定義イベントをロギングします。												
構文	<code>xp_logevent error_number, message[, type]</code>												
パラメータ	<p><i>error_number</i> ユーザが割り当てるエラー番号です。50000 以上にしてください。</p> <p><i>message</i> イベント・ビューワの記述フィールドに表示されるメッセージのテキストです。メッセージの最長は 255 バイトです。メッセージは引用符で囲んでください。</p> <p><i>type</i> イベントの緊急度を表します。有効な値は、informational、warning、error です。デフォルトは informational です。値は引用符で囲んでください。</p>												
例	<p>例 1 番号 55555 の情報イベントは、Windows NT イベント・ログにログされます。イベント詳細ウィンドウ内の記述テキストは、“Email message deleted” です。</p> <pre>xp_logevent 55555, 'Email message deleted.'</pre> <p>例 2 番号 66666 のエラー・イベントは、Windows NT イベント・ログにログされます。イベント詳細ウィンドウ内の記述テキストは、“DLL not found” です。</p> <pre>xp_logevent 66666, 'DLL not found.', 'error'</pre>												
使用法	次の表は、 xp_logevent によって生成されるイベントのデフォルトのイベント詳細を示します。												
	<table border="1"> <thead> <tr> <th>詳細</th> <th>値</th> </tr> </thead> <tbody> <tr> <td>ユーザ</td> <td>なし</td> </tr> <tr> <td>コンピュータ</td> <td>XP Server を実行するマシンの名前</td> </tr> <tr> <td>イベント ID</td> <td>12</td> </tr> <tr> <td>ソース</td> <td>Adaptive Server の名前</td> </tr> <tr> <td>カテゴリ</td> <td>User</td> </tr> </tbody> </table>	詳細	値	ユーザ	なし	コンピュータ	XP Server を実行するマシンの名前	イベント ID	12	ソース	Adaptive Server の名前	カテゴリ	User
詳細	値												
ユーザ	なし												
コンピュータ	XP Server を実行するマシンの名前												
イベント ID	12												
ソース	Adaptive Server の名前												
カテゴリ	User												
パーミッション	xp_logevent を実行できるのは、システム管理者だけです。												

xp_readmail

説明 (Windows のみ) Adaptive Server のメッセージ・インボックスからメッセージを読み込みます。

構文

```
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]]
```

パラメータ

msg_id

xp_readmail で読み込むメッセージのメッセージ識別子を指定します。*msg_id* パラメータが指定されなかった場合、unread が true であればメッセージ・ボックスにある最初の未読メッセージが、unread が false であればメッセージ・ボックスにある最初のメッセージが、それぞれデフォルトで読み込まれます。

recipients

セミコロンで区切られたメッセージ受信者のリストです。

sender

メッセージの送信側です。

date_received

メッセージを受信した日付です。

subject

メッセージのサブジェクト・ヘッダです。

cc

cc (カーボン・コピー) によって同時送信したメッセージを受け取る受信者のリストです (セミコロンで区切られています)。

message

メッセージ本文のテキストです。*msg_length* 出力パラメータにはメッセージの本文の長さが返されますが、その長さが 255 バイトを超える場合は、*byte_to_skip* と *msg_length* パラメータを使用して、255 バイトずつメッセージを読み込みます。

attachments

添付ファイルのテンポラリ・パスのリストです(セミコロンで区切られています)。suppress_attach が true の場合、attachments は無視されます。

suppress_attach

true に設定すると、添付ファイル用のテンポラリ・ファイルは作成されません。デフォルトは true です。

peek

false に設定すると、メッセージが読み込まれたあと、既読を示すフラグをメッセージに付けます。true に設定すると、メッセージが読み込まれたあとでも未読を示すフラグをメッセージに付けます。デフォルトは false です。

unread_only

true に設定すると、xp_readmail は未読メッセージだけを読み込みます。false に設定すると、メッセージに付いているフラグが既読であろうと未読であろうと、xp_readmail はすべてのメッセージを対象にします。デフォルトは true です。

msg_length

メッセージの全長です(バイト単位)。byte_to_skip パラメータと一緒に使用すると、xp_readmail で 255 バイトずつメッセージを読むことができます。

bytes_to_skip

入力時は、0 以外の値にすると、メッセージの次の 255 バイトを message 出力パラメータに読み込む前に読み飛ばすバイト数を指定します。出力時には、次の 255 バイトの読み込みを開始するメッセージのオフセット値 (byte_to_skip の元の値に、今回の呼び出しで出力された msg_length を加算した値) が出力されます。

type

MAPI メール定義に基づくメッセージ・タイプです。サポートされるメッセージ・タイプは CMC:IPM だけです。NULL 値または値なしで指定すると、CMC:IPM がデフォルトになります。

例

例 1 xp_readmail は、メッセージ・インボックスにある最初の未読メッセージを読み込みます。このメッセージのメッセージ識別子を @msgid 変数から取得します。この変数には xp_findnextmsg ESP によって値が格納されています。xp_readmail は、送信側の名前を @originator 変数に、メッセージ本文を @mess 変数に、それぞれ格納します。

```
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
```

例 2 xp_findnextmsg が出力するメッセージ識別子のメッセージの最初の 255 バイトを読み込みます。メッセージの全長が 255 バイトを超えた場合、次の 255 バイトを読み込み、読み込むバイトがなくなるまで繰り返し読み込みます。

```
declare @msgid binary(255)
declare @mess varchar(255)
declare @msg_length char(255)
declare @len int
declare @skip int
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
```

使用法

- xp_readmail は、Adaptive Server のメッセージ・インボックスからメッセージを読み込みます。
- メッセージ・インボックス内で、現在読み込んでいるメッセージの次にあるメッセージの識別子を取得するには、[xp_findnextmsg](#) を使用します。

パーミッション

デフォルトでは、xp_readmail を実行できるのはシステム管理者だけです。システム管理者は、このパーミッションをほかのユーザに付与できます。

xp_sendmail

説明

(Windows のみ) 指定した受信者にメッセージを送信します。メッセージはテキストまたは Transact-SQL クエリの結果のどちらかです。

構文

```
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}]
    [, no_output = {true | false}]
    [, width]
    [, separator]
    [, dbuser]
    [, dbname]
    [, type]
    [, include_query = {true | false}]
```

パラメータ

recipient

メッセージを受信するユーザの電子メール・アドレスです。少なくとも一人の受信者が必要です。複数の受信者はセミコロンで区切ります。

subject

メッセージの省略可能なサブジェクト・ヘッダです。このパラメータを使用しなければ、“Sybase SQL Server Message” がデフォルトのサブジェクトになります。

cc_recipient

cc (カーボン・コピー) によって同時送信したメッセージを受け取る受信者のリストです (セミコロンで区切られています)。

bcc_recipient

bcc (ブラインド・カーボン・コピー) によって同時送信したメッセージを受け取る受信者のリストです (セミコロンで区切られています)。

query

1 つ以上の Transact-SQL 文です。その結果はメッセージ受信者に送信されます。 *query* を使用すると、 *message* は使用できません。

message

送信するメッセージのテキストです。*message* を使用すると、*query* は使用できません。*message* パラメータを使用したときに無視されるオプションの詳細なリストについては、「使用法」の項を参照してください。

attachname

query パラメータを使用した場合にクエリの結果が格納されるファイルの名前です。このファイルはメッセージに添付されます。*attachname* を使用するときは、*attach_result* を *true* に設定します。*attach_result* が *true* で、*attachname* が指定されていない場合、生成される添付ファイル名のプレフィックスは “*syb*” になり、そのあとに 5 つのランダムな数字、さらに “.*txt*” 拡張子が続きます。たとえば *syb84840.txt* のようになります。*message* パラメータを使用すると、このパラメータは無視されます。

attach_result

true に設定すると、クエリの結果をメッセージへの添付ファイルとして送信します。*false* に設定すると、メッセージの本文部内に直接結果を渡します。デフォルトは *false* です。*message* パラメータを使用すると、このパラメータは無視されます。

echo_error

true に設定すると、クエリの結果とともに、影響を受けたロー数を示すメッセージなどの Adaptive Server メッセージを送信します。*false* に設定すると、Adaptive Server メッセージは送信されません。デフォルトは *true* です。*message* パラメータを使用すると、このパラメータは無視されます。

include_file

メッセージの添付ファイルとして組み込まれるファイルのリストです。セミコロンで区切られています。ファイルにはファイル名、パス名、または相対パス名を指定できます。また、指定できるファイルはテキストまたはバイナリのいずれかです。

no_column_header

true に設定すると、カラム・ヘッダがクエリの結果と一緒に送信されます。*false* に設定すると、カラム・ヘッダは送信されません。デフォルトは *false* です。*message* パラメータを使用すると、このパラメータは無視されます。

no_output

true に設定すると、メールを送信したセッションには出力が送信されません。**false** に設定すると、メールを送っているセッションが出力を受信します。デフォルトは **false** です。**message** パラメータを使用すると、このパラメータは無視されます。

width

メッセージでクエリの結果を送信するときの結果セットの幅を文字数で指定します。**width** パラメータは、**isql** の **-w** オプションと同じです。指定した **width** 値に達すると、結果ローが改行文字によって折り返されます。デフォルトでは、80 文字です。**message** パラメータを使用すると、このパラメータは無視されます。

separator

メッセージでクエリの結果を送信するときのカラム・セパレータとして使用する文字を指定します。**separator** は、**isql** の **-s** オプションと同じです。デフォルトはタブ文字です。**message** パラメータを使用すると、このパラメータは無視されます。

dbuser

query パラメータが使用されたときに、このパラメータで指定したデータベース・ユーザのユーザ・コンテキストでクエリを実行します。デフォルトは “**guest**” です。**message** パラメータを使用すると、このパラメータは無視されます。

dbname

query パラメータが使用されたときに、このパラメータで指定したデータベースのデータベース・コンテキストでクエリを実行します。デフォルトは “**master**” です。**message** パラメータを使用すると、このパラメータは無視されます。

type

MAPI メール定義を基本にした入力メッセージ・タイプです。サポートされるメッセージ・タイプは **CMC:IPM** だけです。**NULL** 値または値なしで指定すると、**CMC:IPM** がデフォルトになります。

include_query

true に設定すると、**query** パラメータ内で使用されているクエリ (1 つまたは複数) が結果セットに追加されます。**false** に設定すると、クエリは追加されません。デフォルトは **false** です。**message** パラメータを使用すると、**include_query** は無視されます。

例 **例 1** `xp_sendmail` は、Adaptive Server のバックアップ・ステータスに関するテキスト・メッセージを、“admin” グループへの cc とともに、“sally” と “ramon” に送信します。

```
xp_sendmail @recipient = "sally;ramon",
@subject = "Adaptive Server Backup Status",
@message = "Adaptive Server Backup for SERVER2 is
complete.",
@copy_recipient="admin"
```

例 2 `authors` テーブルに関するクエリ結果を `peter` に送信します。クエリ結果は、`au_lis.res` という名前の添付ファイルとしてメッセージに添付されます。このファイルは、サーバが実行されたディレクトリ内に置かれます。

```
xp_sendmail "peter",
@query = "select * from authors",
@attachname = "au_list.res",
@attach_result= true
```

使用法

- `attachname`、`attach_result`、`echo_error`、`no_column_header`、`no_output`、`width`、`separator`、`dbuser`、`dname`、`include_query` の各パラメータは、`query` パラメータが使用されたときにメッセージで送信されるクエリの結果に関するパラメータです。これらのパラメータは、`message` パラメータが使用されたときは無視されます。

パーミッション

デフォルトでは、`xp_sendmail` を実行できるのはシステム管理者だけです。システム管理者は、このパーミッションをほかのユーザに付与できます。

xp_startmail

説明	(Windows のみ) Adaptive Server のメール・セッションを開始します。
構文	<code>xp_startmail [mail_user] [, mail_password]</code>
パラメータ	<p><i>mail_user</i></p> <p>Windows NT のメール・システムにログインするために Adaptive Server で使用されるメール・プロファイル名です。<i>mail_user</i> が使用されなければ、<code>xp_startmail</code> は、Sybmail の Adaptive Server アカウントをセットアップするために使われたメール・ユーザ名を使用します。</p> <p><i>mail_password</i></p> <p>Windows NT のメール・システムにログインするために Adaptive Server で使用されるメール・パスワードです。<i>mail_password</i> が使用されなければ、<code>xp_startmail</code> は、Sybmail の Adaptive Server アカウントをセットアップするために使われたメール・パスワードを使用します。</p>
例	<p>例 1 Sybmail のユーザ・アカウント用のメール・ユーザ名とパスワードを使用して、Adaptive Server のメール・セッションを開始します。</p> <pre>xp_startmail</pre> <p>例 2 プロファイル名 “mailuser” とこれに対応するパスワードを使用して Adaptive Server メール・セッションを開始します。</p> <pre>xp_startmail "mailuser", "tre55uu"</pre>
使用法	<ul style="list-style-type: none">Adaptive Server メール・セッションがすでに実行されている場合、<code>xp_startmail</code> はメール・セッションを開始しません。Sybmail 関連のシステム ESP または <code>sp_processmail</code> ストアド・プロシージャを実行するには、<code>xp_startmail</code> を明示的に呼び出すか、Adaptive Server の起動時に Adaptive Server メール・セッションを自動的に開始するように設定して、Adaptive Server メール・セッションを開始してください。Adaptive Server メール・セッションを Adaptive Server の起動時に自動的に開始する方法については、『システム管理ガイド』の <code>start mail session</code> を参照してください。Windows NT の automail セッションがオンになっていない場合は、<code>xp_startmail</code> で、<i>mail_user</i> パラメータと <i>mail_password</i> パラメータを使用してください。

- ユーザ・アカウント sybmail の *fullname* フィールドから *mail_user* のデフォルト値を確認するには、次のように [sp_displaylogin](#) システム・プロシージャを使用します。

```
sp_displaylogin sybmail
```

パーミッション

デフォルトでは、`xp_startmail` を実行できるのはシステム管理者だけです。システム管理者は、このパーミッションをほかのユーザーに付与できます。

xp_stopmail

説明	(Windows のみ) Adaptive Server のメール・セッションを停止します。
構文	xp_stopmail
パラメータ	なし
例	Adaptive Server のメール・セッションを停止します。 <pre>xp_stopmail</pre>
使用法	Adaptive Server のメール・セッションが xp_stopmail によって停止したあとは、Sybmail 関連のシステム ESP と sp_processmail ストアド・プロシージャは実行できません。
パーミッション	デフォルトでは、xp_stopmail を実行できるのはシステム管理者だけです。システム管理者は、このパーミッションをほかのユーザに付与できます。

この章では、dbcc ストアド・プロシージャについて説明します。

この章では、次の項目について説明します。

トピック名	ページ
概要	905
オブジェクトの名前と日付の指定	906

概要

このプロシージャは、dbccdb データベースまたは代替データベース dbccalt にあるテーブルのみにアクセスします。dbccdb または dbccalt の設定の詳細については、『システム管理ガイド』を参照してください。これらのデータベースで使用するテーブルについては、『リファレンス・マニュアル：テーブル』の「[第 2 章 dbccdb のテーブル](#)」を参照してください。

表 4-1 は、この章で説明する dbcc ストアド・プロシージャを示します。dbcc システム・プロシージャ sp_plan_dbccdb の詳細については、sp_plan_dbccdb を参照してください。このシステム・プロシージャと dbcc ストアド・プロシージャの詳細については、『システム管理ガイド』を参照してください。

dbcc ストアド・プロシージャのパーミッション・チェックは、細密なパーミッションの設定によって異なります。細密なパーミッションの詳細については、『セキュリティ管理ガイド』を参照してください。

表 4-1: dbcc ストアド・プロシージャ

プロシージャ名	説明
sp_dbcc_alterws	指定した作業領域のサイズを指定した値に変更して、作業領域を初期化する。
sp_dbcc_configreport	指定したデータベースについて、dbcc checkstorage オペレーションで使われる設定情報を提供するレポートを生成する。

オブジェクトの名前と日付の指定

プロシージャ名	説明
<code>sp_dbcc_createews</code>	指定されたセグメントとデータベース上に、指定されたタイプとサイズで作業領域を作成する。
<code>sp_dbcc_deletedb</code>	指定したターゲット・データベースに関するすべての情報を <code>dbccdb</code> から削除する。
<code>sp_dbcc_deletehistory</code>	指定した日付と時間より前にターゲット・データベースについて実行された <code>dbcc checkstorage</code> の結果を削除する。
<code>sp_dbcc_differentialreport</code>	2つの <code>dbcc</code> オペレーションの間で発生した I/O 統計とフォールトの変化を示すレポートを生成する。
<code>sp_dbcc_evaluatedb</code>	ターゲット・データベースの設定情報を再計算して、現在の設定情報と比較する。
<code>sp_dbcc_exclusions</code>	<code>checkverify</code> および <code>sp_dbcc_faultreport</code> で使用する永続的な除外リストをユーザが作成および管理できるようにする。
<code>sp_dbcc_faultreport</code>	ターゲット・データベースの指定したオブジェクトについて、指定した日付に実行された <code>dbcc checkstorage</code> オペレーションのフォールト統計のレポートを生成し、テーブルおよびインデックスの順にリストする。
<code>sp_dbcc_fullreport</code>	<code>sp_dbcc_summaryreport</code> 、 <code>sp_dbcc_configreport</code> 、 <code>sp_dbcc_statisticsreport</code> 、 <code>sp_dbcc_faultreport</code> を実行する。
<code>sp_dbcc_help_fault</code>	指定されたフォールト・タイプと推奨される修復方法の説明を表示する。
<code>sp_dbcc_patch_finishtime</code>	アポートされた <code>checkverify</code> オペレーションと <code>checkstorage</code> オペレーションに関するレポートを生成する。
<code>sp_dbcc_recommendations</code>	<code>checkstorage</code> の実行結果に基づき、データベースの推奨事項をテーブルとインデックスでグループ化して表示する。
<code>sp_dbcc_runcheck</code>	指定したデータベースについて <code>dbcc checkstorage</code> を実行してから、 <code>sp_dbcc_summaryreport</code> または指定するレポートを実行する。
<code>sp_dbcc_statisticsreport</code>	ターゲット・データベースにある指定したオブジェクトについて、アロケーション統計のレポートを生成する。
<code>sp_dbcc_summaryreport</code>	指定したデータベースについてサマリ・レポートを生成する。
<code>sp_dbcc_updateconfig</code>	ターゲット・データベースの設定情報を使用して、 <code>dbccdb</code> にある <code>dbcc_config</code> テーブルを更新する。

オブジェクトの名前と日付の指定

`dbcc` スタアド・プロシージャには、オブジェクト名と日付用のパラメータを使うものがあります。この項では、オブジェクト名と日付の指定についての重要な情報を説明します。

オブジェクト名の指定

オブジェクト名には、レポート生成の対象となるテーブルまたはインデックスの名前だけを指定します。オブジェクト名を指定するときには、データベース名 (*dbname*) も一緒に指定します。オブジェクトの所有者を指定することはできません。指定したオブジェクト名がターゲット・データベースの中でユニークでないと、システム・プロシージャは、指定した名前を持つすべてのオブジェクトについてのレポートを生成します。

日付の指定

日付と時間を指定するには、下記の構文 (省略可能) を使います。

```
mm/dd/yy[:hh:mm:ss]
```

24 時間表示形式を前提としています。

日付を指定すると、システム・プロシージャは以下のように解釈します。

- 日付と時間の両方が指定されると、指定された日付と時間に完了する **dbcc** オペレーションが、レポート用に選択されます。
- 日付に現在の日付を指定し、時間を指定しないと、時間は自動的に現在の時間に設定されます。この 24 時間内で、現在の時間に最も近い完了時間に完了する **dbcc** オペレーションが、レポート用に選択されます。
- 指定した日付が現在の日付ではなく、時間も指定しない場合、時間は自動的に "23:59:59" に設定されます。完了日時が、指定された日付とシステムが設定した時間に最も近い、**dbcc checkstorage** オペレーションが、レポート用に選択されます。

たとえば、最後に実行された **dbcc checkstorage** オペレーションが 1997 年 3 月 4 日の 10:20:45 に完了したとします。

日付を "03/04/97" と指定すると、システム・プロシージャはこれを 03/04/97:23:59:59 と解釈します。

日付を "03/04/97:10:00:00" に指定すると、10:20:45 に完了するオペレーションは、レポートに使用されません。これは、指定日時以前に完了したオペレーションだけが条件に合致するからです。

日付を "03/06/97" に指定すると、最新のオペレーションの完了日時が、指定日時よりも 24 時間以上前なので、レポートは作成されません。

sp_dbcc_alterws

説明 指定した作業領域のサイズを指定した値に変更して、作業領域を初期化します。

構文 `sp_dbcc_alterws dbname, wsname, "wssize[K|M]"`

パラメータ

dbname

作業領域が存在するデータベースの名前です。dbccdb か dbccalt を指定します。

wsname

変更する作業領域の名前を指定します。

wssize

作業領域の新しいサイズを **K** (キロバイト) か **M** (メガバイト) で指定します。K か M を指定しないと、wssize はページ数を指定します。ページ・サイズはプラットフォームによって異なります。作業領域の最小サイズは 24 ページです。

例 dbccdb の作業領域 scan_ws_000001 のサイズを 30MB に変更します。

```
sp_dbcc_alterws dbccdb, scan_ws_000001, "30M"
```

```
Workspace scan_ws_000001 has been altered successfully to size 30MB
```

使用法

- sp_dbcc_alterws は指定された作業領域のサイズを指定された値に変更して、作業領域を初期化します。
- 最大のパフォーマンスを得るには、少なくとも **16K** のバッファ・プールが設定されていることを確かめてから、作業領域を変更してください。
- 作業領域を変更する前にサイズを見積もるには、[sp_plan_dbccdb](#) を使用してください。
- 作業領域を変更するには、その前にその領域が存在していることを確認してください。作業領域の作成の詳細については、[sp_dbcc_createws](#) を参照してください。
- 作業領域を削除するには、dbccdb で次の文を発行します。

```
drop table workspace_name
```

パーミッション	sp_dbcc_alterws のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、dbccdb (または dbccalt) のデータベース所有者であることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、指定したデータベースのデータベース所有者か、sa_role を持つユーザであることが必要。

参照 scan と text の作業領域、および dbccalt データベースの詳細については、『システム管理ガイド』を参照してください。

コマンド dbcc

dbcc ストアド・プロシージャ [sp_dbcc_createws](#)、[sp_dbcc_evaluatedb](#)

システム・プロシージャ [sp_plan_dbccdb](#)、[sp_helpdb](#)

sp_dbcc_configreport

説明 指定したデータベースについて、`dbcc checkstorage` オペレーションで使用される設定情報を提供するレポートを生成します。

構文 `sp_dbcc_configreport [dbname]`

パラメータ *dbname*
データベースの名前を指定します。*dbname* が指定されなければ、レポートには `dbccdb..dbcc_operation_log` にあるすべてのデータベースに関する情報が入ります。

例 `sybssystemprocs` データベースの `dbcc` 関連の設定情報について、レポートを生成します。“Value” カラムには、該当するオブジェクトがあれば、その名前とサイズがリストされます。

```
sp_dbcc_configreport
```

```
Reporting configuration information of database sybssystemprocs.
```

Parameter Name	Value	Size
database name	sybssystemprocs	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	

使用方法

- `sp_dbcc_configreport` は、指定されたデータベースについて、`dbcc` オペレーションで使われる設定情報を提供するレポートを生成します。この情報は、`dbcc_config` テーブルに格納されます。
- 最新の設定パラメータを調べるには、`sp_dbcc_updateconfig` を実行してから `sp_dbcc_configreport` を実行してください。
- 作業領域の設定値を変更するには、`sp_dbcc_alterws` を使用します。

パーミッション `sp_dbcc_configreport` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>dbccdb</code> (または <code>dbccalt</code>) のデータベース所有者であること、または指定したデータベースに対する <code>report checkstorage</code> 権限が必要。
---------------	---

細密なパーミッションが無効	細密なパーミッションが無効の場合、指定したデータベース名で有効なすべてのユーザが <code>sp_dbcc_configreport</code> を実行できる。
---------------	--

参照 **コマンド** `dbcc`

dbcc ストアド・プロシージャ [sp_dbcc_alterws](#)、[sp_dbcc_fullreport](#)、[sp_dbcc_statisticsreport](#)、[sp_dbcc_summaryreport](#)、[sp_dbcc_updateconfig](#)

sp_dbcc_createws

説明 指定したセグメントとデータベースについて、指定したタイプとサイズの作業領域を作成します。

構文 `sp_dbcc_createws dbname, segname, [wsname], wstype, "wssize[K|M]"`

パラメータ

dbname

作業領域を作成するデータベースの名前です。値は `dbccdb` と `dbccalt` です。

segname

作業領域用のセグメントの名前です。

wsname

作業領域の名前です。値が `null` なら、`sp_dbcc_createws` は `scan` 作業領域に `scan_wsnnnnnn`、`text` 作業領域には `text_wsnnnnnn` の名前を生成します。ここで `nnnnnn` は、ユニークな 6 桁の数字です。

wstype

作成する作業領域のタイプを指定します。値は `scan` と `text` です。

wssize

作業領域のサイズを `K` (キロバイト) か `M` (メガバイト) で指定します。`K` か `M` を指定しないと、`wssize` はページ数を指定します。作業領域の最小サイズは 24 ページです。

例 **例 1** `dbccdb` の `scanseg` セグメントに、`scan_wspubs2` という名前の `scan` 作業領域を 10MB 作成します。

```
sp_dbcc_createws dbccdb, scanseg, scan_wspubs2, scan, "10M"
```

例 2 `dbccdb` の `textseg` セグメントに、`text_ws000001` という名前の `scan` 作業領域を 14MB 作成します。

```
sp_dbcc_createws dbccdb, textseg, text, "14M"
```

使用法

- `sp_dbcc_createws` は、指定した名前とサイズを持つ作業領域を作成して、これを初期化します。
- `sp_addsegment` を使ってセグメントを作成してから、作業領域を作成してください。
- 最大のパフォーマンスを得るには、少なくとも `16K` のバッファ・プールが設定されていることを確かめてから、作業領域を作成してください。
- 作業領域を作成するときに使用するページの割り付けスキームはサイズが大きいため、デバイス上に必要なスペースに 5% のオーバーヘッドを加算してください。

- サイズを見積もるには、[sp_plan_dbccdb](#) を使います。
- `dbcc_config` に新しい設定情報を記録するには、作業領域を作成してから [sp_dbcc_updateconfig](#) を実行します。
- 各作業領域はユニークな名前にしてください。
- 作業領域を削除するには、`dbccdb` で次の文を発行します。

```
drop table workspace_name
```

パーミッション

`sp_dbcc_createws` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>dbccdb</code> (または <code>dbccalt</code>) のデータベース所有者であることが必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <code>sp_dbcc_createws</code> を実行するには、 <code>dbccdb</code> (または <code>dbccalt</code>) のデータベース所有者であるか、 <code>sa_role</code> が必要。

参照

`scan` と `text` の作業領域、および `dbccalt` データベースの詳細については、『システム管理ガイド』を参照してください。

コマンド [dbcc](#)

dbcc スタアド・プロシージャ [sp_dbcc_alterws](#)、[sp_dbcc_evaluatedb](#)

システム・プロシージャ [sp_addsegment](#)、[sp_plan_dbccdb](#)、[sp_helpsegment](#)

sp_dbcc_deletedb

説明	指定したターゲット・データベースに関するすべての情報を dbccdb から削除します。
構文	sp_dbcc_deletedb [dbname dbid]
パラメータ	<p>dbname 設定情報を削除するターゲット・データベースの名前を指定します。dbname の値を指定しなければ、Adaptive Server は dbccdb..dbcc_config にあるすべてのデータベースからデータを削除します。ターゲット・データベースが dbccdb であり、dbccalt が存在すると、Adaptive Server は dbccalt からデータを削除します。</p> <p>dbid 設定情報を削除するターゲット・データベースのデータベース ID を指定します。</p>
例	<p>dbccdb から、engdb という名前のデータベースの情報すべてを削除します。</p> <pre>sp_dbcc_deletedb "engdb"</pre> <p>All information for database engdb has been deleted from dbccdb.</p>
使用法	<ul style="list-style-type: none"> • sp_dbcc_deletedb は、設定情報と前の dbcc checkstorage オペレーションの結果など、指定したターゲット・データベースに関するすべての情報を dbccdb から削除します。 • 削除されるデータベースが dbccdb であり、dbccalt データベースが存在する場合、sp_dbcc_deletedb は dbccalt から設定情報と dbccdb の結果を削除します。 • 特定の日付より前に作成された dbcc checkstorage の結果を削除するには、sp_dbcc_deletehistory を使います。 • すでに削除されているデータベースに関する情報を dbccdb データベースから削除するには、dbid オプションを使用します。それ以外の方法では削除できません。
パーミッション	sp_dbcc_deletedb のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、dbccdb (または dbccalt) のデータベース所有者であること、または指定したデータベースに対する manage checkstorage 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、sp_dbcc_deletedb を実行するには、指定したデータベースのデータベース所有者であるか、または sa_role が必要。

参照

dbccalt データベースの詳細については、『システム管理ガイド』を参照してください。

コマンド [dbcc](#)

dbcc ストアド・プロシージャ [sp_dbcc_deletehistory](#)、[sp_dbcc_evaluatedb](#)

システム・プロシージャ [sp_plan_dbccdb](#)

sp_dbcc_deletehistory

説明 指定した日付と時間より前にターゲット・データベースについて実行された dbcc checkstorage オペレーションの結果を削除します。

注意 作業領域は事前に割り当てられており、固定サイズであるため、sp_dbcc_deletehistory では、削除された履歴データに関連する領域は解放されません。

構文

sp_dbcc_deletehistory [*cutoffdate* [, *dbname* | *dbid*]]

パラメータ

cutoffdate

この日付以前に作られたエントリすべてを削除します。このパラメータは **datetime** 型です。日付が指定されないと、最後の結果だけが残ります。詳細については、「[日付の指定](#)」(907 ページ)を参照してください。

dbname

データを削除するデータベースの名前を指定します。指定しなければ、sp_dbcc_deletehistory は dbccdb.dbcc_config にあるすべてのデータベースの履歴情報を削除します。

dbid

履歴情報を削除するターゲット・データベースのデータベース ID を指定します。

例

1997 年 3 月 4 日以前にデータベース pubs2 で実行されたすべてのオペレーションの結果を削除します。

```
sp_dbcc_deletehistory "03/04/1997", "pubs2"
```

使用法

- sp_dbcc_deletehistory は、指定した日時より前にターゲット・データベースで実行された dbcc checkstorage の結果を削除します。
- ターゲット・データベースが dbccdb であり、dbccalt データベースが存在する場合、sp_dbcc_deletehistory は dbccalt から dbccdb の履歴データを削除します。
- *cutoffdate* に指定された値は、各 dbcc オペレーションの終了時間と比較されます。
- すでに削除されているデータベースの履歴情報を dbccdb データベースから削除するには、*dbid* オプションを使用します。
- すでに削除されているデータベースの履歴情報を dbccdb データベースから削除するには、*dbid* オプションを使用する以外に方法はありません。

- *cutoffdate* の値を決定するために `dbcc checkstorage` が実行された日付を調べるには、[sp_dbcc_summaryreport](#) を実行してください。

パーミッション

`sp_dbcc_deletehistory` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>dbccdb</code> (または <code>dbccalt</code>) のデータベース所有者であること、または指定したデータベースに対する <code>manage checkstorage</code> 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、特定のデータベースに対して <code>sp_dbcc_deletehistory</code> を実行するには、指定したデータベースのデータベース所有者か、 <code>sa_role</code> を持つユーザであることが必要。データベース名を指定しないで <code>sp_dbcc_deletehistory</code> を実行できるのは、 <code>sa_role</code> を持つユーザのみ。

参照

`dbccalt` データベースの詳細については、『システム管理ガイド』を参照してください。

コマンド `dbcc`

dbcc スタアド・プロシージャ [sp_dbcc_deletedb](#)、[sp_dbcc_evaluatedb](#)

システム・プロシージャ [sp_plan_dbccdb](#)

sp_dbcc_differentialreport

説明	2 つの dbcc オペレーションの間で発生した I/O 統計とフォールトの変化を示すレポートを生成します。
構文	<pre>sp_dbcc_differentialreport [dbname [, objectname]], [db_op] [, "date1" [, "date2"]]</pre>
パラメータ	<p>dbname データベースの名前を指定します。 dbname を指定しなければ、レポートには dbccdb..dbcc_operation_log にあるすべてのデータベースの情報が入ります。</p> <p>objectname レポートを生成するテーブルまたはインデックスの名前を指定します。 object_name が指定されなければ、ターゲット・データベースにあるすべてのオブジェクトの統計がレポートされます。</p> <p>db_op レポートに使用するデータ・ソースを指定します。値は checkstorage だけです。ターゲット・データベース内にある指定のオブジェクトについて、 date1 の日付と date2 の日付で db_op によって指定されたデータに関するレポートが生成されます。日付が指定されなければ、 db_op タイプの最後の 2 つのオペレーションが比較されます。</p> <p>date1 比較する dbcc checkstorage オペレーションの最初の日付を指定します。</p> <p>date2 比較する dbcc checkstorage オペレーションの最後の日付を指定します。</p>
例	<p>1997 年 5 月 1 日 ~ 1997 年 5 月 4 日の間に sysprocedures テーブルで発生した I/O 統計とフォールトの変化を示すレポートを生成します。</p> <pre>sp_dbcc_differentialreport master, sysprocedures, checkstorage, "05/01/97", "05/04/97"</pre>
使用法	<ul style="list-style-type: none"> • sp_dbcc_differentialreport は、2 回の dbcc オペレーション間における I/O 統計とフォールトの変化を示すレポートを生成します。これは、dbcc checkstorage の 2 つのインスタンスからレポートされたカウンタ値を比較します。変更された値だけがレポートされます。 • 日付が 1 つだけ指定されると、指定された日付によって選ばれた dbcc checkstorage オペレーションの結果は、選ばれたオペレーション直前の dbcc checkstorage オペレーションの結果と比較されます。

- 日付が指定されなければ、最後の 2 つの dbcc checkstorage オペレーションの結果が比較されます。
- sp_dbcc_differentialreport が *object_name* の番号を返す場合は、dbcc checkstorage オペレーションが完了したあとにそのオブジェクトが削除されたことを意味します。
- 指定されたオペレーションの間で何も変化がないと、sp_dbcc_differentialreport はレポートを生成しません。

パーミッション

sp_dbcc_differentialreport のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、dbccdb (または dbccalt) のデータベース所有者であること、または指定したデータベースに対する report checkstorage 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、指定したデータベース名で有効なすべてのユーザが sp_dbcc_differentialreport を実行できる。

参照

コマンド [dbcc](#)

dbcc スタアド・プロシージャ [sp_dbcc_fullreport](#)、[sp_dbcc_statisticsreport](#)、[sp_dbcc_summaryreport](#)、[sp_dbcc_updateconfig](#)

sp_dbcc_evaluatedb

説明 ターゲット・データベースの設定情報を再計算して、現在の設定情報と比較します。

構文 sp_dbcc_evaluatedb [dbname]

パラメータ *dbname*
 ターゲット・データベースの名前を指定します。*dbname* を指定しなかった場合は、sp_dbcc_evaluatedb は dbcc_config テーブルにリストされたすべてのデータベースを比較します。

例 現在のデータベース sybssystemprocs の設定情報を再計算して、いくつかのパラメータについて新しい推奨値を提案します。

```
1sp_dbcc_evaluatedb
2> go
```

Recommended values for workspace size, cache size and process count are:

Database name :one_G

	current	suggested
scan workspace size :	750M	16M
text workspace size :	2K	48K
cache size :	10240K	1280K
process count :	3	2
compression mem size:	2048K	12M

Each of the reported quantities is reported in a scaled unit according to
 G if size > 10G
 M if 10M < size <=10 G
 K otherwise

- 使用法**
- 圧縮されたデータまたはログ・デバイスを持つアーカイブ・データベースがある場合、出力には推奨される圧縮メモリ・サイズを示す行が含まれます。
 - sp_dbcc_evaluatedb は、ターゲット・データベースの設定情報を再計算して、現在の設定情報と比較します。これは、dbcc_counters テーブル内のターゲット・データベースについて記録されたカウンタ値を使用します。
 - キャッシュ・サイズは、キャッシュ内の 16K バッファ・プールのサイズです。2K バッファ・プールの場合、このキャッシュの最小サイズは、推奨値に 512 を加算した値にしてください。
 - ターゲット・データベースのサイズとデータの分散形態を変更するときは、設定情報を最適化するために sp_dbcc_evaluatedb を実行してください。

- ターゲット・データベースの設定情報を初めて収集する場合は、[sp_plan_dbccdb](#) を使用します。
- [sp_dbcc_updateconfig](#) を実行して最新の設定情報を評価していることを確かめてから、[sp_dbcc_evaluatedb](#) を実行してください。

パーミッション

[sp_dbcc_evaluatedb](#) のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 dbccdb (または dbccalt) のデータベース所有者であること、または指定したデータベースに対する <code>manage checkstorage</code> 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、指定したデータベースのデータベース所有者か、 <code>sa_role</code> を持つユーザであることが必要。データベース名を指定しないで sp_dbcc_evaluatedb を実行できるのは、システム管理者のみ。

参照

コマンド [dbcc](#)

[dbcc](#) ストアド・プロシージャ [sp_dbcc_updateconfig](#)

システム・プロシージャ [sp_plan_dbccdb](#)

sp_dbcc_exclusions

説明	checkverify と sp_dbcc_faultreport で使用される永続的な除外リストをユーザが作成、管理できるようにします。
構文	sp_dbcc_exclusions <i>dbname</i> , <i>op</i> , <i>type</i> , <i>exclusion_list</i>
パラメータ	<p>dbname 除外を適用する対象のデータベースの名前です。すべてのデータベースに適用する場合は null を指定します。</p> <p>op 実行するオペレーションを指定します。有効な値は次のとおりです。</p> <ul style="list-style-type: none"> • add — 新しい除外を登録する (重複は無視)。 • drop — 指定した除外が既に登録されている場合は削除する。 • listall — すべてのデータベースについて、記録されている除外をリストする。 <p>type 除外する項目のタイプです。有効な値は faults、tables、combo、および null (op が null または listall のいずれかの場合) です。型は varchar です。</p> <p>exclusion_list フォールト、テーブル、テーブルとフォールトのエントリ、または null のカンマ区切りのリストです。型は varchar です。</p>
例	<p>例 1 sp_dbcc_faultreport でのすべてのデータベースに対する処理から、syslogs テーブルと syscomments テーブルを除外します。</p> <pre>sp_dbcc_exclusions null, 'add', 'tables', 'syslogs, syscomments'</pre> <p>例 2 データベース my_db の処理から、フォールト・タイプ 100036 を除外します。</p> <pre>sp_dbcc_exclusions my_db, 'add', 'faults', '100036'</pre> <p>例 3 my_db に対応する除外リストに、テーブル mytable に関するフォールト・タイプ 100002 と、テーブル syslogs に関するフォールト・タイプ 100035 を追加します。</p> <pre>sp_dbcc_exclusions my_db, 'add', 'combo', 'mytable:100002, syslogs:100035'</pre> <p>例 4 my_db に対応する除外リストから、フォールト・タイプ 100036 を削除します。</p> <pre>sp_dbcc_exclusions my_db, 'drop', 'faults', '100036'</pre>

例 5 my_db に対応する除外リストを表示します。

```
sp_dbcc_exclusions my_db
```

例 6 すべてのデータベースについて、記録されている除外を表示します。

```
sp_dbcc_exclusions null, 'listall'
```

使用法

- *listall* を指定する場合は、*dbname* には *null* を指定します。*op* が *null* の場合、*sp_dbcc_exclusions* では、指定したデータベースに対して記録されている除外がリストされます。
- *sp_dbcc_exclusions* で *dbname* パラメータに *null* 以外を指定できるのは、システム管理者またはデータベース所有者だけです。
- *dbname* パラメータと *op* パラメータが *null* の場合、ユーザは、システム管理者であるか、除外が記録されているデータベースの少なくとも 1 つの所有者である必要があります。
- *dbname* パラメータが *null* で、*op* パラメータが *listall* の場合、ユーザは、システム管理者であるか、除外が記録されているデータベースの少なくとも 1 つの所有者である必要があります。ユーザがシステム管理者でない場合、そのユーザが所有するデータベースに対して記録されている除外のみがレポートされます。

パーミッション

sp_dbcc_exclusions のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <i>dbccdb</i> (または <i>dbccalt</i>) のデータベース所有者であること、または指定したデータベースに対する <i>manage checkstorage</i> 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、 <i>sa_role</i> を持つユーザであることが必要。

sp_dbcc_faultreport

説明 ターゲット・データベースの指定したオブジェクトについて、指定した日付に実行された dbcc checkstorage オペレーションのフォールト統計のレポートを生成し、テーブルおよびインデックスの順にリストします。

構文

```
sp_dbcc_faultreport [report_type [, dbname [, objectname
                    [, date [, hard_only [, exclusion_mode[, exclusion_faults
                    [, exclusion_tables [, exclusion_combo
                    [, display_recommendations [, opid [, fault_type_in]]]]]]]]]]]]]
```

パラメータ

report_type

フォールト・レポートのタイプを指定します。値は short と long です。デフォルトは short です。

dbname

ターゲット・データベースの名前、たとえば master..sysdatabases を指定します。dbname が指定されなければ、レポートには dbccdb..dbcc_operation_log にあるすべてのデータベースに関する情報が入ります。

object_name

レポートを生成するテーブルまたはインデックスの名前を指定します。object_name が指定されなければ、ターゲット・データベースにあるすべてのオブジェクトの統計がレポートされます。

date

dbcc checkstorage オペレーションが終了した正確な日付と時刻を指定します。この値は、dbcc_operation_log.finish にあります。この値は、sp_dbcc_summaryreport 出力の start time の日付と、end time の時間と分を組み合わせて作成できます。date を指定しなければ、Adaptive Server は最新オペレーションの日付を採用します。

date パラメータを指定するときは、オペレーションを実行した日時よりも後の日時を指定してください。sp_dbcc_faultreport は、このパラメータで指定した日時以前に発生したフォールトしかレポートしません。

注意 date パラメータに重点を置くには、ほかのすべてのパラメータに null を指定します。すべてのパラメータを省略すると、sp_dbcc_faultreport は正しいレポートを生成できません。

hard_only

1 を指定すると、ハード・フォールトについてのレポートが有効になります。指定できる値は 0 と 1 で、デフォルトは 0 です。

display_recommendations

sp_dbcc_recommendations によって生成される推奨事項のレポートを有効にします。exclusion_mode、exclusion_faults、exclusion_tables、display_recommendations、および exclusion_combo の各パラメータは、除外に対応するオプションのパラメータです。

exclusion_mode

型は varchar で、デフォルトでオンです。これを無効にするには、sp_dbcc_faultreport を実行するたびに ignore を指定してください。次のいずれかを使用します。

- ignore — 永続的な除外リストを無視し、一時的な除外リストを使用する (除外リストが指定されている場合)。型は varchar。
- extend — 一時的な除外リストと永続的な除外リストの両方を適用する (型は varchar)。

exclusion_faults

レポートから除外するフォールト・タイプのカンマ区切りリストです (型は varchar)。

exclusion_tables

レポートから除外するテーブルのカンマ区切りリストです (型は varchar)。

exclusion_combo

レポートから除外するフォールトとテーブルの組み合わせのカンマ区切りリストです (型は varchar)。

opid

直前のオペレーション ID ではなく、特定の日付の特定のオペレーション ID に対するフォールト・レポートを有効にします。デフォルトではオペレーション ID は指定されません。

fault_type_in

特定のフォールト・タイプに対するフォールト・レポートを有効にします。デフォルトは NULL です。

例 **例 1** sysystemprocs データベースに格納されているテーブル内で見つけたフォールトの短いレポートを生成します。レポートには、テーブル名、フォールトが発生したインデックス番号、フォールトのタイプ・コード、フォールトの簡単な説明、フォールトが発生したページ番号が入ります。

```
sp_dbcc_faultreport "short"
```

```
Database Name :sybssystemprocs
```

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

例 2 sybssystemprocs データベースに格納されているテーブル内で見つけたフォールトの長いレポートを生成します。この例は長いレポート出力の初めの部分を示しています。この完全なレポートには、dbcc checkstorage によってフォールトが発見された「ターゲット・データベース」内の各オブジェクトの情報が表示されます。"page header" フィールドの下にある長い数字列の下のデータ ("Header for 14151, next 14216, previous 14150 ...") は、"page header" 文字列の要素を記述しています。

```
sp_dbcc_faultreport "long"
```

```
Generating 'Fault Report' for object sysprocedures in database
sybssystemprocs.
```

```
Type Code:100031; Soft fault, possibly spurious
```

```
Page reached by the chain is not allocated.
```

```
page id:14151
```

```
page
```

```
header:0x00003747000037880000374600000005000648B803EF0001000103FE0080000F
```

```
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)
```

```
.
.
.
```

例 3 sp_dbcc_summaryreport の出力で End Time として表示されている日時に完了したオペレーションの、すべてのデータベースのすべてのテーブルで見つけたフォールトの短いレポートを生成します。date パラメータには正確な終了日時を指定することが重要です。たとえば、

```
7/25/2000 9:58
```

と指定すると、

```
7/25/2000 9:58:0:190
```

と指定した場合と異なり、9:58 までに発生したフォールトだけがレポートされ、それ以降のフォールトはレポートされません。オペレーションの正確な終了日時を指定しないで、9:59 と指定する方法もあります。

```
sp_dbcc_faultreport "short", NULL, NULL,  
"07/25/00 9:59"
```

この場合、9:59 までに発生したフォールトがレポートされます。

例 4 mydb というデータベースに対する直近の checkstorage の実行でレポートされたハード・フォールトのみについて、短い形式のレポートを生成します。

```
sp_dbcc_faultreport short, mydb, @hard_only = 1
```

例 5 データベース my_db のフォールト・レポートに対し、推奨された修正を追加します。

```
sp_dbcc_faultreport @dbname = my_db,  
@display_recommendations = 1
```

例 6 修正の推奨を含まないフォールト・レポートを生成します。

```
sp_dbcc_faultreport @dbname = my_db
```

例 7 データベース my_db に対し、永続的な除外リストを無効にして sp_dbcc_faultreport を実行します。

```
sp_dbcc_faultreport @dbname = 'my_db', @exclusion_mode = 'ignore'
```

例 8 データベース my_db に対し、永続的な除外リストを有効にし、さらにフォールト・タイプ 100036 の処理を除外して sp_dbcc_faultreport を実行します。

```
sp_dbcc_faultreport @dbname = 'my_db', @exclusion_mode = 'extend',  
@exclusion_faults = '100036'
```

例 9 データベース my_db に対し、永続的な除外リストを有効にし、さらにテーブル tab を除外して sp_dbcc_faultreport を実行します。

```
sp_dbcc_faultreport @dbname = 'my_db', @exclusion_mode = 'extend',  
@exclusion_tables = 'tab'
```

例 10 データベース my_db に対し、永続的な除外リストを無効にし、テーブル tab とフォールト・タイプ 100036 の処理を除外する一時的な除外リストを有効にして、sp_dbcc_faultreport を実行します。

```
sp_dbcc_faultreport @dbname = 'my_db', @exclusion_mode = 'ignore',  
@exclusion_faults = '100036', @exclusion_tables = 'tab'
```


例 11 データベース `my_db` に対し、永続的な除外リストを無効にし、テーブル `mytable` に関するフォールト・タイプ '100002' の処理と、テーブル `tab` に関するフォールト・タイプ 100035 の処理を除外する一時的な除外リストを有効にして、`sp_dbcc_faultreport` を実行します。

```
sp_dbcc_faultreport @dbname = 'my_db', @exclusion_mode = 'ignore',
  @exclusion_combo = 'mytable:100002, tab:100035'
```

例 12 `mydb` データベースに対する最新の `checkstorage` の実行でレポートされた 100029 フォールトについて、長い形式のレポートを生成します (100029 はページ・ヘッダ・エラーのフォールト・タイプ)。

```
sp_dbcc_faultreport long, mydb, @fault_type_in = 100029
```

例 13 `mydb` データベースに対してオペレーション ID 5 で実行された `checkstorage` によりレポートされたフォールトについて、短い形式のレポートを生成します。

```
sp_dbcc_faultreport short, mydb, @opid = 5
```

使用法

- `sp_dbcc_faultreport` は、ターゲット・データベースにある指定されたオブジェクトのフォールトすべてを示すレポートを生成します。
- 次の処理を行うと、`sp_dbcc_faultreport` はエラー番号 10028 のエラー・メッセージを大量に出力します。
 - `sp_placeobject` を使用して、すでに領域が割り付けられているオブジェクトに新しいセグメントで新しい割り付けを行う場合。
 - `sp_dropsegment` を使用して、セグメントに配置されているオブジェクトの割り付けを含むフラグメントからそのセグメントのマップを解除する場合。

エラー番号 100028 のエラー・メッセージは、重大なエラーを示すメッセージではなく、情報メッセージです。このメッセージを受け取りたくない場合は、このエラー (またはその他のエラー) をレポートしないレポート用プロシージャを作成します。たとえば、`installdbccdb` スクリプトで定義されている標準の `sp_dbcc_faultreport` ストアド・プロシージャの先頭に次の文を追加することで、そのようなプロシージャを作成できます。

```
print "removing 100028 errors from dbcc_faults table"
delete dbcc_faults where type_code = 100028
```

- `sp_dbcc_faultreport` が `object_name` の番号を返すと、`dbcc checkstorage` オペレーションのあと、そのオブジェクトは削除されたことを意味します。

パーミッション	sp_dbcc_faultreport のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、dbccdb (または dbccalt) のデータベース所有者であること、または指定したデータベースに対する report checkstorage 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、指定したデータベース名で有効なすべてのユーザが sp_dbcc_faultreport を実行できる。
参照	フォールト ID とフォールト・ステータスの詳細については、『システム管理ガイド』の「type_code カラム」を参照してください。 コマンド dbcc dbcc スタアド・プロシージャ sp_dbcc_fullreport 、 sp_dbcc_statisticsreport 、 sp_dbcc_summaryreport 、 sp_dbcc_updateconfig

sp_dbcc_fullreport

説明	指定された <i>date</i> 日以前の <i>database..object_name</i> に対し、 <code>sp_dbcc_summaryreport</code> 、 <code>sp_dbcc_configreport</code> 、 <code>sp_dbcc_statisticsreport</code> 、 <code>sp_dbcc_faultreport short</code> を実行します。
構文	<code>sp_dbcc_fullreport [dbname [, objectname [, date]]]</code>
パラメータ	<p><i>dbname</i> データベースの名前を指定します。たとえば、<code>master..sysdatabases</code> を指定します。<i>dbname</i> を指定しなければ、レポートには <code>dbccdb..dbcc_operation_log</code> にあるすべてのデータベースの情報が入ります。</p> <p><i>object_name</i> レポートを生成するテーブルまたはインデックスの名前を指定します。<i>object_name</i> を指定しなければ、ターゲット・データベースにあるすべてのオブジェクトの統計がレポートされます。</p> <p><i>date</i> <code>dbcc checkstorage</code> オペレーションが実行された日付を指定します。<i>date</i> を指定しなければ、最後のオペレーションの日付が使用されます。</p>
例	<p><code>master</code> データベース内の <code>sysprocedures</code> テーブル上で実行された最新の <code>dbcc checkstorage</code> オペレーションに対し、<code>sp_dbcc_summaryreport</code>、<code>sp_dbcc_configreport</code>、<code>sp_dbcc_statisticsreport</code>、<code>sp_dbcc_faultreport short</code> を実行します。</p> <pre>sp_dbcc_fullreport master, sysprocedures</pre>
使用法	<ul style="list-style-type: none"> <code>sp_dbcc_fullreport</code> は、指定された日付以前の <i>database..object_name</i> に対し、<code>sp_dbcc_summaryreport</code>、<code>sp_dbcc_configreport</code>、<code>sp_dbcc_statisticsreport</code>、<code>sp_dbcc_faultreport short</code> を実行します。
パーミッション	<code>sp_dbcc_fullreport</code> のパーミッション・チェックは、細密なパーミッションの設定によって異なります。
細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>dbccdb</code> (または <code>dbccalt</code>) のデータベース所有者であること、または指定したデータベースに対する <code>report checkstorage</code> 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、指定したデータベース名で有効なすべてのユーザが <code>sp_dbcc_fullreport</code> を実行できる。
参照	<p>コマンド <code>dbcc</code></p> <p>dbcc ストアド・プロシージャ <code>sp_dbcc_statisticsreport</code>、 <code>sp_dbcc_summaryreport</code>、<code>sp_dbcc_updateconfig</code></p>

sp_dbcc_help_fault

説明 指定されたフォールト・タイプと推奨される修復方法の説明を表示します。

構文 sp_dbcc_help_fault [fault_type]

パラメータ *fault_type*
説明と推奨の修復方法をレポートする対象のフォールト・タイプを指定します。このパラメータの型は `int` です。*fault_type* を指定しない場合、`sp_dbcc_help_fault` はすべてのフォールト・タイプについてレポートします。

例 **例 1** フォールト・タイプ 100038 の説明と推奨される修復方法を参照するには、次のように入力します。

```
sp_dbcc_help_fault 100038
```

例 2 すべてのフォールト・タイプの説明と推奨される修復方法を参照するには、次のように入力します。

```
sp_dbcc_help_fault
```

使用法 `sp_dbcc_help_fault` は、指定されたフォールト・タイプと推奨される修復方法の説明を表示します。

パーミッション `sp_dbcc_help_fault` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>dbccdb</code> (または <code>dbccalt</code>) のデータベース所有者であることが必要。
---------------	---

細密なパーミッションが無効	細密なパーミッションが無効の場合、すべてのユーザが <code>sp_dbcc_help_fault</code> を実行できる。
---------------	---

sp_dbcc_patch_finishtime

説明	アボートされた <code>checkverify</code> オペレーションと <code>checkstorage</code> オペレーションに関するレポートを生成します。
構文	<code>sp_dbcc_patch_finishtime dbname, opid [,optype [,seq [,finishtime]]]</code>
パラメータ	<p>dbname アボート時に <code>checkstorage</code> または <code>checkverify</code> が処理していたデータベースの名前です。このパラメータは <code>varchar</code> 型です。</p> <p>opid アボートされたオペレーションに対応するオペレーション ID です。このパラメータは <code>smallint</code> 型です。</p> <p>optype 調べるオペレーションの種類です。有効な値は 'checkstorage' と 'checkverify' です。このパラメータは <code>varchar</code> 型です。</p> <p>seq <code>checkverify</code> のシーケンス番号です (<code>checkstorage</code> では使用しませんが、<code>checkverify</code> では必須です)。このパラメータは <code>smallint</code> 型です。</p> <p>finishtime <code>checkstorage</code> または <code>checkverify</code> のオペレーションがアボートされた時刻を表す <code>datetime</code> 値です。デフォルト値は現在の時刻です。</p>
例	<p>例 1 データベース <code>my_db</code> について、次のエラーが発生したときの <code>checkstorage</code> と <code>checkverify</code> のレポートを有効にします。</p>

```
dbcc checkstorage (my_db)
```

```
Checking my_db:Logical pagesize is 2048 bytes
00:00000:00014:2003/01/20 11:50:05.01 server Error:9960, Severity:20,
State:1
A non-recoverable error has occurred in the CHECKSTORAGE operation.The
operation has been aborted.

Msg 9970, Level 20, State 1:
Line 2:
DBCC cannot update the finish time in dbcc_operation_log table for this
operation(opid = '1') of database 'my_db'.This can be patched by executing
sp_dbcc_patch_finishtime.
```

例 2 データベース `my_db` について、次のエラーが発生したときの `checkstorage` と `checkverify` のレポートを有効にします。

```
dbcc checkstorage (my_db)
```

```
Checking my_db:Logical pagesize is 2048 bytes
00:00000:00014:2003/01/20 11:50:05.01 server Error:9960, Severity:20,
State:1
A non-recoverable error has occurred in the CHECKSTORAGE operation.The
operation has been aborted.
```

```
Msg 9970, Level 20, State 1:
Line 2:
DBCC cannot update the finish time in dbcc_operation_log table for this
operation(opid = '1') of database 'my_db'.This can be patched by executing
sp_dbcc_patch_finishtime.
```

エラー・メッセージに含まれている情報で `sp_dbcc_patch_finishtime` を実行します。

```
sp_dbcc_patch_finishtime my_db, 1
```

使用法

`checkstorage` または `checkverify` オペレーションがアボートされる時、オペレーションの ID と、オペレーションのアボート時に調査されたデータベース名を含むメッセージが出力されます。`checkverify` オペレーションがアボートされた場合は、メッセージにシーケンス番号も出力されます。メッセージには、`sp_dbcc_patch_finishtime` を実行するように指示があり、`dbname` と `opid` が示されます。さらに、`checkverify` オペレーションの場合は、シーケンス番号 `seq` も示されます。`sp_dbcc_patch_finishtime` を実行した後、アボートされたオペレーションに関するフォールト・レポートを作成できます。

パーミッション

`sp_dbcc_patch_finishtime` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効	細密なパーミッションが有効の場合、 <code>dbccdb</code> (または <code>dbccalt</code>) のデータベース所有者であること、または指定したデータベースに対する <code>manage checkstorage</code> 権限が必要。
細密なパーミッションが無効	細密なパーミッションが無効の場合、指定したデータベースのデータベース所有者か、 <code>sa_role</code> を持つユーザであることが必要。

sp_dbcc_recommendations

- 説明** 指定されたオペレーション ID または日付に該当する `checkstorage` オペレーションによってレポートされたフォールトを分析し、ターゲット・データベース内の指定されたオブジェクトに関して推奨される対処法のリストを生成します。
- 構文** `sp_dbcc_recommendations dbname [, "date" [, opid [, "objectname"]]]`
- パラメータ**
- dbname**
推奨事項を生成する対象のデータベースの名前です。型は `varchar` で、このパラメータは必須です。
- date**
`dbcc checkstorage` オペレーション (レポートされたフォールトを分析する対象のオペレーション) が終了した日時を表す `datetime` 値です。 `date` と `opid` を指定しなかった場合は、最後にオペレーションを実行した日付が使用されます。 `date` と `opid` の両方を指定した場合、`date` は無視されます。 `date` はオプションです。
- opid**
`checkstorage` オペレーション (レポートされたフォールトを分析する対象のオペレーション) のオペレーション ID です。 `opid` と `date` を指定しなかった場合は、最後にオペレーションを実行した日付が採用されます。 `date` と `opid` の両方が指定された場合、`date` は無視されます。このパラメータの型は `int` です。
- objectname**
`sp_dbcc_recommendations` で推奨事項を生成する対象のオブジェクトの名前です。 `objectname` を指定しない場合は、データベースにあるすべてのオブジェクトについての推奨事項が生成されます。このパラメータの型は `varchar` です。
- 例**
- 例 1** オペレーション ID 2 に対応する `checkstorage` オペレーションでレポートされたフォールトに基づいて、データベース `my_db` のオブジェクト `t1` について、推奨される修復方法のリストを生成します。
- ```
sp_dbcc_recommendations my_db, null, 2, 't1'
```
- 例 2** 2002 年 9 月 15 日 7:10:18:463PM に完了した `checkstorage` オペレーションでレポートされたフォールトに基づいて、データベース `my_db` のすべてのオブジェクトについて、推奨される修復方法のリストを生成します。

```
sp_dbcc_recommendations my_db, 'Sep 15 2002 7:10:18:463PM'
```

**例 3** 最後に実行した `checkstorage` オペレーションに基づいて、データベース `my_db` のすべてのオブジェクトについて、推奨される修復方法のリストを生成します。

```
sp_dbcc_recommendations my_db
```

**使用法**

`sp_dbcc_recomendations` は、指定されたオペレーション ID または日付に該当する `checkstorage` オペレーションによってレポートされたフォールトを分析し、ターゲット・データベース内の指定されたオブジェクトに関して推奨される対処法のリストを生成します。

**パーミッション**

`sp_dbcc_recommendations` のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

細密なパーミッションが有効

細密なパーミッションが有効の場合、`dbccdb` (または `dbccalt`) のデータベース所有者であること、または指定したデータベースに対する `report checkstorage` 権限が必要。

細密なパーミッションが無効

細密なパーミッションが無効の場合、指定したデータベース名で有効なすべてのユーザが `sp_dbcc_recommendations` を実行できる。



## sp\_dbcc\_runcheck

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 説明    | 指定したデータベースについて <code>dbcc checkstorage</code> を実行してから、 <a href="#">sp_dbcc_summaryreport</a> または指定したレポートを実行します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 構文    | <code>sp_dbcc_runcheck dbname [, user_proc]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| パラメータ | <p><i>dbname</i><br/>         検査を行うデータベースの名前を指定します。</p> <p><i>user_proc</i><br/>         dbcc ストアド・プロシージャまたは <a href="#">sp_dbcc_summaryreport</a> の代わりに実行するユーザ作成のストアド・プロシージャの名前を指定します。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 例     | <p><b>例 1</b> engdb データベースを調べて、検出された情報のサマリ・レポートを生成します。</p> <pre>sp_dbcc_runcheck "engdb"</pre> <p><b>例 2</b> pubs2 データベースを調べて、完全なレポートを生成します。</p> <pre>sp_dbcc_runcheck "pubs2", sp_dbcc_fullreport</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 使用法   | <ul style="list-style-type: none"> <li>• <code>sp_dbcc_runcheck</code> は、指定されたデータベースについて、<code>dbcc checkstorage</code> を実行します。</li> <li>• <code>sp_dbcc_runcheck</code> は <code>dbcc checkstorage</code> の実行中にエラーを検出すると、<code>dbcc checkverify</code> を自動的に実行して <code>checkstorage</code> のソフト・フォールトを確定または処理してから、<code>sp_dbcc_summaryreport</code> を実行します。</li> <li>• <code>dbcc checkstorage</code> オペレーションが完了してから、<code>sp_dbcc_runcheck</code> はサマリ・レポートを生成する <a href="#">sp_dbcc_summaryreport</a> を実行します。他のレポート生成 <code>dbcc</code> ストアド・プロシージャのどれかを <code>dbcc_report</code> に指定すると、<code>sp_dbcc_runcheck</code> は、<a href="#">sp_dbcc_summaryreport</a> ではなく、そのプロシージャを実行します。<code>dbccdb</code> で提供されているすべてのレポート生成ストアド・プロシージャの簡単な説明と例については、『システム管理ガイド』を参照してください。</li> <li>• レポート生成ストアド・プロシージャを自分で作成し、<code>user_proc</code> にその名前を指定できます。ストアド・プロシージャは、自己包含型でなければなりません。<code>sp_dbcc_runcheck</code> は、Adaptive Server にどのようなパラメータも渡せません。</li> </ul> |

|               |                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------|
| パーミッション       | sp_dbcc_runcheck のパーミッション・チェックは、細密なパーミッションの設定によって異なります。                                                 |
| 細密なパーミッションが有効 | 細密なパーミッションが有効の場合、指定したデータベースに対する <code>dbcc checkstorage</code> 権限と <code>dbcc checkverify</code> 権限が必要。 |
| 細密なパーミッションが無効 | 細密なパーミッションが無効の場合、指定したデータベースのデータベース所有者か、 <code>sa_role</code> を持つユーザであることが必要。                            |

参照

コマンド [dbcc](#)

**dbcc** ストアド・プロシージャ [sp\\_dbcc\\_summaryreport](#)

## sp\_dbcc\_statisticsreport

**説明** ターゲット・データベースにある指定したオブジェクトについて、アロケーション統計のレポートを生成します。

**構文** `sp_dbcc_statisticsreport [dbname [, objectname [, date]]]`

**パラメータ**

*dbname*

ターゲット・データベースを指定します。*dbname* が指定されなければ、レポートには `dbccdb..dbcc_operation_log` にあるすべてのデータベースに関する情報が入ります。

*objectname*

レポートを生成するテーブルまたはインデックスの名前を指定します。*objectname* を指定しなければ、Adaptive Server はターゲット・データベースのすべてのオブジェクトの統計をレポートします。

*date*

`dbcc checkstorage` オペレーションが実行された日付を指定します。*date* を指定しなければ、Adaptive Server は最新オペレーションの日付を採用します。

**例**

`sybsystemprocs` データベースの `sysobjects` テーブルについて、統計レポートを生成します。

```
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   |

#### 使用法

- `sp_dbcc_statisticsreport` は、ターゲット・データベースにある、指定されたオブジェクトのアロケーション統計のレポートを生成します。このコマンドは、`dbcc_counters` テーブルのデータを使用します。このテーブルにはターゲット・データベース内のオブジェクトごとのページ使用率とエラー統計の情報が格納されます。
- `sp_dbcc_statisticsreport` が `object_name` の番号を返すと、`dbcc checkstorage` オブジェクトが完了したあと、そのオブジェクトは削除されたことを意味します。

- `sp_dbcc_statisticsreport` は、データ型 5000-5024 について `dbcc_counters` テーブルに記録された値をレポートします。詳細については、『リファレンス・マニュアル：テーブル』の `dbcc_counters` を参照してください。

`sp_dbcc_statisticsreport` は、`bytes data`、`bytes used`、`overflow pages` に、すべてのパーティションとデバイスについて報告された値を合計した値をレポートします。

`sp_dbcc_statisticsreport` は、`count`、`max count`、`max size`、`max level` に、すべてのパーティションとデバイスについて報告された値の中から最大値をレポートします。

`sp_dbcc_statisticsreport` は、次の各ローに対し、ターゲット・データベースのオブジェクトで使われた各デバイスとパーティションの情報をレポートします。

- `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`、`page extent gaps` は、オブジェクトのデータ・ページがデータベース・デバイスにどのように分散されているかを示します。値が大きいことは、大きなバッファ・サイズの使用とデータ・プリフェッチの点で、効果が少ないことを示します。

- 複数の `dbcc checkstorage` オペレーションが同じ日にターゲット・データベースで実行されると、`sp_dbcc_statisticsreport` は、指定された時間前に完了した最後の `dbcc checkstorage` オペレーションの結果に基づいて、レポートを生成します。

## sp\_dbcc\_statisticsreport

---

|               |                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------|
| パーミッション       | sp_dbcc_statisticsreport のパーミッション・チェックは、細密なパーミッションの設定によって異なります。                                    |
| 細密なパーミッションが有効 | 細密なパーミッションが有効の場合、dbccdb (または dbccalt) のデータベース所有者であること、または指定したデータベースに対する report checkstorage 権限が必要。 |
| 細密なパーミッションが無効 | 細密なパーミッションが無効の場合、指定したデータベース名で有効なすべてのユーザが sp_dbcc_statisticsreport を実行できる。                          |

### 参照

**コマンド** [dbcc](#)

**dbcc ストアド・プロシージャ** [sp\\_dbcc\\_fullreport](#)、[sp\\_dbcc\\_summaryreport](#)、[sp\\_dbcc\\_updateconfig](#)

## sp\_dbcc\_summaryreport

**説明** 指定したデータベースについてサマリ・レポートを生成します。

**構文** sp\_dbcc\_summaryreport [dbname [, date [, op\_name [, display\_recommendations]]]]

**パラメータ**

**dbname**

レポートを生成するデータベースの名前を指定します。 *dbname* を指定しなければ、sp\_dbcc\_summaryreport は、dbccdb..dbcc\_operation\_log にあるデータベースすべてについてレポートを生成します。日付は *date* オプションで指定される日付と時刻以前になります。

**date**

dbcc checkstorage が実行された日付を指定します。日付を指定しなければ、sp\_dbcc\_summaryreport は、**ターゲット・データベース**で実行された最後の dbcc checkstorage オペレーションの日付を使います。このパラメータのデータ型は *datetime* です。 *date* に日付と時間の両方が指定されると、指定された時間以前に実行されたすべてのオペレーションのサマリ結果がレポートされます。日付が指定されなければ、すべてのオペレーションがレポートされます。

**opname**

オペレーションを指定します。 *opname* には、デフォルトの checkstorage または checkverify のどちらか (あるいはその両方) を指定できます。 *opname* を指定しなければ、すべてのオペレーションに関するレポートが生成されます。

**display\_recommendations**

sp\_dbcc\_recommendations によって生成される推奨事項のレポートを有効にします。

**例**

**例 1** sybsystemprocs データベースのサマリ・レポートを生成して、実行されたすべての dbcc checkstorage と dbcc checkverify オペレーションに関する情報を提供します。

```
sp_dbcc_summaryreport
```

```
DBCC Operation :checkstorage
```

| Database Name  | Start time  | End Time     | Operation ID |
|----------------|-------------|--------------|--------------|
| Hard Faults    | Soft Faults | Text Columns | Abort Count  |
| -----          |             |              |              |
| -----          |             |              |              |
| -----          |             |              |              |
| sybsystemprocs | 05/11/1999  | 14:53:11     | 14:53:32:163 |
| 0              | 0           | 0            | 0            |
| sa             |             |              |              |
| sybsystemprocs | 05/11/1999  | 14:55:06     | 14:55:29:200 |
|                |             |              | 2            |

## sp\_dbcc\_summaryreport

```

 0 0 0 0 sa
sybssystemprocs 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           |              |

**例 2** ユーザ・データベース testdb のサマリ・レポートを生成して、実行されたすべての dbcc checkstorage オペレーションに関する情報を提供します。このデータベースで実行されたオペレーションは dbcc checkstorage だけなので、レポートには dbcc checkverify 情報は示されません。

```
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                   | sa           |              |
| testdb        | 05/11/1999 14:56:28 | 14:56:40:666 | 3            |
| 0             | 0                   | sa           |              |

**例 3** sybssystemprocs データベースのサマリ・レポートを生成して、実行されたすべての dbcc checkverify オペレーションに関する情報を提供します。指定されているオペレーションは dbcc checkverify だけなので、レポートには dbcc checkstorage 情報は示されません。

```
1> sp_dbcc_summaryreport null, null, "checkverify"
```

```
2> go
```

DBCC Operation :checkverify

| Database Name | Start time | End Time    | Operation ID | Run Srl   |
|---------------|------------|-------------|--------------|-----------|
| Table Name    | Table Id   | Hard Faults | Soft Faults  | User Name |



```


testdb 08/31/2004 11:06:11 11:6:11:370 3 1
 NULL NULL 0 0 sa

```

(1 row affected)

**例 4** sybsystemprocs データベースのサマリ・レポートを生成して、実行されたすべての dbcc checkstorage オペレーションに関する情報を提供します。指定されているオペレーションは dbcc checkstorage だけなので、レポートには dbcc checkverify 情報は示されません。

```
sp_dbcc_summaryreport sybsystemprocs, null, "checkstorage"
```

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            | sa           |
| sybsystemprocs | 05/11/1999 14:55:06 | 14:55:29:200 | 2            |
| 0              | 0                   | 0            | sa           |
| sybsystemprocs | 05/11/1999 14:56:10 | 14:56:27:750 | 3            |
| 0              | 0                   | 0            | sa           |

**例 5** データベース my\_db のサマリ・レポートに、推奨される修復方法を追加します。

```
sp_dbcc_summaryreport @dbname = my_db,
@display_recommendations = 1
```

使用法

- sp\_dbcc\_summaryreport は、指定されたデータベースでの checkstorage または checkverify のどちらかのオペレーション、あるいはこの両方についてのサマリ・レポートを生成します。
- レポートは、チェックされたデータベースの名前、dbcc checkstorage の実行開始時刻と終了時刻、検出したソフトとハードのフォールト数を示します。
- Operation ID カラムには、指定されたデータベースに対する特定時間の dbcc checkstorage オペレーションの結果を識別する数字が入ります。レポートに提供される数字は、dbcc\_operation\_log テーブルの opid カラムから検索されます。詳細については、『システム管理ガイド』を参照してください。

- “Text Columns” カラムは、dbcc checkstorage が実行中に検出した null でない text カラムの数を示します。
- “Abort Count” カラムは、dbcc checkstorage のテーブル・チェックをアボートさせるようなエラーのあるテーブルの数を示します。エラーの詳細を調べるには、sp\_dbcc\_faultreport を実行します。

パーミッション

sp\_dbcc\_summaryreport のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

|               |                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------|
| 細密なパーミッションが有効 | 細密なパーミッションが有効の場合、dbccdb (または dbccalt) のデータベース所有者であること、または指定したデータベースに対する report checkstorage 権限が必要。 |
| 細密なパーミッションが無効 | 細密なパーミッションが無効の場合、指定したデータベース名で有効なすべてのユーザが sp_dbcc_summaryreport を実行できる。                             |

参照

**コマンド** dbcc

**dbcc ストアド・プロシージャ** [sp\\_dbcc\\_fullreport](#)、[sp\\_dbcc\\_statisticsreport](#)、[sp\\_dbcc\\_updateconfig](#)

## sp\_dbcc\_updateconfig

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 説明    | ターゲット・データベースの設定情報を使用して、dbccdb にある dbcc_config テーブルを更新します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 構文    | sp_dbcc_updateconfig <i>dbname</i> , <i>type</i> , " <i>str1</i> " [, " <i>str2</i> "]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| パラメータ | <p><i>dbname</i><br/>設定情報を更新するターゲット・データベースの名前です。デフォルト値を設定するには、<i>dbname</i> パラメータに null を指定します。</p> <p><i>type</i><br/>dbcc_types テーブルからタイプ名を指定します。表 4-2 (947 ページ) は、<i>type</i> の有効な値を示しています。</p> <p><i>str1</i><br/>dbcc_config テーブル内の指定された <i>type</i> を更新する 1 つめの設定値を指定します。表 4-2 (947 ページ) では、指定された <i>type</i> ごとに <i>str1</i> の予期値を説明しています。</p> <p><i>str2</i><br/>dbcc_config テーブル内の指定された <i>type</i> を更新する 2 つめの設定値を指定します。表 4-2 (947 ページ) では、指定された <i>type</i> ごとに <i>str2</i> の予期値を説明しています。</p>                                                                                                                                                                                                         |
| 例     | <p><b>例 1</b> pubs2 データベースを確認するときに dbcc checkstorage が使用するワーカー・プロセスの最大数によって、dbcc_config を更新します。ワーカー・プロセスの新しい最大数は 4 です。</p> <pre>sp_dbcc_updateconfig pubs2, "max worker processes", "4"</pre> <p><b>例 2</b> max worker processes を 2 に設定します。</p> <pre>sp_dbcc_updateconfig null, 'max worker processes', '2'</pre> <p><b>例 3</b> dbcc の名前付きキャッシュ pubs2_cache のサイズによって、dbcc_config を更新します。新しいサイズは 10K です。</p> <pre>sp_dbcc_updateconfig pubs2, "dbcc named cache", pubs2_cache, "10K"</pre> <p><b>例 4</b> pubs2 データベースの scan 作業領域の新しい名前によって、dbcc_config を更新します。新しい名前は scan_pubs2 です。この更新は、sp_dbcc_alterws を使用して scan 作業領域の名前を変更したあとに行います。</p> <pre>sp_dbcc_updateconfig pubs2, "scan workspace", scan_pubs2</pre> |

**例 5** pubs2 データベースの text 作業領域の新しい名前によって、dbcc\_config を更新します。新しい名前は text\_pubs2 です。この更新は、[sp\\_dbcc\\_alterws](#) を使用して text 作業領域の名前を変更したあとに行います。

```
sp_dbcc_updateconfig pubs2, "text workspace", text_pubs2
```

**例 6** pubs2 データベースの OAM カウント・スレッシュホールド値によって、dbcc\_config を更新します。新しい値は 5 です。

```
sp_dbcc_updateconfig pubs2, "OAM count threshold", "5"
```

**例 7** pubs2 データベースの I/O エラー・アボート値によって、dbcc\_config を更新します。新しい値は 3 です。

```
sp_dbcc_updateconfig pubs2, "IO error abort", "3"
```

**例 8** pubs2 データベースのリンケージ・エラー・アボート値によって、dbcc\_config を更新します。新しい値は 8 です。

```
sp_dbcc_updateconfig pubs2, "linkage error abort", "8"
```

**例 9** データベース my\_db に対して automatic workspace expansion を有効にします。

```
sp_dbcc_updateconfig my_db, "enable automatic workspace expansion", "1"
```

#### 使用法

- sp\_dbcc\_updateconfig は、ターゲット・データベースの dbcc\_config テーブルを更新します。
- ターゲット・データベースの名前が dbccdb で、データベース dbccalt が存在する場合、sp\_dbcc\_updateconfig は dbccalt の dbcc\_config テーブルを更新します。
- dbcc\_config にターゲット・データベースの名前がない場合、ほかの設定情報を更新する前に、sp\_dbcc\_updateconfig はターゲット・データベースの名前を加えてオペレーション・シーケンス番号を 0 に設定します。
- 指定の type に予期される値が数字である場合、sp\_dbcc\_updateconfig は、str1 および str2 に指定する値を数字に変換します。
- [表 4-2 \(947 ページ\)](#) では type に使用する有効なタイプ名と、タイプごとに str1 または str2 に使用する予期値を示しています。

表 4-2: type 名と予期値

| type 名                        | str1 または str2 に使用する予期値                                                                             |
|-------------------------------|----------------------------------------------------------------------------------------------------|
| dbcc named cache              | str1 で指定するキャッシュの名前。str2 で指定する新しいサイズ ( キロバイトかメガバイト ) または 2K ページの数。                                  |
| IO error abort                | str1 が指定する新しいエラー・カウント値。値は 0 より大きい数字にする。このタイプでは、str2 は使用しない。                                        |
| linkage error abort           | str1 で指定する新しいリンケージ・エラー・カウント値。値は 0 より大きい数字にする。このタイプでは、str2 は使用しない。                                  |
| max worker processes          | str1 が指定する新しいワーカー・プロセスの数。値は 0 より大きい数字にする。このタイプでは、str2 は使用しない。                                      |
| OAM count threshold           | str1 が指定する新しいスレッシュホールド・カウント値。値は 0 より大きい数字にする。このタイプでは、str2 は使用しない。                                  |
| scan workspace                | str1 には scan 作業領域の新しい名前を指定する。このタイプでは、str2 は使用しない。                                                  |
| text workspace                | str1 には text 作業領域の新しい名前を指定する。このタイプでは、str2 は使用しない。                                                  |
| automatic workspace expansion | 各セグメントに適切な領域がある場合に、checkstorage が作業領域を自動的に拡張できるようにする。デフォルト値 1 を指定すると作業領域の自動拡張が有効になり、0 を指定すると無効になる。 |

- OAM count threshold パラメータは、実際のロー・カウントと (OAM ページでレポートされる) ロー・カウントの差が何パーセントを超えたら、dbcc checkstorage がエラー 100025、row count error を生成するかを示します。通常、OAM count threshold はデフォルト値の 2% のままにしておいてかまいません。
- type の名前と値の詳細については、『システム管理ガイド』を参照してください。

## パーミッション

sp\_dbcc\_updateconfig のパーミッション・チェックは、細密なパーミッションの設定によって異なります。

|               |                                                                                                      |
|---------------|------------------------------------------------------------------------------------------------------|
| 細密なパーミッションが有効 | 細密なパーミッションが有効の場合、dbccdb ( または dbccalt ) のデータベース所有者であること、または指定したデータベースに対する manage checkstorage 権限が必要。 |
| 細密なパーミッションが無効 | 細密なパーミッションが無効の場合、指定したデータベースのデータベース所有者か、sa_role を持つユーザであることが必要。                                       |

## 参照

コマンド [dbcc](#)

dbcc ストアド・プロシージャ [sp\\_dbcc\\_alterws](#)、[sp\\_dbcc\\_evaluatedb](#)

システム・プロシージャ [sp\\_plan\\_dbccdb](#)



# 索引

## 記号

- “ ” (引用符)
  - パラメータ値を囲む 4, 859
  - 予約語を囲む 133
  - 一重引用符、**quoted\_identifier** 140
- () (カッコ)
  - ユーザ定義データ型 63
- .. (ドット) のデータベース・オブジェクト名での使用 43
- , (カンマ)
  - ユーザ定義データ型 63
- @@**thresh\_hysteresis** グローバル変数  
スレッシュホールドの位置 59

## 数字

- 0 リターン・ステータス 1, 858
- 2 フェーズ・コミット
  - probe プロセス 699
- 7 ビット端末、**sp\_helpsort** 出力 491
- 8 ビット端末、**sp\_helpsort** 出力 491

## A

- abort tran on log full** データベース・オプション 239
- adhoc** 監査オプション 74
- all** 監査オプション 74
- allow nulls by default** データベース・オプション 239
- ALS (非同期ログ・サービス) 448
  - sp\_helpdb** 450
- alter database** コマンド
  - sp\_dbremap** 250
- alter** 監査オプション 74
- ANYENGINE** エンジン・グループ 26

## ASCII 文字

- sp\_checknames** 125
- at 記号 (@) とプロシージャのパラメータ 4
- auto identity** データベース・オプション 239

## B

- Backup Server
  - 『ASE ユーティリティ・ガイド』参照情報 489
  - ダンプされる量、指定 340
  - 複数 57
  - ボリューム処理メッセージ 838-841
- bcp**
  - sp\_audit** 74
- bcp** (バルク・コピー・ユーティリティ)
  - select into/bulkcopy/pllsort** 242
  - 高速バージョン 242
- bind** 監査オプション 74

## C

- checkpoint** コマンド
  - データベース・オプションの設定 234
- cmdtext** 監査オプション 74
- CPU 使用量、モニタ 597
- create database** コマンド
  - sp\_logdevice** と比較した **log on** 555
- create** 監査オプション 74

## D

- dbaccess** 監査オプション 74
- dbcc** (データベース一貫性チェック)
  - スクリプトと **sp\_checkreswords** 134
  - 領域の割り付け 654

**dbcc** 監査オプション 74  
*dbccdb* データベース  
    **dbcc checkstorage** 履歴の削除 915  
    I/O 統計のレポート 917  
    アロケーション統計のレポート 937  
    作業領域サイズの変更 908  
    作業領域の作成 911  
    詳細についてのレポート 929  
    ストアド・プロシージャ、使用 905  
    設定情報のレポート 910, 923, 929  
    ターゲット・データベース情報の削除 913  
    フォールト情報のレポート 917, 923  
    包括情報のレポート 929  
*dbccdb* データベースからのレポート  
    I/O 統計 917  
    アロケーション統計 937  
    詳細 929  
    設定情報 910, 923, 929  
    フォールト情報 917, 923  
    包括情報 929  
DB-Library プログラムと識別子名の変更 134  
**dbo use only** データベース・オプション  
    **sp\_dboption** 239  
**ddl in tran** データベース・オプション 240  
*default* セグメント  
    削除 329  
    マッピング 52  
**defncopy** コーティリティ・コマンド 133  
**delete** 監査オプション 74  
**disk** 監査オプション 74  
DLL (ダイナミック・リンク・ライブラリ)、  
    アンロード 401  
**drop** 監査オプション 74  
dsync 設定 266  
**dump transaction** コマンド  
    **trunc log on chkpt** 244  
**dump** 監査オプション 74

## E

**errors** 監査オプション 75  
**exec\_procedure** 監査オプション 75  
**exec\_trigger** 監査オプション 75

**exp\_row\_size** オプション  
    **sp\_chgattribute** 145  
    **sp\_help** レポート 418  
**expand\_down** パラメータ  
    **sp\_activeroles** 6  
    **sp\_displayroles** 284  
**external engine auto start** オプション  
    **sp\_serveroption** 713

## F

*fid* (ファミリー ID) 番号 381  
    **sp\_lock** レポート 549  
**from** キーワード、句の中に表示されるオブジェクトの **sp\_tables** リスト 883  
**func\_dbaccess** 監査オプション 75  
**func\_obj\_access** 監査オプション 75  
**futureonly** オプション  
    **sp\_bindefault** 94  
    **sp\_bindrule** 103, 104  
    **sp\_unbindefault** 826  
    **sp\_unbindrule** 832

## G

**grant** オプション  
    **sp\_helprotect** 478  
    **sp\_role** 704  
**grant** 監査オプション 75

## H

**holdlock** キーワード  
    **select** 548

## I

I/O  
    **concurrency\_opt\_threshold** 146  
    サイズの設定 658  
    使用量の統計 698



制限 10  
 ログ・サイズ 562

## ID

代替 18

## IDENTITY カラム

自動 239, 244  
 データベース・オプション、使用 241  
 ユニークでないインデックス 241

ID、時間範囲 15

ID、ユーザ

「ログイン」参照

*image* データ型、サイズ 778

**insert** 監査オプション 75

interfaces ファイル

**sp\_addserver** 55  
 サーバ名の変更 139

## J

Java 項目と **sp\_helpjava**

システム・プロシージャ 465

## L

*LASTONLINE* エンジン・グループ 26

LDAP ユーザ認証

エンジンあたりの LDAP 記述子数 532  
 エンジンあたりの最大 *ldapua* ネイティブ・ス  
 レッド数 531

LDAP、URL 検索文字列 526

**load**

インデックスの再構築 665

**load** 監査オプション 75

**log on** オプション

**create database** と **sp\_logdevice** 555

**login** 監査オプション 75

**login\_locked** 監査オプション 75

**logout** 監査オプション 75

*logsegment* ログ記憶領域、削除 329

## M

*master* データベース

**sp\_dboption** 233  
 システム・プロシージャ・テーブル 5  
 スレシヨルド 60, 588  
**sp\_checkreswords** 131

**max\_rows\_per\_page** オプション

**sp\_chgattribute** 145  
**sp\_relimit** 145

*model* データベース

データベース・オプションの変更 243

*model* データベース、データベース・オプション  
 の変更 233

MRU 置換方式

無効化 117

## N

**negotiated logins** オプション

**sp\_serveroption** 715

**net password encryption** オプション

**sp\_serveroption** 713

not null 値

**sp\_addtype** 64  
 ユーザ定義データ 64

null 値

**sp\_addtype** 63  
 ユーザ定義データ型 63

## O

ODBC (Open Database Connectivity)

API データ型 860

ODBC。「ODBC (Open Database Connectivity)

API データ型」参照

Open Client アプリケーション

接続セキュリティ 49

**optdiag** ユーティリティ

インメモリ統計値のフラッシュ 390

**output** オプション

**sp\_getmessage** 403

## P

**p\_cleanpwdchecks** ストアド・プロシージャ 151  
**PC DB-Library**。「DB-Library プログラム」参照  
**probe** プロセス、2 フェーズ・コミットの設定 699  
 “probe” ログイン・アカウント 699  
 “public” グループ  
   **sp\_addgroup** 34  
   **sp\_adduser** 70  
   **sp\_changegroup** 123  
   **sp\_helpgroup** レポート 460  
 情報レポート 460  
 「グループ」参照

## R

**read only** データベース・オプション  
   **sp\_dboption** 242  
   **sp\_setsuspect\_granularity** 732  
**reference** 監査オプション 75  
**reindex** オプション、dbcc  
   **sp\_indsuspect** 511  
**reservepagegap** オプション  
   **sp\_chgattribute** 145  
   **sp\_help** レポート 418  
 Resource Governor 577  
**revoke** 監査オプション 75  
**rpc** 監査オプション 75

## S

**security** 監査オプション 76  
**select into/bulkcopy/pllort** データベース・オプション  
 トランザクション・ログ 242  
**select** 監査オプション 76  
**server cost** オプション  
   **sp\_serveroption** 714  
**server login** オプション  
   **sp\_serveroption** 715  
**set** コマンド  
   **sp\_setlangalias** および **language** オプション 721  
**setting**  
 監査オプション 74  
**setuser** 監査オプション 76

**single user** データベース・オプション  
   **sp\_dboption** 243  
**size of auto identity column**  
 設定パラメータ 239, 244  
**sp\_activeroles** システム・プロシージャ 6-7  
**sp\_add\_qpgroup** システム・プロシージャ 8  
**sp\_add\_resource\_limit**  
 システム・プロシージャ 9-13  
**sp\_add\_time\_range**  
 システム・プロシージャ 14-17  
**sp\_addalias** システム・プロシージャ 18-19  
**sp\_addauditrecord**  
 システム・プロシージャ 20-22  
**sp\_addaudittable** システム・プロシージャ 23-24  
**sp\_addengine** システム・プロシージャ 25-26  
**sp\_addexclass** システム・プロシージャ 27-28  
**sp\_addextendedproc**  
 システム・プロシージャ 29-30  
**sp\_addexternlogin**  
 システム・プロシージャ 31-33  
**sp\_addgroup** システム・プロシージャ 34-35  
**sp\_addlanguage** システム・プロシージャ 36-38  
**sp\_addmessage** システム・プロシージャ 40-42  
**sp\_addobjectdef** システム・プロシージャ 43-46  
**sp\_adremotelogin**  
 システム・プロシージャ 47-50  
**sp\_addsegment** システム・プロシージャ 51-53  
 データとログが混在するデータベース 52  
**sp\_addserver** システム・プロシージャ 54-57  
**sp\_addthreshold** システム・プロシージャ 58-62  
**sp\_addtype** システム・プロシージャ 63-66  
**sp\_addumpdevice**  
 システム・プロシージャ 67-69  
**sp\_adduser** システム・プロシージャ 70-71  
**sp\_altermessage** システム・プロシージャ 72-73  
**sp\_audit** システム・プロシージャ 74-81  
**sp\_autoconnect** システム・プロシージャ 82-83  
**sp\_autoformat** システム・プロシージャ 84-88  
**sp\_bindcache** システム・プロシージャ 89-93  
**sp\_bindefault** システム・プロシージャ 94-96  
 create default 95  
**sp\_bindexclass** システム・プロシージャ 97-100  
**sp\_bindmsg** システム・プロシージャ 101-102  
**sp\_bindrule** システム・プロシージャ 103-105  
**sp\_cacheconfig**  
 システム・プロシージャ 106-116

- sp\_cachestrategy** システム・プロシージャ 117-120
- sp\_changedbowner** システム・プロシージャ 121-122
- sp\_changegroup** システム・プロシージャ 123-124
- sp\_dropgroup** 315
- 「sp\_dropgroup」参照
- sp\_checknames** システム・プロシージャ 125-127
- sp\_checkreswords** システム・プロシージャ 128-141
- リターン・ステータス 131
- sp\_checksource** システム・プロシージャ 142-144
- sp\_chgattribute** システム・プロシージャ 145-150
- sp\_clearpsex** システム・プロシージャ 152-153
- sp\_clearstats** システム・プロシージャ 154-155
- sp\_client\_addr** システム・プロシージャ 156
- sp\_cluster\_logical** ストアド・プロシージャ 159
- sp\_cmp\_all\_qplans** システム・プロシージャ 183
- sp\_cmp\_qplans** システム・プロシージャ 186
- sp\_column\_privileges** カタログ・ストアド・プロシージャ 862-863
- sp\_columns** カタログ・ストアド・プロシージャ 864-866
- sp\_datatype\_info** 868
- データ型コード番号 860
- sp\_commonkey** システム・プロシージャ 188-189
- sp\_companion** システム・プロシージャ 190-193
- sp\_config\_dump** システム・プロシージャ 195-198
- sp\_configure** システム・プロシージャ 199-207
- 表示レベルの設定 275
- sp\_copy\_all\_qplans** システム・プロシージャ 208
- sp\_copy\_qplan** システム・プロシージャ 210-211
- sp\_countmetadata** システム・プロシージャ 212
- sp\_cursorinfo** システム・プロシージャ 215-217
- sp\_databases** カタログ・ストアド・プロシージャ 867
- sp\_datatype\_info** カタログ・ストアド・プロシージャ 868
- sp\_dbcc\_alterws** ストアド・プロシージャ 908-909
- sp\_dbcc\_configreport** ストアド・プロシージャ 910
- sp\_dbcc\_createws** ストアド・プロシージャ 911-912
- sp\_dbcc\_deletedb** ストアド・プロシージャ 913-914
- sp\_dbcc\_deletehistory** ストアド・プロシージャ 915-916
- sp\_dbcc\_differentialreport** ストアド・プロシージャ 917-918
- sp\_dbcc\_evaluatedb** ストアド・プロシージャ 919-920
- sp\_dbcc\_faultreport** ストアド・プロシージャ 923-928
- sp\_dbcc\_fullreport** ストアド・プロシージャ 929
- sp\_dbcc\_runcheck** ストアド・プロシージャ 935-936
- sp\_dbcc\_statisticsreport** ストアド・プロシージャ 937-940
- sp\_dbcc\_summaryreport** ストアド・プロシージャ 941-944
- sp\_dbcc\_updateconfig** ストアド・プロシージャ 945-947
- sp\_dbextend** システム・プロシージャ 218-227
- sp\_dboption** システム・プロシージャ 229-246
- sp\_dbrecovery\_order** システム・プロシージャ 247-249
- sp\_dbremap** システム・プロシージャ 250-251
- sp\_defaultloc** システム・プロシージャ 252-254
- sp\_deletesmobj** ストアド・プロシージャ、構文と使用法 255
- sp\_depends** システム・プロシージャ 258-265
- データベース・オブジェクト所有者 258
- sp\_deviceattr** システム・プロシージャ 266-268
- sp\_diskdefault** システム・プロシージャ 269-270
- sp\_displayaudit** システム・プロシージャ 271-274
- sp\_displaylevel** システム・プロシージャ 275-276
- sp\_displaylogin** システム・プロシージャ 277-283
- sp\_displayroles** システム・プロシージャ 284
- sp\_drop\_all\_qplans** システム・プロシージャ 297

- [sp\\_drop\\_qpgroup](#) システム・プロシージャ 298
- [sp\\_drop\\_qplan](#) システム・プロシージャ 299
- [sp\\_drop\\_resource\\_limit](#) システム
  - ・プロシージャ 300-303
- [sp\\_drop\\_time\\_range](#) システム・プロシージャ 304
- [sp\\_dropalias](#) システム・プロシージャ 295-296
- [sp\\_dropdevice](#) システム・プロシージャ 305-306
- [sp\\_dropengine](#) システム・プロシージャ 307
- [sp\\_dropexeclss](#) システム・プロシージャ 309
- [sp\\_dropextendedproc](#) システム・プロシージャ 310
- [sp\\_dropexternlogin](#) システム
  - ・プロシージャ 311-312
- [sp\\_dropglockpromote](#) システム
  - ・プロシージャ 313-314
- [sp\\_dropgroup](#) システム・プロシージャ 315
- [sp\\_dropkey](#) システム・プロシージャ 316-317
- [sp\\_droplanguage](#) システム・プロシージャ 318-319
- [sp\\_dropmessage](#) システム・プロシージャ 321
- [sp\\_dropobjectdef](#) システム・プロシージャ 322-323
- [sp\\_dropremotelogin](#) システム
  - ・プロシージャ 324-325
- [sp\\_droprowlockpromote](#) システム
  - ・プロシージャ 326
- [sp\\_dropsegment](#) システム・プロシージャ 328-329
  - [sp\\_placeobject](#) 329
- [sp\\_dropserver](#) システム・プロシージャ 330-331
- [sp\\_droptreshold](#) システム・プロシージャ 332-333
- [sp\\_droptype](#) システム・プロシージャ 334
- [sp\\_dropuser](#) システム・プロシージャ 335-336
- [sp\\_dumpoptimize](#) システム
  - ・プロシージャ 337-339, 340-345
- [sp\\_encryption](#) システム・プロシージャ 346-359
- [sp\\_engine](#) システム・プロシージャ 362-365
- [sp\\_estspace](#) システム・プロシージャ 368-373
- [sp\\_export\\_qpgroup](#) システム・プロシージャ 374
- [sp\\_extendsegment](#) システム・プロシージャ 376-377
  - [create index](#) 376
  - [create table](#) 376
- [sp\\_extengine](#) システム・プロシージャ 378-379
- [sp\\_extrapwdchecks](#) 380
- [sp\\_familylock](#) システム・プロシージャ 381-383
- [sp\\_find\\_qplan](#) システム・プロシージャ 384-386
- [sp\\_fixindex](#) システム・プロシージャ 387-389
- [sp\\_fkeys](#) カタログ・ストアド
  - ・プロシージャ 869-870
- [sp\\_flushstats](#) システム・プロシージャ 390-391
- [sp\\_forceonline\\_db](#) システム
  - ・プロシージャ 392-393
- [sp\\_forceonline\\_object](#) システム
  - ・プロシージャ 394-396
- [sp\\_forceonline\\_page](#) システム
  - ・プロシージャ 397-398
- [sp\\_foreignkey](#) システム・プロシージャ 399-400
- [sp\\_freedll](#) システム・プロシージャ 401
- [sp\\_getmessage](#) システム・プロシージャ 403-404
- [sp\\_grantlogin](#) システム・プロシージャ 405
- [sp\\_ha\\_admin](#) システム・プロシージャ 407-408
  - [installhasvss](#) 407
- [sp\\_help](#) システム・プロシージャ 409-419
- [sp\\_help\\_qpgroup](#) システム
  - ・プロシージャ 424-425
- [sp\\_help\\_qplan](#) システム・プロシージャ 426-427
- [sp\\_help\\_resource\\_limit](#) システム
  - ・プロシージャ 420-423
- [sp\\_helppartition](#) システム・プロシージャ 429
- [sp\\_helpcache](#) システム・プロシージャ 434-435
- [sp\\_helpcomputedcolumn](#) システム
  - ・プロシージャ 436
- [sp\\_helpconfig](#) システム・プロシージャ 437-442
- [sp\\_helpconstraint](#) システム
  - ・プロシージャ 443-447
- [sp\\_helppdb](#) システム・プロシージャ 448-452
  - [tempdb](#) 450
- [sp\\_helpdevice](#) システム・プロシージャ 453-455
- [sp\\_helpextendedproc](#) システム
  - ・プロシージャ 456-457
- [sp\\_helpexternlogin](#) システム・プロシージャ 458
- [sp\\_helpgroup](#) システム・プロシージャ 460-461
- [sp\\_helpindex](#) システム・プロシージャ 462-464
- [sp\\_helpjava](#) システム・プロシージャ 465-467
- [sp\\_helpjoins](#) システム・プロシージャ 468-469
- [sp\\_helpkey](#) システム・プロシージャ 470-471
- [sp\\_helplanguage](#) システム
  - ・プロシージャ 472-473
- [sp\\_helplog](#) システム・プロシージャ 474
- [sp\\_helpobjectdef](#) システム
  - ・プロシージャ 475-476
- [sp\\_helpremotelogin](#) システム・プロシージャ 477
- [sp\\_helpprotect](#)
  - [暗号化カラム](#) 482
- [sp\\_helpprotect](#) システム・プロシージャ 478-483

- sp\_helpsegment** システム  
 ・ プロシージャ 484-488
- sp\_helpserver** システム・プロシージャ 489-490
- sp\_helpsort** システム・プロシージャ 491-492
- sp\_helptext** システム・プロシージャ 493-499
- sp\_helptext** ストアド・プロシージャ、変更 497
- sp\_helpthreshold** システム  
 ・ プロシージャ 502-503
- sp\_helpuser** システム・プロシージャ 504-506
- sp\_hidetext** システム・プロシージャ 507
- sp\_import\_qpgroup** システム  
 ・ プロシージャ 509-510
- sp\_indsuspect** システム・プロシージャ 511
- sp\_jreconfig** 512
- sp\_ldapadmin** システム・プロシージャ 526-533
- sp\_listener** システム・プロシージャ 534
- sp\_listsuspect\_db** システム・プロシージャ 538
- sp\_listsuspect\_object** システム  
 ・ プロシージャ 539-540
- sp\_listsuspect\_page** システム  
 ・ プロシージャ 541
- sp\_lmconfig** システム・プロシージャ 542-545
- sp\_lock** システム・プロシージャ 546-551
- sp\_logdevice** システム・プロシージャ 555-557  
 create database への log on 拡張機能 555
- sp\_loginconfig** システム  
 ・ プロシージャ 558-559
- sp\_logininfo** システム・プロシージャ 560-561
- sp\_logiosize** システム・プロシージャ 562
- sp\_maplogin** システム・プロシージャ 568-569
- sp\_metrics** システム・プロシージャ 573-574
- sp\_modify\_resource\_limit** システム  
 ・ プロシージャ 575-578
- sp\_modify\_time\_range** システム  
 ・ プロシージャ 579-580
- sp\_modifystats** システム  
 ・ プロシージャ 582-585
- sp\_modifythreshold** システム  
 ・ プロシージャ 586-590
- sp\_monitor** システム・プロシージャ 591-599  
 @@connections 598
- sp\_monitorconfig** システム  
 ・ プロシージャ 600-607
- sp\_object\_stats** システム  
 ・ プロシージャ 609-612
- sp\_passthru** システム・プロシージャ 629-631
- sp\_passwordpolicy** 633
- sp\_pciconfig** 648
- sp\_pkeys** カタログ・ストアド  
 ・ プロシージャ 871
- sp\_placeobject** システム・プロシージャ 653-654
- sp\_plan\_dbccdb** システム  
 ・ プロシージャ 655-657
- sp\_poolconfig** システム・プロシージャ 658-664
- sp\_post\_xoload** システム  
 ・ プロシージャ 665-666  
 インデックスの再構築 665
- sp\_primarykey** システム・プロシージャ 667-668
- sp\_foreignkey** 399
- sp\_processmail** システム  
 ・ プロシージャ 669-671
- sp\_procxmode**  
 連鎖トランザクション・モード 673
- sp\_procxmode** システム・プロシージャ 672-674
- sp\_querysmobj** ストアド・プロシージャ、  
 構文と使用法 675
- sp\_recompile** システム・プロシージャ 678-679
- sp\_remotoption** システム  
 ・ プロシージャ 684-686
- sp\_remotesql** システム・プロシージャ 687-689
- sp\_rename** システム・プロシージャ 690-692
- sp\_rename\_qpgroup** システム  
 ・ プロシージャ 693
- sp\_renamedb** システム  
 ・ プロシージャ 136, 695-697
- sp\_reportstats** システム・プロシージャ 698-699
- sp\_revokelogin** システム・プロシージャ 702
- sp\_role** システム・プロシージャ 704-705
- sp\_sendmsg** システム・プロシージャ 710-712
- sp\_server\_info** カタログ・ストアド  
 ・ プロシージャ 872-874
- sp\_tables** 883
- sp\_serveroption**  
 external engine auto start 713  
 negotiated logins 715  
 server costt 714  
 server login 715
- sp\_serveroption** システム  
 ・ プロシージャ 713-718
- sp\_set\_qplan** システム・プロシージャ 719-720
- sp\_setlangalias** システム・プロシージャ 721
- sp\_setpglockpromote** システム  
 ・ プロシージャ 722-725

- sp\_setpsex** システム・プロシージャ 726  
**sp\_setrowlockpromote** システム・プロシージャ 728  
**sp\_setsuspect\_granularity** システム  
   ・プロシージャ 732-735  
**sp\_setsuspect\_threshold** システム  
   ・プロシージャ 736-737  
**sp\_shmdumpconfig** システム・プロシージャ 739  
**sp\_showcontrolinfo** システム・プロシージャ 747  
**sp\_showexeclass** システム・プロシージャ 750  
**sp\_showplan** システム・プロシージャ 760  
**sp\_showpsex** システム・プロシージャ 762  
**sp\_spaceused** システム・プロシージャ 778-781  
**sp\_special\_columns** カタログ・ストアド  
   ・プロシージャ 875  
**sp\_sproc\_columns** カタログ・ストアド  
   ・プロシージャ 877  
   データ型コード番号 860  
**sp\_ssladmin** システム・プロシージャ 782-786  
**sp\_statistics** カタログ・ストアド  
   ・プロシージャ 879  
**sp\_stored\_procedures** カタログ・ストアド  
   ・プロシージャ 881  
   **sp\_server\_info** 情報 873  
**sp\_syntax** システム・プロシージャ 787-788  
**sp\_sysmon** システム・プロシージャ 789-795  
**sp\_table\_privileges** カタログ・ストアド  
   ・プロシージャ 882  
**sp\_tables** カタログ・ストアド・プロシージャ 883  
   **sp\_server\_info** 情報 873  
**sp\_tempdb** システム・プロシージャ 797-807  
**sp\_thresholdaction** システム  
   ・プロシージャ 810-812  
   スレッシュホールド・プロシージャ 59, 587  
**sp\_transactions** システム・プロシージャ 814-820  
**sp\_unbindcache** システム・プロシージャ 821-823  
**sp\_unbindcache\_all** システム  
   ・プロシージャ 824-825  
**sp\_unbinddefault** システム・プロシージャ 826-827  
**sp\_unbindexclass** システム・プロシージャ 828  
**sp\_unbindmsg** システム・プロシージャ 831  
**sp\_unbindrule** システム・プロシージャ 832-834  
**sp\_version** システム・プロシージャ 835-837  
**sp\_volchanged** システム・プロシージャ 838-841  
**sp\_webservices** システム・プロシージャ 842-851  
**sp\_who** システム・プロシージャ 852-855  
   返されるカラム 854  
  
**spid** 番号  
   **sp\_who** 出力 854  
**spt\_committab** テーブル 5  
**spt\_datatype\_info** テーブル 860  
**spt\_datatype\_info\_ext** テーブル 860  
**spt\_monitor** テーブル 5  
**spt\_server\_info** テーブル 860  
**spt\_values** テーブル 5  
SQL 規格  
   SQL パターン一致 860  
   ユーザ定義データ型 64  
**suspect** (疑わしい) インデックス  
   強制的にオンラインにする 394, 539  
**suspect** データベース、リスト 538  
**sybdiagdb** データベース 441  
**sybsyntax** データベース 788  
**sysalternates** テーブル  
   **sp\_dropalias** 295  
**sysusers** テーブル 18  
   エイリアス 18  
**syscomments** テーブル  
   ソース・テキスト 497  
**sysconstraints** テーブル  
   **sp\_bindmsg** 101  
**sysdevices** テーブル 269, 454  
**syskeys** テーブル  
   **sp\_dropkey** 316  
   **sp\_foreignkey** 399  
   **sp\_primarykey** 667  
**syslanguages** テーブル 472  
   **sp\_droplanguage** 318  
**syslkstats** テーブル 611  
**syslogs** テーブル 555  
   別のデバイスへの配置 555  
**sysmessages** テーブル  
   エラー・メッセージ・テキスト 403  
**sysremotelogins** テーブル 47-50, 330  
   **sp\_droptremotelogin** 324  
**sysresourcelimits** テーブル  
   **sp\_help\_resource\_limit** 422  
   ログイン・セッションに適用できる制限値 12  
**syssservers** テーブル  
   **sp\_addserver** 54  
   **sp\_helpserver** 489  
**sysessions**  
   古いエントリの削除 407

*systabstats* テーブル  
統計値のフラッシュ 390

*system* セグメント  
削除 329  
マッピング 52

*systimeranges* テーブル  
ID 番号の保管 15  
範囲名の保管 9

*systypes* テーブル 334

*sysusermessages* テーブル  
**sp\_dropmessage** 321  
エラー・メッセージ・テキスト 403

*sysusers* テーブル  
*sysalternates* テーブル 18

## T

**table\_access** 監査オプション 76

tempdb  
**sp\_helpdb** 450  
セッションごとに使用される  
ページ数の制限 10, 576

tempdb データベース  
**auto\_identity** データベース・オプション 239  
**unique auto\_identity index** データベース・オプション 244

tempdb\_space 10, 576

text  
**defncopy** 133  
ユーザ定義メッセージ 40

text データ型  
記憶領域のサイズ 778

Transact-SQL  
予約語 131

**true | false** 句  
**sp\_dboption** 229  
**sp\_remotoption** 684

**trunc log on chkpt** データベース  
・オプション 243

**truncate** 監査オプション 76

trusted モード  
リモート・ログイン 49

## U

UDP のメッセージ機能 710

**unbind** 監査オプション 76

**unique auto\_identity index** データベース  
・オプション 244

**update** 監査オプション 76

us\_english 言語 37

**use message confidentiality** サーバ  
・オプション 715

**use message integrity** サーバ・オプション 715

## V

**view\_access** 監査オプション 76

## W

**writetext** コマンド  
**select into/bulkcopy/pllsort** データベース・オプション 242

## X

XP Server 886  
解放、メモリ 401

**xp\_cmdshell context** 設定パラメータ 888

**xp\_cmdshell** システム拡張ストア  
・プロシージャ 887

**xp\_deletemail** システム拡張ストア  
・プロシージャ 890  
**sp\_processmail** 670

**xp\_enumgroups** システム拡張ストア  
・プロシージャ 891

**xp\_findnextmsg** システム拡張ストア  
・プロシージャ 892  
**sp\_processmail** 670

**xp\_logevent** システム拡張ストア  
・プロシージャ 893

**xp\_readmail** システム拡張ストア  
・プロシージャ 894  
**sp\_processmail** 670

**xp\_sendmail** システム拡張ストアド  
 ・プロシージャ 897  
**sp\_processmail** 670  
**xp\_startmail** システム拡張ストアド  
 ・プロシージャ 901  
**xp\_stopmail** システム拡張ストアド  
 ・プロシージャ 903

## あ

アカウントティング、チャージバック  
**sp\_clearstats** 154  
 アカウント。「ログイン」参照  
 アクション  
 リソース制限情報 421  
 リソース制限値の指定 11  
 リソース制限の変更 576

## 値

**sp\_server\_info** 872  
 アプリケーション  
 リソース制限情報 420  
 リソース制限値の適用 9  
 リソース制限の削除 300  
 リソース制限の変更 575

## 暗号化

コンパイル済みオブジェクトの  
 ソース・テキスト 507  
 リバース 508

## 暗号化カラム

**sp\_helprotect** 482

## い

依存性、データベース・オブジェクト  
 名前の変更 133  
**sp\_depends** システム・プロシージャ 258-265  
 再コンパイル 691  
 移動  
 新しいグループへのユーザの移動 123  
 インデックス 653  
 テーブル 653  
 トランザクション・ログ 555

意図的なテーブル・ロック 381, 548  
 インデックス

**sp\_statistics** 879  
**sp\_placeobject** 領域の割り付け 653-654  
 suspect (疑わしい) 511  
 情報 462  
 使用領域 780  
 データ・キャッシュからのバインド解除 821  
 データ・キャッシュへのバインド 89  
 名前の変更 135, 690-692  
 順序、**sp\_helpindex** 464  
**sp\_checknames** 126  
**sp\_checkreswords** 131  
 必要な領域と時間の見積もり 368  
 ユニークでないインデックスでの  
 IDENTITY カラム 241

## インデックスの再構築

**sp\_post\_xoload** 665

## インデックス・ページ

ロック 550

## 引用符 (" ")

一重引用符、**quoted\_identifier** 140  
 パラメータ値を囲む 4, 859  
 予約語を囲む 133

## 引用符付き識別子

使用 132, 140-141  
 テスト 133

## う

ウォッシュ・エリア  
 設定 663  
 デフォルト 663  
 疑わしいパーティション、プラットフォーム間の  
 ダンプとロード 666  
 疑わしいページ  
 オンラインにする 392-393, 397-398  
 リカバリ時のアイソレート 736-737  
 リスト作成 541



## え

- エイリアス、言語
  - 定義 36
  - 割り当て 721
- エイリアス、サーバ 55
- エイリアス、ユーザ
  - sysalternates* テーブル 18, 295
  - 異なる名前の割り当て 70
  - 削除 295–296, 335
  - データベース所有権の譲渡 121
  - ヘルプ 505
  - 割り当て 18–19
  - 「ログイン」「ユーザ」参照
- エラー
  - 数 597
- エンジンあたりの記述子数 532
- エンジンあたりの最大 *ldapua* ネイティブ・スレッド数 531

## お

- オーバヘッド、データ・キャッシュ 114
- オブジェクト名、データベース
  - sp\_checknames* 126
  - sp\_checkreswords* 131
- オブジェクト。「データベース・オブジェクト」  
「データベース」参照
- オブジェクト所有者。「データベース・オブジェクト所有者」参照
- オプション
  - リモート・サーバ 713–718
  - リモート・ログイン 684–686
  - 「設定パラメータ」参照
  - 「ダンプ設定パラメータ」参照
- オペティミスティック・インデックス・ロック 145, 146, 148, 409, 413
- オペレーティング・システム・コマンド 887
- オペレーティング・システム・コマンド (*xp\_cmdshell*) のユーザ
  - ・コンテキスト 888
- オペレーティング・システム
  - ・コマンドの実行 887

## か

- カーソル 215
- カーソルのネスト 215
- 階層
  - ロック・プロモーション
    - ・スレッショルド 723, 729
  - データ・キャッシュのバインド 90
  - ユーザ定義データ型 65
  - 役割、*sp\_activeroles* 6
  - 「優先度」参照
- 外部キー
  - sp\_fkeys* 情報 869
  - sp\_helpkey* 470
  - 削除 316
  - 挿入 399–400
- 拡張
  - セグメント 376
- 拡張ストアド・プロシージャ
  - 削除 310
  - 作成 29–30
  - 表示 456
- 数(量)
  - インデックス 212
  - オープン・オブジェクト 212
  - 制約ごとのメッセージ 101
  - sp\_countmetadada* 212
  - ユーザー一人あたりのグループ 123
- カスタム監査レコード 74
- カスタム・データ型。
  - 「ユーザ定義データ型」参照
- 仮想ページ番号 455
- カタログ・ストアド・プロシージャ 857–884
  - sp\_columns* 864–866
  - sp\_databases* 867
  - sp\_datatype\_info* 868
  - sp\_fkeys* 869–870
  - sp\_pkeys* 871
  - sp\_server\_info* 872–874
  - sp\_special\_columns* 875
  - sp\_sproc\_columns* 877
  - sp\_statistics* 879
  - sp\_stored\_procedures* 881
  - sp\_tables* 883
  - リスト 858

- 構文 859
- リターン・ステータス 858
- カッコ ()
  - ユーザ定義データ型 63
- 可変長文字データ、フォーマット 84-88
- カラム
  - sp\_unbindrule** によるルー  
ルのバインド解除 832-834
  - 依存性、検出 135
  - 外部キー 399-400, 869
  - 共通キー 188-189
  - ジョイン 468
  - データ型 864
  - デフォルト 94-96
  - デフォルトのバインド解除 826-827
  - sp\_who** 854
  - パーミッション 862
  - プライマリ・キー 667
  - ルール 103-105
- カラムのペア。「共通キー」「ジョイン」参照
- カラムのペア。「ジョイン」「キー」参照
- カラム名
  - sp\_checknames** 126
  - 変更 135, 690-692
- カラム・データ。「データ型」参照
- 監査
  - オプション、表示 271
  - 監査テーブルの追加 23
- 監査オプション
  - adhoc** 74
  - all** 74
  - alter** 74
  - bcp** 74
  - bind** 74
  - cmdtext** 74
  - create** 74
  - dbaccess** 74
  - dbcc** 74
  - delete** 74
  - disk** 74
  - drop** 74
  - dump** 74
  - errors** 75
  - exec\_procedure** 75
  - exec\_trigger** 75
  - func\_dbaccess** 75
  - func\_obj\_access** 75
  - grant** 75
  - insert** 75
  - load** 75
  - login** 75
  - login\_locked** 75
  - logout** 75
  - reference** 75
  - revoke** 75
  - rpc** 75
  - security** 76
  - select** 76
  - setting** 74
  - setuser** 76
  - table\_access** 76
  - truncate** 76
  - unbind** 76
  - update** 76
  - view\_access** 76
- 監査証跡、コメントの追加 20
- カンマ (,)
  - ユーザ定義データ型 63

## き

- キーワード
  - 識別子として使用 128
- キー、テーブル
  - syskeys** テーブル 188, 399, 667
  - 削除 316
  - 情報 470
  - 「共通キー」「インデックス」参照
- 記憶領域管理プロパティ
  - sp\_chgattribute** 145
- 規制
  - オブジェクトの検索 263, 417
- キャッシュ、データ
  - logonly** タイプ 115
  - オーバーヘッド 114, 434
  - オブジェクトのバインド 89
  - オブジェクトのバインド解除 821
  - 削除 115

情報 110, 434  
 ステータス 112  
 すべてのオブジェクトのバインド解除 824  
 設定 106-116  
 メモリ・プール 658-664  
 リカバリ 110  
 競合、ロック  
**sp\_object\_stats** によるモニタ 609-612  
 共通キー  
 削除 316  
 ジョイン候補 469  
 定義 188-189  
 レポート 470-471  
 「外部キー」「ジョイン」  
 「プライマリ・キー」参照  
 共有ロー・ロック 550  
 共有ロック 381, 548

## く

クエリ  
**sp\_tables** 883  
 コンパイルと最適化 678  
 クエリ処理、**sp\_add\_resource\_limit** 9  
 区切り識別子  
 使用 132, 140-141  
 テスト 133  
 句読表記  
 引用符で囲む 4, 859  
 ユーザ定義データ型 63  
 位取り、データ型  
 ユーザ定義データ型 63  
 クラスタード・インデックス、  
*indid* は 1 と等しくない 464  
 クリア、アカウント統計 154-155  
 グループ  
**sp\_addgroup** 34-35  
 「"public" グループ」参照  
**sp\_adduser** プロシージャ 70  
 Windows NT ドメイン 891  
 削除 315  
 情報 460  
 変更 123-124

グローバル変数  
 「個々の変数名」参照  
**sp\_monitor** レポート 595

## け

言語、代替  
 名前の変更 137, 139  
*syslanguages* テーブル 472  
 インストール 36  
 エイリアス 721  
 言語モジュールを使用しない 36  
 公式名 721  
 削除 318-319  
 システム・メッセージ 403  
 情報 472  
**sp\_checkreswords** 132  
 日付フォーマット 36  
 メッセージの削除 321  
 ユーザ定義メッセージ 40  
 現在の使用量の統計 698-699  
 現在のデータベース  
**sp\_helpdb** 451  
 使用領域 778-781  
 現在のデータベース情報の取得 451  
 現在のロック、**sp\_lock**  
 システム・プロシージャ 546  
 検索  
 エラー・メッセージ・テキスト 403  
 オブジェクト情報 409  
 オブジェクトの依存性 258, 263  
 キャッシュのバインド 106, 434  
 言語 472  
 サーバ名 489  
 スレッシュホールド 502  
 制約 443  
 セグメント 484  
 設定パラメータ 437  
 データ型 409  
 データベース・オブジェクト 417  
 データベース・オプション 229  
 データベース設定 448

データベース内のユーザ 504  
 デバイス 453  
 パーティション情報 429  
 パーミッション 478  
 文字セット 491  
 予約語 128  
 リソース制限 420  
 検査制約  
 ソース・テキストの表示 493  
 名前の変更 690-692  
 ユーザ・メッセージのバインド 101

## こ

高可用性  
 Adaptive Server の設定 190  
 公式言語名 37, 721  
 「エイリアス」「言語、代替」参照  
 更新ロー・ロック 550  
 構文  
 カタログ・ストアド・プロシージャ 859  
 予約語の確認 131  
 コード  
 ODBC データ型 860  
 データ型 868  
 コピー  
 プラン 208, 210  
 プラン・グループ 208  
 コメント  
 監査証跡への追加 20  
 コントローラ、デバイス  
**sp\_helpdevice** および番号 455  
 コンパイル済みオブジェクト  
 ソース・テキストの確認 142  
 ソース・テキストの表示 493  
 ソース・テキストを隠す 507  
 コンパニオン・サーバ  
 設定 190-193

## さ

サーバ  
 アップグレードと **sp\_checknames** 126  
 アップグレードと **sp\_checkreswords** 131  
**sp\_server\_info** 情報 872  
 アクティビティのモニタリング 591  
 オプション、**sp\_serveroption** による変更  
 713-718  
 削除 330-331  
 属性名 872  
 追加 54-57  
 名前 54  
 リモート 489  
 リモート・ログインの情報 477  
 ローカル 54  
 ロー・ロック・プロモーション・スレッショルドの設定 728  
 「プロセス (サーバのタスク)」「リモート・サーバ」参照  
 サーバ・エイリアス 55  
 サーバ・プロセスの ID 番号。  
 「プロセス (サーバのタスク)」参照  
 サーバ情報オプション。「情報 (サーバ)」参照  
 再コンパイル  
 ストアド・プロシージャ 678  
 最初のページ  
 パーティション、**sp\_helppartition** 429  
 ログ・デバイス 474  
 サイズ  
*image* データ型 778  
*text* 記憶領域 778  
 ログ・デバイス 556  
 最適化  
 クエリ (**sp\_recompile**) 678  
 作業領域  
 削除 912  
 削除  
 ロック・プロモーション  
 ・スレッショルド 313  
 エイリアス・ユーザ 295-296  
*dbccdb* 913  
**dbcc checkstorage** 履歴の *dbccdb* 915  
 グループ 315

グループからのユーザの削除 123  
 作業領域 912  
 時間範囲 304  
 抽象プラン・グループ 298  
 データベースからセグメント 328–329  
 データベースからのユーザの削除 335–336  
 データベース・デバイス 305–306  
 ファイル 305  
 プラン 297, 299  
 プロシージャ 310  
 ユーザ定義データ型 334  
 ユーザ定義メッセージ 321  
 リソース制限 300  
 リモート・サーバ 330–331  
 リモート・ログイン 324–325, 330  
 ロー・ロック・プロモーション  
   ・スレッショルド 326  
 「消去」参照  
 削除。「削除」「消去」参照  
 作成  
   *dbccdb* 作業領域 911  
   拡張ストアド・プロシージャ 29–30  
   時間範囲 14  
   リソース制限 9  
   実行クラス 27  
   スレッショルド 58–62  
   制限 9  
   抽象プラン・グループ 8  
   データ型 63–66  
   名前付き時間範囲 14  
   ユーザ・エイリアス 18–19  
   ユーザ定義監査レコード 74  
   ユーザ・グループ 34  
 参照情報  
   *dbcc* ストアド・プロシージャ 905  
   カタログ・ストアド・プロシージャ 857  
   システム拡張ストアド・プロシージャ 885  
   システム・プロシージャ 1–5  
 参照整合性制約  
   名前の変更 690–692  
   ユーザ・メッセージのバインド 101

## し

時間間隔  
   インデックス作成の見積もり 368  
   制限 10  
   前回の **sp\_monitor** 実行以降 595  
 時間範囲  
   at all times 15, 304  
   ID 15  
   アクティブな時間範囲の変更 16  
   削除 304  
   作成 14  
   全日 14  
   重複する 15  
   追加 14  
   変更 579  
 識別子  
   **set quoted\_identifier on** 132, 140–141  
   **sp\_checkreswords** 132  
   引用符付き 133  
   区切り 133  
   名前の変更 134  
   予約語 128–141  
 システム拡張ストアド・プロシージャ 885–903  
   **xp\_cmdshell** 887  
   **xp\_deletemail** 890  
   **xp\_enumgroups** 891  
   **xp\_findnextmsg** 892  
   **xp\_logevent** 893  
   **xp\_readmail** 894  
   **xp\_sendmail** 897  
   **xp\_startmail** 901  
   **xp\_stopmail** 903  
 システム・テーブル  
   ルール 104  
   領域の割り付け 653  
   キャッシュへのバインド 91  
   更新 1  
   直接更新の危険性 137  
   デフォルト 95  
 システム標準の役割  
   **sp\_activeroles** 6  
 システム・プロシージャ  
   名前の変更 135  
   **sp\_reconfig** 512

- sp\_pciconfig 648
- 拡張ストア・プロシージャ 885-903
- カタログ・ストアド 857-884
- 構文の表示 787-788
- 使用 1
- ソース・テキストの表示 493
- パーミッション 2
- ヘルプ・レポート 409-506
- リターン・ステータス 1
- システム・プロシージャ・テーブル 5
  - カタログ・ストアド・プロシージャ 860
- システム・プロシージャの結果。
  - 「情報 (サーバ)」参照
- リソース制限
  - 作成 9
  - 種類 9
- 受信パケット、数 597
- 出力、数
  - パケット 597
- 順序
  - 日付要素 36
  - 「インデックス」「優先度」「ソート順」参照
- ジョイン
  - sp\_commonkey 188
  - 情報 468
- 障害、メディアおよび trunc log on chkpt
  - データベース・オプション 243
- 消去
  - ブランチ 297, 299
  - 「削除」参照
- 衝突、ハッシュ・キー 425
- 情報 (サーバ)
  - suspect (疑わしい) インデックス 511
  - インデックス 462
  - 現在のロック 546
- 情報、レポート
  - suspect (疑わしい) インデックス 511
  - インデックス 462
  - 拡張ストアド・プロシージャ 456
  - カラムのジョイン 468
  - キー 470
  - キャッシュのバインド 92
  - グループ 460, 504
- 言語 472
- 現在のロック 546
- コンパイル済みオブジェクトのソース・テキスト 493
- サーバ・ユーザ 277, 852
- サーバ・プロセス 852
- スレッシュホールド 502
- セグメント 484
- ダンプ・デバイス 454
- データ型 409
- データ・キャッシュ 110
- データベース 448
- データベース・オブジェクト 409
- データベース所有者 504
- データベース・デバイス 453
- 統計、モニタ 591
- トランザクション・ログ・デバイス 474
- パーミッション 478
- パフォーマンス 789
- ユーザ、データベース 504
- リソース制限 420
- リモート・サーバ 489
- リモート・サーバ・ログイン 477
- 領域の使用状況 778
- ログイン 852
- ログの最初のページ 474
- ロック 546, 609
- 省略した名前要素のドット (..) 43
- 使用量の統計 698
- 所有権とダンプ・デバイス 68
- シングルユーザ・モード 243
  - sp\_renamedb 695

## す

- ステータス
  - データベース・デバイス 269
- ストアド・プロシージャ
  - sp\_cluster logical 159
  - sp\_dbcc\_alterwrs 908-909
  - sp\_dbcc\_configreport 910
  - sp\_dbcc\_createws 911-912

**sp\_dbcc\_deletedb** 913-914  
**sp\_dbcc\_deletehistory** 915-916  
**sp\_dbcc\_differentialreport** 917-918  
**sp\_dbcc\_evaluatedb** 919-920  
**sp\_dbcc\_faultreport** 923-928  
**sp\_dbcc\_fullreport** 929  
**sp\_dbcc\_runcheck** 935-936  
**sp\_dbcc\_statisticsreport** 937-940  
**sp\_dbcc\_summaryreport** 941-944  
**sp\_dbcc\_updateconfig** 945-947  
**sp\_recompile** 678  
**dbccdb** データベース 905  
**sp\_checkreswords** 132  
**sp\_procxmode** によるトランザクション  
   ・モードの変更 672-674  
 オブジェクトの依存性 258-265  
 カタログ 857-884  
 キャッシュのバインド 91, 823  
 データベース名の変更 696  
 名前の変更 690-692  
 「データベース・オブジェクト」  
   「システム・プロシージャ」参照  
 スレッシュホールド  
 I/O 削減の最適化 146  
 間隔 60  
 最大数 60, 588  
 削除 332-333  
 情報 502  
 超過 59  
 追加 58-62  
 ヒステリシス値 59, 587  
 変更 586-590  
 無効化 62, 332, 589  
 ラストチャンス 59, 332, 587  
 ロー・ロック・プロモーション 729  
 スレッシュホールド・プロシージャ 59  
 作成 810  
 実行 61, 589  
 渡されるパラメータ 60, 588

## せ

制限された曜日  
   時間範囲の指定 14  
   時間範囲の変更 579  
   リソース制限情報 421  
 制限時間  
   時間範囲の指定 14  
   時間範囲の変更 579  
   リソース制限情報 420  
 制限タイプ 9  
 I/O コスト 9  
 tempdb\_space 9  
   値の指定 10  
   値の変更 576  
   返されるローの数 9  
   経過時間 9  
 精度、データ型  
   **sp\_help** レポート 413  
   ユーザ定義データ型 63  
 制約  
   **sp\_unbindmsg** によるメッセージの  
     バインド解除 831  
   情報 429, 443  
   ソース・テキストの表示 493  
   名前の変更 690-692  
   ユーザ・メッセージのバインド 101  
 セキュリティ  
   ユーザ定義 Web サービス 850  
 セグメント  
   名前の変更 137, 139  
   **sp\_helpthreshold** のレポート 502  
   空き領域のモニタリング 58-62, 586-590  
   拡張 52, 376  
   削除 328-329  
   情報 484  
   追加 51-53  
   **sp\_checkreswords** 132  
   マッピング 52  
   「データベース・デバイス」、「ログ・セグメン  
     ト」、「領域の割り付け」参照  
 設定パラメータ  
   表示レベル 275  
   ヘルプ情報 437  
   変更 199-207

設定パラメータの基本表示レベル 275  
 設定パラメータの包括表示レベル 275

## そ

ソース・テキスト  
 暗号化、リバース 508  
 隠す 507  
 存在の確認 142  
 表示 493  
 ソース・テキストの暗号化解除 508  
 ソース・テキストの暗号化のリバース 508  
 ソート順  
 情報 491  
 変更、および **sp\_indsuspect**  
   システム・プロシージャ 511  
 属性  
**sp\_addobjectdef** 45  
 サーバ (**sp\_server\_info**) 872  
 実行クラス 27

## た

ダンプ設定パラメータ  
 変更 195–198  
 ダンプ・レコードの消去 337  
 ダンプ・デバイス  
 削除 305–306  
 追加 67–69  
 パーミッションと所有権の問題 68  
 リスト作成 454  
 「データベース・デバイス」、  
 「ログ・デバイス」参照  
 ダンプ、データベース  
 割り込み 250  
 端末  
 7 ビット、**sp\_helpsort** の出力例 491  
 8 ビット、**sp\_helpsort** の出力例 491

## ち

チェックポイント・プロセス  
**trunc log on chkpt**  
   データベース・オプション 244  
 チャージバック・アカウンティング  
**sp\_clearstats** プロシージャ 154–155  
**sp\_reportstats** プロシージャ 698–699  
 抽象プラン・グループ  
 インポート 509  
 エクスポート 374  
 削除 298  
 追加 8  
 名前の変更 693  
 抽象プラン・グループのインポート 509  
 抽象プランの変更 719  
 抽象プラン  
 情報 426  
**sp\_help\_qplan** 426  
 重複する時間範囲 15  
 直接更新  
 システム・テーブル 136

## つ

追加  
*sysusermessages* へのメッセージの追加 40–42  
 エイリアス 18–19  
 エンジン・グループ 25  
 外部キー 399–400  
 グループへのエンジンの追加 25  
 グループへのユーザの追加 70–71, 123–124  
 サーバ 54–57  
 時間範囲 14  
 リソース制限 9  
 実行クラス 27  
 スレッショルド 58–62  
 制限 9  
 セグメント 51–53  
 ダンプ・デバイス 67–69  
 抽象プラン・グループ 8  
 データベースへのグループの追加 34–35  
 データベースへのユーザの追加 70–71



- 名前付き時間範囲 14
- 日付文字列 36–38
- ユーザ定義データ型 63–66
- リモート・ログイン 47–50
- 月の値
  - 代替言語 36
- て**
- ディスク・デバイス
  - 追加 67–69
- ディスク・ミラーリング
  - sp\_who** レポート 855
- データ型
  - ODBC 860
  - sp\_datatype\_info** 情報 868
  - sp\_help** の情報 419
  - sp\_unbindrule** によるルールのバインド解除 832–834
  - 階層 65
  - コード 860, 868
  - デフォルト 94–96
  - デフォルトのバインド解除 826–827
  - 物理 63
  - ユーザ定義の削除 334
- データ型の優先度。「優先度」参照
- データ型、カスタム。
  - 「ユーザ定義データ型」参照
- データ・キャッシュ
  - logonly** タイプ 115
  - オーバーヘッド 114, 434
  - オブジェクトのバインド 89
  - オブジェクトのバインド解除 821
  - 削除 115
  - 情報 110, 434
  - ステータス 112
  - すべてのオブジェクトのバインド解除 824
  - 設定 106–116
  - メモリ・プール 658–664
  - リカバリ 110
- データの依存性。「依存性、データベース・オブジェクト」参照
- データベース
  - ユーザの追加 70
  - suspect** データベースのリスト 538
  - suspect** ページのリスト 541
  - 記憶領域使用量の情報 451, 778
  - 記憶領域の情報 778
  - グループの追加 34
  - 所有権 121
  - スレッシュホールド 810
  - セグメントの削除 328–329
  - データ・キャッシュからのバインド解除 821
  - データ・キャッシュへのバインド 89, 91
  - 名前の変更 695–697
  - sp\_helpdb** 448
  - sp\_databases** 867
  - sp\_checknames** 126
  - ヘルプ 448
  - ユーザの削除 335
  - 領域不足 810
  - ロー・ロック・プロモーション
    - ・スレッシュホールドの削除 326
  - ロー・ロック・プロモーション
    - ・スレッシュホールドの設定 728
  - ロック・プロモーション
    - ・スレッシュホールド 723
  - 「データベース・オブジェクト」参照
  - データベース・オブジェクト
    - sp\_tables** リスト 883
    - 依存性 258–265
    - キャッシュへのバインド 89
    - 検索 263, 417
    - 使用領域 778–781
    - デフォルトのバインド 94–96
    - 名前の変更 690–692
    - パーミッション 478
    - リスト 409
    - ルールのバインド 103
  - データベース・オブジェクト所有者と **sp\_depends**
    - システム・プロシージャ 258
  - データベース・オプション 239–246
    - 個々のオプション名も参照
    - 設定の表示 233, 448
    - リスト作成 229–246

- データベースからのセグメントのマッピング解除 328–329
- データベース所有者
  - ユーザの追加 70
  - dbo use only** データベース・オプション 240
  - 譲渡、所有権 121
  - 情報 504–506
  - 変更 121
  - 「データベース・オブジェクト所有者」「パーミッション」参照
- データベース・デバイス
  - defaulton** または **defaultoff** ステータス 269–270
  - dsynch** 設定 266
  - sp\_helpdevice** システム・プロシージャ 453
  - 削除 305–306
  - ステータス 269
  - セグメントの削除 328–329
  - リスト 453
- データベースの設計
  - キーの削除 316
  - 論理関係 188, 399
- データベースのダンプ 340
- データベース・ファイル。「ファイル」参照
- データベース・リカバリ順序
  - sp\_dbrecovery\_order** システム・プロシージャ 247–249
  - システム・データベース 248
- テーブル・ダンプ・デバイス
  - 追加 67–69
- テーブル
  - sp\_recompile** 678
  - sp\_tables** 883
  - 名前の変更 134
  - sp\_column\_privileges** からのカラム・パーミッション情報 862–863
  - sp\_placeobject** 領域の割り付け 653–654
  - sp\_table\_privileges** 情報 882
  - suspect (疑わしい) インデックス 511
  - オブジェクトの依存性 258–265
  - カラム情報 864
  - キーの削除 316
  - 共通キー 188–189
  - システム・プロシージャ 5, 860
  - ジョインされた共通キー 188–189
  - 使用領域 780
- データ・キャッシュからのバインド解除 821
- データ・キャッシュへのバインド 89
- 名前の変更 690–692
- sp\_checkreswords** 131
- プライマリ・キー 667
- 保持されたロック 381, 548
- 領域の見積もり 368
- ロー・ロック・プロモーション
  - ・スレッシュホルドの削除 326
- ロー・ロック・プロモーション
  - ・スレッシュホルドの設定 728
- ロック・プロモーション
  - ・スレッシュホルド 723
- ロック、種類 381, 548
- デバイス
  - 名前の変更 137, 139
  - dsync** 設定 266
  - ログの情報 474
  - 「*sysdevices* テーブル」参照
- デバイス・フラグメント、**sp\_helpdb** レポート 448
- デフォルト
  - システム・テーブル 95
  - ソース・テキストの表示 493
  - 名前の変更 135, 690–692
  - sp\_checkreswords** 131
  - バインド 94–96
  - バインド解除 826–827
- デフォルト設定
  - 設定パラメータ 202
- デフォルト・データベース
  - 「*sysdevices* テーブル」参照
- デフォルト・データベース・デバイス
  - sp\_helpdevice** 454
  - sp\_diskdefault** 269
- デフォルトの言語
  - 追加 36–38
- テンポラリ・テーブル
  - sp\_help** 418
  - システム・プロシージャ 5
- テンポラリ名。「エイリアス、ユーザ」参照

## と

## 統計

**sp\_monitor** 591  
**sp\_reportstats** 698–699  
**sp\_clearstats** プロシージャ 154  
 グローバル変数によって戻される統計 591  
**systabstats** 390

同時実行性の最適化 146

## 独立性レベル

**identity in nonunique index**  
 データベース・オプション 241  
 カタログ・ストアド・プロシージャ 858  
 システム・プロシージャ 1

## トランザクション

モード 672–674

## トランザクション・ログ

**select into/bulkcopy/pilsort**  
 データベース・オプション 242  
**trunc log on chkpt** オプション 244  
 スレッシュホールド 332  
 データ・キャッシュ 662  
 別のデバイス 555–557  
 ログ I/O サイズ 662

## トリガ

**sp\_recompile** 678  
 名前の変更 135  
 オブジェクトの依存性 258–265  
 ソース・テキストの表示 493  
 データベース名の変更 696  
 名前の変更 690–692  
**sp\_checkreswords** 131

## な

## 名前

DLL ファイル 401  
 エイリアス 18, 295, 335  
 異なる名前の割り当て、  
 エイリアスとの比較 70  
 サーバ 54  
 サーバ属性 872  
 識別子の変更 134

データベース・オブジェクトの変更 690–692

**sp\_checknames** 126  
**sp\_checkreswords** 128  
 リモート・ユーザ 324

## 名前付き時間範囲

at all times 15, 304  
 ID 15  
 アクティブな時間範囲の変更 16  
 削除 304  
 作成 14  
 全日 14  
 重複する 15  
 追加 14  
 変更 579

名前付き時間範囲の開始時刻 14

名前付き時間範囲の開始曜日 14

名前付き時間範囲の終了時刻 14

名前の変更 690–692

警告 691, 696

データベース 695–697

「**sp\_rename** システム・プロシージャ」参照

名前付き時間範囲の終了曜日 14

## に

7ビット ASCII 文字、**sp\_checknames** 125

クエリ・プラン、**sp\_recompile** 678

**identity in nonunique index** データベース・オプション、**sp\_dboption** 241

**no chkpt on recovery** データベース・オプション、**sp\_dboption** 242

**no free space acctg** データベース・オプション、**sp\_dboption** 241

## ね

ネットワーク上でのログイン  
 ・パスワードの保護 714

## の

ノンクラスタード・インデックスの  
**max\_rows\_per\_page** の式 149

## は

- パスワードのセキュリティ
  - ネットワーク上でのログイン
    - ・パスワードの保護 714
- パーミッション
  - sp\_column\_privileges** 862
  - 新しいデータベース所有者 121
  - システム・プロシージャ 2
  - 情報 478
  - ダンプ・デバイス 68
  - 取り消し 478
  - 付与 478
  - ユーザ・パーミッションの表示 277
- 排他ロー・ロック 550
- 排他ロック 381, 548
- バインド
  - 制約へのユーザ
    - ・メッセージのバインド 101-102
  - データ・キャッシュ 89-93
  - データ・キャッシュへのオブジェクトのバインド 89-93
  - デフォルト 94-96
  - バインド解除 821-823, 826
  - ルール 103-105
- バインド解除
  - キャッシュからのオブジェクトのバインド解除 821-823
  - データ・キャッシュ 821-823
  - デフォルト 826-827
- パススルー・モード
  - sp\_autoconnect** システム・プロシージャ 82
  - sp\_passthru** システム・プロシージャ 629
  - sp\_remotesql** システム・プロシージャ 687
- パス名、ダンプ・デバイス 67
- パスワード
  - sp\_remotoption** 684
  - sp\_serveroption** 717
  - trusted ログインまたは確認 684
  - 最後の変更の日付 283
  - ネットワーク間での暗号化 717
- パスワードの確認。「パスワード」「**sp\_remotoption** システム・プロシージャ」参照
- 破損したデータベース
  - リカバリ・フォールト・アイソレーション・モード 732
  - リスト作成 538
- 破損したページ
  - オンラインにする 397-398
  - リカバリ時のアイソレート 736-737
  - リスト作成 541
- パターン一致とカタログ・ストアド・プロシージャのパラメータ 860
- ハッシュ・キーの衝突 425
- パフォーマンス
  - 情報 789
  - 同時実行性の最適化 146
- パラメータ、プロシージャ
  - 指定する方法 4, 859
- バルク・コピー。「**bcp** (バルク・コピー・ユーティリティ)」参照
- ハロウィーン問題
  - 問題回避、**unique auto\_identity index** オプション 245
- 範囲ロック 550
- 範囲、リソース制限値の指定 9
- 番号
  - ODBC データ型コード 860
  - spid** (サーバ・プロセス ID) 852
  - グローバル変数の単位 597
  - データ型コード 860
  - デバイス 455
  - メッセージ 40, 72, 321, 403
  - 曜日名 36
  - 「ID、ユーザ」参照

## ひ

- 日付要素、順序 36
- 非同期プリフェッチ
  - 制限の設定 663
- 非同期ログ・サービス (ALS) 448
  - sp\_helpdb** による出力 450
- ビュー
  - オブジェクトの依存性 258-265
  - カラム 864
  - キーの削除 316

共通キー 188-189  
 ソース・テキストの表示 493  
 データベース名の変更 696  
 名前の変更 135, 690-692  
**sp\_checkreswords** 131  
 プライマリ・キー 667  
 表示  
   コンパイル済みオブジェクト  
     のソース・テキスト 493  
   データベース・オプション 229-246  
   文字セット 491  
   モジュールの構文 787  
**ふ**  
 ファイル  
   **interfaces**、サーバ名 55  
   削除 305  
   **sp\_dropdevice** 305  
   ローカライゼーション 139  
   「テーブル」「トランザクション・ログ」参照  
 ファイル名  
   DLL 401  
   設定ファイル 200  
 プール、メモリ  
   設定 658  
   デフォルト 108  
 フォーマット、可変長文字データ 84-88  
 フォーマット、名前付き時間範囲の時刻 14  
 フォールト・アイソレーション  
   インデックス・レベル 394, 539  
 複雑なパスワード  
   **sp\_cleanpwdchecks** 151  
   **sp\_extrapwdchecks** 380  
   **sp\_passwordpolicy** 633  
 物理データ型 63  
 物理デバイス名 67  
 プライマリ・キー  
   **sp\_dropkey** プロシージャ 316  
   **sp\_foreignkey** 399  
   **sp\_helpkey** 470  
   **sp\_primarykey** 定義 667

プラットフォーム間のダンプとロード、疑わしい  
   パーティションの処理 666  
 プラン  
   **sp\_showplan** 出力 760  
   検索 384  
   コピー 208, 210  
   削除 297, 299  
   消去 297  
   比較 183, 186  
   変更 719  
 ブランク  
   カタログ・ストアド・プロシージャ  
     のパラメータ値 859  
   システム・プロシージャのパラメータ値 4  
 プラン・グループ  
   エクスポート 374  
   コピー 208  
   削除 298  
   作成 8  
   情報 424  
   すべてのプランの削除 297  
   追加 8  
   テーブルへのコピー 374  
   比較 183  
   レポート 424  
 プラン・グループのエクスポート 374  
 プラン・グループの比較 183  
 プランの比較 183, 186  
 プリフェッチ  
   無効化 117  
   有効化 117  
 プレースホルダ  
   エラー・メッセージのパーセント(%)記号 41  
 プロシージャ「ストアド・プロシージャ」「シス  
   テム・プロシージャ」参照  
 プロセス(サーバのタスク)  
   ID番号 852  
   **sp\_showplan** 表示 760-761  
   **sp\_who** レポート 852-855  
   保持されているロックのチェック 546  
   ロックの確認 381-383, 546-551  
 プロセスのブロック  
   **sp\_lock** レポート 382, 549  
   **sp\_who** レポート 855

プロセスの論理名。「論理デバイス名」参照  
プロモーション、ロック 722  
分散トランザクション管理 (DTM) 814

## へ

ページ・ロック、種類 381, 548  
ページ、データ  
保持されたロック 381, 548  
数の計算、**sp\_spaceused** 780  
ヘルプ・レポート  
インデックス 462  
拡張ストアド・プロシージャ 456  
キー 470  
グループ 460  
言語、代替 472  
コンパイル済みオブジェクト  
のソース・テキスト 493  
システム・プロシージャ 409–506  
ジョイン 468  
スレッシュホールド 502  
制約 443  
セグメント 484  
ダンプ・デバイス 453  
データ型 409  
データベース 448  
データベース・オブジェクト 409  
データベース・デバイス 453  
テーブル 409  
パーミッション 478  
ユーザ 504–506  
リソース制限 420  
リモート・サーバ 489  
ログイン 477  
「情報 (サーバ)」「システム・プロシージャ」参照  
ヘルプ、**sp\_sysmon** 表示 789  
変換  
ユーザ定義メッセージ 41  
変更  
*dbccdb* 作業領域サイズ 908  
オブジェクト名 690–692  
言語のエイリアス 721

時間範囲 579  
スレッシュホールド 586–590  
設定パラメータ 199  
設定パラメータの表示レベル 275  
ダンプ設定パラメータ 195  
抽象プラン・グループの名前 693  
データ・キャッシュ内のメモリ・プール 658  
データベース・オプション 229–246  
データベース所有者 121–122  
名前付き時間範囲 579  
ユーザのグループ 123–124  
リソース制限 575

## ほ

保護システム  
グループ 34  
ボリューム処理 838

## ま

マシン目盛り (チック) 597  
マッピング  
データベース 250  
リモート・ユーザ 47  
丸一日の時間範囲 14  
マルチバイト文字セット  
**sp\_helpsort** 出力 492  
ソート順 492

## み

未使用領域  
**sp\_spaceused** レポート 780  
ミラーリング。「ディスク・ミラーリング」参照

## め

命名  
グループ 34

時間範囲 14  
 ユーザ定義データ型 65  
 メール・メッセージ、サーバ  
 削除 890  
 処理 669–671  
 セッションの開始 901  
 セッションの停止 903  
 送信 897  
 読み込み 894  
 メッセージ  
**sp\_getmessage** プロシージャ 403–404  
**sp\_unbindmsg** によるバインド解除 831  
*sysusermessages* テーブル 40–42  
 言語設定 321, 403  
 システム・プロシージャ 5  
**sp\_droplanguage** 318  
 番号 40, 72, 321, 403  
 ユーザ定義の削除 321  
 ユーザ定義の追加 40–42  
 ログイン 72–73  
 メッセージの取得。「**sp\_getmessage** システム・  
 プロシージャ」参照  
 メモリ  
 解放、XP Server から 401  
 設定パラメータが使用 437  
 マッピング 250  
 メモリ・プール  
**sp\_logiosize** 562, 568  
 ウォッシュ・パーセンテージの設定 663  
 最小サイズ 661  
 設定 658  
 デフォルト 108  
 トランザクション・ログ 662  
 非同期プリフェッチ制限の設定 663

## も

文字セット  
 名前の変更 137, 139  
**sp\_helpsort** での表示 491  
**sp\_checknames** 125  
**sp\_checkreswords** 132  
 マルチバイト 492

文字セットのバイナリ・ソート順 492  
 モジュール、構文の表示 787  
 モニタリング  
 空き領域 58, 59, 587  
 システム・アクティビティ 591  
 ロック競合 609

## や

役割  
**sp\_activeroles** 6  
 役割の階層、表示  
**sp\_activeroles** 6  
**sp\_displayroles** 284

## ゆ

ユーザ  
 名前の変更 139  
**sp\_who** レポート 852–855  
*sysusers* テーブル 18  
 アカウント統計 154, 698  
 エイリアスの削除 295–296  
 グループの変更 123–124  
 情報 277, 504  
 追加 70–71  
 データベースからの削除 335–336  
 パーミッション 478  
 リモート 477  
 リモートの削除 330  
 ユーザ ID  
**sp\_import\_qpgroup** 509  
 表示 283  
 ユーザ ID。「エイリアス」「ログイン」「ユーザ」  
 参照  
 ユーザが作成したオブジェクト。  
 「データベース・オブジェクト」参照  
 ユーザ定義 Web サービス  
 セキュリティ 850  
 ユーザ定義エラー・メッセージの  
 フォーマット文字列 41  
 ユーザ定義監査レコード 74

## ユーザ定義データ型

名前の変更 135

**sp\_unbindrule** によるルー

ルのバインド解除 832-834

階層 65

削除 334

作成 63-66

デフォルトのバインド 94-96

デフォルトのバインド解除 826-827

**sp\_checkreswords** 131

命名 65

ルールのバインド 103

## ユーザ定義の役割

**sp\_activeroles** 6

## ユーザ定義プロシージャ

**sp\_addextendedproc** 29

## ユーザ定義メッセージ 40-42

**sp\_unbindmsg** によるバインド解除 831

## ユーザ定義メッセージの置き換え 40

## ユーザ・データグラム・プロ

トコルのメッセージ機能 710

## ユーザのリンク。「エイリアス、ユーザ」参照

## ユーザ名

**sp\_checkreswords** 132

変更 137

「データベース・オブジェクト所有者」

「ログイン」参照

## ユーザ・パーミッション。「データベース所有者」

「パーミッション」参照

## 優先度

**sp\_setpsex** 726

カラムとデータ型へのデフォルトのバインド 95

リソース制限 12

ルールのバインド 104

## ユーティリティ・コマンド

構文の表示 787-788

『ASE ユーティリティ・ガイド』参照

## ㇿ

## 曜日

時間範囲 14

代替言語 36

## 曜日の日付値

最初の曜日 36

名前と番号 36

## 予約語

カタログ・ストアド・プロシージャ 859

識別子として使用 128-141

システム・プロシージャ 4

## ら

ラストチャンス・スレッショルド 59, 587

## り

## リカバリ

suspect データベースのリスト 538

オフライン・ページのリスト 541

スレッショルドの設定 736

データ・キャッシュ 110

**sp\_forceonline\_db** 392**sp\_forceonline\_page** 397

モードの設定 732

モードの表示 732

## リカバリ・フォールト・ア

イソレーション 394, 539

## リスト

**dbcc** ストアド・プロシージャ 905

カタログ・ストアド・プロシージャ 857

## リスト作成

データベース・オプション 229

デバイス 454

## リソース制限

削除 300

情報 420

変更 575

変更例 577

## リソース制限の実施 10

## リソース制限のスコープ

アクティブな時間範囲への変更 16

指定 11

情報 421



リターン・ステータス  
**sp\_checkreswords** 131  
 カタログ・ストアド・プロシージャ 858  
 システム・プロシージャ 1

リモート・サーバ  
 名前の変更 137, 139  
**sp\_remotoption** 684-686  
 情報 489  
 名前 54  
 ログイン情報 477  
 ログインの削除 324  
 「サーバ」参照

リモート・ログイン  
**sp\_remotoption** 684-686  
*sysremotelogins* テーブル 47-50  
 trusted モードと untrusted モード 684  
 削除 324-325  
 情報 477

「ログイン」「ユーザ」参照  
 リモート・ユーザ。「リモート・ログイン」参照  
 領域

**sp\_modifythreshold** による空き領域のモニタ  
 リング 586-590

**sp\_spaceused** プロシージャ 778-781  
 テーブルとインデックス  
 ・サイズの見積もり 368-373

未使用 780  
 「サイズ」、「領域の割り付け」参照

領域の割り付け  
**sp\_placeobject** プロシージャ 653-654  
 今後の割り付け 653-654  
 ログ・デバイス 556  
 「データベース・デバイス」  
 「セグメント」参照

## る

ルール  
 名前の変更 135  
 システム・テーブル 104  
 ソース・テキストの表示 493  
 名前の変更 690-692  
**sp\_checkreswords** 131

バインド 103-105  
 バインド解除 832-834  
 ユーザが作成したルールの命名 103

## れ

レコード、監査 20  
 レポート  
**sp\_who** 852-855  
 プラン・グループ 424

## ろ

ローカライゼーション、言語名とファイルの変更  
 139

ローカル・エイリアス、言語 721  
 ローカル・サーバ 54  
 「リモート・サーバ」「サーバ」参照

ロー・ロック 550  
 ロー・ロック・プロモーション・スレッシュホールド  
 データベースの設定についての **sp\_helpdb** による  
 レポート 448

**sp\_droprowlockpromote** 326  
**sp\_setrowlockpromote** 728

ロー、テーブル  
 返される数の制限 10  
 数の計算、**sp\_spaceused** 780

ロギング  
 メッセージ 72-73  
 ユーザ定義メッセージ 40

ログイン  
 ロックの解除 552-554  
*sysremotelogins* テーブル 47-50, 324, 330  
 アカウント統計 154, 698  
 エイリアス 18, 295  
 数 598

現在のデータベース所有者の変更 121  
 削除 330  
 情報 277, 477  
 リソース制限情報 420  
 リソース制限値の適用 9  
 リソース制限の削除 300

リソース制限の変更 575  
 リモート 324-325, 330  
 リモートのオプション 684  
 ロック 552-554  
 “probe” 699  
 「リモート・ログイン」「ユーザ」参照  
 ロック・プロモーション・スレッシュホールド  
**sp\_help** レポート 418  
 データベースの設定についての **sp\_helpdb**  
 によるレポート 448  
**sp\_drowlockpromote** 326  
**sp\_setrowlockpromote** 728  
 ログイン・アカウントのロック解除 552  
 ログ・セグメント  
**sp\_helplog** レポート 474  
**sp\_helpthreshold** のレポート 502  
 ログ・デバイス情報 474  
 ロック  
**sp\_familylock** システム・プロシージャ 381-383  
**sp\_lock** システム・プロシージャ 546-551  
 意図的テーブル 381, 548  
 キャッシュのバインド 91  
 キャッシュのバインド解除 823  
 競合のモニタリング 609  
 共有テーブル 381, 548  
 共有ページ 381, 548  
 種類 381, 548  
 情報の表示 546, 609  
 制御 546-551  
**sp\_lock** 546  
 排他テーブル 381  
 排他テーブルとページ 548  
 排他ページ 381  
 page 381, 548  
 ロー 550  
 ログイン 552  
 “FAM DUR” ステータス 383  
 ロック・スキーム、**sp\_help** レポート 418  
 論理デバイス名 67, 269  
*syslogs* テーブル 555

## わ

ワーカー・プロセスのファミリー  
**sp\_lock** によってレポートされる *fid* 549  
*fid* の **sp\_familylock** 381  
 ワイルドカード文字  
 SQL 規格のパターン一致 (\$ と \_) 860  
  
 一重引用符。「引用符」参照  
 階層、役割。「役割の階層」参照  
 権限。「パーミッション」参照  
 今後の領域の割り付け。  
 「領域の割り付け」参照 653  
 参照、オブジェクト。「依存性、データベース・オブジェクト」参照  
 所有者。「データベース・オブジェクト所有者」  
 「データベース所有者」参照  
 照合順。「ソート順」参照  
 代替の ID。「エイリアス、ユーザ」参照  
 代替言語。「言語、代替」参照  
 低位の役割と高位の役割。「役割の階層」参照