

学校的理想装备

电子图书·学校专集

校园网上的最佳资源

SQL Server 7

参考手册 (四)





## 返回总目录

第 22 章系统表 .....	2
22.1 系统表 .....	2
22.2 信息模式视图 .....	76

# 第 22 章 系统表

## 22.1 系统表

Microsoft SQL Server 使用的信息存放在某种特殊的表中，称为系统表。所有的数据库都包含系统表，这些表包含 Microsoft SQL Server 和每个用户数据库的信息。每个用户数据库都有自己的系统表集，主数据库包含许多系统表，主数据库所包含的系统表用于服务器。

下列系统表位于主系统数据库中，而所有其他系统表都位于分布式系统数据库(或其他与复制有关的数据库)的 msdb 系统数据库中，或位于用户数据库中：

- sysobjects
- sysindexes
- syscolumns
- systypes
- syscomments
- sysfiles1
- syspermissions
- syslockinfo
- syslanguages
- syscharsets
- syscursorrefs
  - syscursors
- syscursorcolumns
  - syscursortables

- sysusers
- sysdepends
- sysreferences
- sysfulltextcatalogs
- sysindexkeys
- sysforeignkeys
- sysmembers
- sysprotects
- sysdatabases
- sysperfinfo
- sysprocesses
- sysxlogins
- syslocks
- sysdevices
- sysmessages
- sysconfigures
- syscurconfigs
- sysservers
- syscacheobjects
  - sysaltfiles
  - sysfiles
    - sysfilegroups
  - sysallocations
- SYSREMOTE\_PROVIDER\_TYPES
- SYSREMOTE\_TABLE\_PRIVILEGES
- SYSREMOTE\_COLUMN\_PRIVILEGES
- SYSREMOTE\_PRIMARY\_KEYS
- SYSREMOTE\_FOREIGN\_KEYS
- SYSREMOTE\_CATALOGS
- SYSREMOTE\_SCHEMATA
- SYSREMOTE\_TABLES
- SYSREMOTE\_VIEWS
- SYSREMOTE\_COLUMNS
- SYSREMOTE\_INDEXES
- SYSREMOTE\_STATISTICS

使用 Allow Updates 选项将允许直接对系统表进行更新。在缺省情况下，该选项是关闭的，故用户不能通过 ad hoc 更新来更新系统表。如果允许更新选项没有打开(1)，用户就只能用系统存储过程来更新系统表。当允许更新选项是关闭的(0)，则不允许进行更新。当允许更新选项打开时，那么拥有许可的任何用

户都可以用 ad hoc 更新直接对系统表进行更新，并可以创建更新系统表的存储过程。

更新系统表可能阻止 SQL Server 启动，也可能导致出现不一致；所以 Microsoft 通常建议不使用更新系统表。另外，如果在创建存储过程的同时，允许更新选项处于打开状态，那么，甚至在关闭允许更新选项后，这些存储过程也可以更新系统表。用下述语法在命令提示行上启动 SQL Server 来直接更新系统表时，可以阻止他人使用 SQL Server：

```
sqlservr_m
```

上述命令将以单用户模式启动 SQL Server，并打开 Allow Updates 选项。

Allow Updates 选项为动态选项，该选项打开或关闭时不需重启 SQL Server，而只需使用 RECONFIGURE WITH OVERRIDE 语句。

要允许直接对系统表进行更新，可使用如下语法：

```
sp_configure Allow Updates , 1
```

```
GO
```

```
RECONFIGURE WITH OVERRIDE
```

```
GO
```

若不允许直接对系统表进行更新，可使用如下语法：

```
sp_configure Allow Updates , 0
```

```
GO
```

```
RECONFIGURE WITH OVERRIDE
```

```
GO
```

在 SQL Server 7.0 版本中，对系统表进行了重大改动，增添了许多新的系统

表，其它系统表也进行了更改；还添加了一些视图。

以下几节将逐一介绍每个系统表，之后，将介绍 Information Schema View。

### 22.1.1 backupfile

backupfile 系统表位于 msdb 系统数据库中，它包含数据和记录文件的备份信息。其中的项目有：

- backup\_set\_id
- first\_family\_number
- first\_media\_number
- filegroup\_name
- page\_size
- file\_number
- backed\_up\_page\_count
- file\_type
- source\_file\_block\_size
- file\_size
- logical\_name
- physical\_drive
- physical\_name.

## 22.1.2 backupmediafamily

backupmediafamily 系统表位于 msdb 系统数据库中，它包含介质家族的备份信息。其中的项目有：

- media\_set\_id
- family\_sequence\_number
- media\_family\_id
- media\_count
- logical\_device\_name
- physical\_device\_name
- device\_type
- physical\_block\_size

## 22.1.3 backupmediaset

backupmediaset 系统表位于 msdb 系统数据库中，它包含备份介质集的备份信息。其中的项目有：

- media\_set\_id
- media\_uuid
- media\_family\_count
- name

- description
- software\_name
- software\_vendor\_id
- MTF\_major\_version

backupset 系统存储过程位于 msdb 系统数据库中，它包含备份集的备份信息。其中的项目有：

- |                         |                          |
|-------------------------|--------------------------|
| · backup_set_id         | · software_major_version |
| · backup_set_uuid       | · software_minor_version |
| · media_set_id          | · software_build_version |
| · first_family_number   | · time_zone              |
| · first_media_number    | · mtf_minor_version      |
| · last_family_number    | · first_lsn              |
| · last_media_number     | · last_lsn               |
| · catalog_family_number | · checkpoint_lsn         |
| · catalog_media_number  | · database_backup_lsn    |
| · position              | · database_creation_date |
| · expiration_date       | · backup_start_date      |
| · software_vendor_id    | · backup_finish_data     |
| · name                  | · type                   |
| · description           | · sort_order             |
| · user_name             | · code_page              |
| · compatibility_level   | · database_name          |



- database\_version
- backup\_size
- server\_name
- machine\_name

### 22.1.5 MSagent\_parameters

MSagent\_parameters 系统表位于 msdb 系统数据库中，它包含代理配置文件的参数信息。其中的项目有：

- profile\_id
- parameter\_name
- value

### 22.1.6 MSagent\_profiles

MSagent\_profiles 系统表位于 msdb 系统数据库中，它包含代理配置文件的的信息，其中的项目有：

- profile\_id
- profile\_name
- agent\_type
- type
- description
- def\_profile

## 22.1.7MSarticles

MSarticles 系统表位于 distribution(分布式)系统数据库中，它包含复制选项的信息。其中的项目有：

- publisher\_id
- publisher\_db
- publication\_id
- article
- article\_id
- destination\_object
- source\_owner
- source\_object
- description

## 22.1.8MSdistpublishers

MSdistpublishers 系统表位于 msdb 系统数据库中，它包含由本地 Distributor (分销商)支持的每个远程出版商的信息。其中的项目有：

- name
- distribution\_db
- working\_directory

- security\_mode
- login
- password
- active
- trusted
- thirdparty\_flag

### 22.1.9 MSdistributiondbs

MSdistributiondb 系统表位于 msdb 系统数据库中，它包含与本地 Distributor 相关的分布式数据库的信息。其中的项目有：

- name
- min\_distretention
- max\_distretention
- history\_retention

### 22.1.10 MSdistribution\_agents

MSdistribution\_agents 系统表位于分布式系统数据库中，它包含在 Distributor 上运行的 Distribution Agent(s)信息。其中的项目有：

- id

- publication
- name
- subscriber\_id
- publisher\_database\_id
- subscriber\_db
- publisher\_id
- subscription\_type
- publisher\_db
- local\_job
- job\_id
- anonymous\_subid
- subscription\_guid
- subscriber\_name
- profile\_id
- virtual\_agent\_id
- anonymous\_agent\_id

### 21.1.11 MSdistribution\_history

MSdistribution\_history 系统表位于分布式系统数据库中，它包含在 Distributor 上运行的 Distributor Agents 的历史信息。其中的项目有：

- agent\_id

- delivered\_transactions
- runstatus
- delivered\_commands
- start\_time
- average\_commands
- time
- delivery\_rate
- duration
- delivery\_latency
- comments
- total\_delivered\_commands
- xact\_seqno
- error\_id
- current\_delivery\_rate
- updateable\_row
- current\_delivery\_latency
- timestamp

## 22.1.12MSdistributor

MSdistributor 系统表位于 msdb 系统数据库中，它具有 Distributor 的属性，其中的项目有 property（属性）和 value（值）。

### 22.1.13MSlogreader\_agents

MSlogreader\_agents 系统表位于分布式系统数据库之中，它包含在 Distributor 上运行的 Log Reader Agent(s)的信息。其中的项目有：

- id
- name
- publisher\_id
- publisher\_db
- publication
- local\_job
- job\_id
- profile\_id

### 22.1.14MSlogreader\_history

MSlogreader\_history 系统表位于分布式系统数据库中，它包含与 Distributor 相关的 Log Reader Agents 的历史信息。其中的项目有：

- agent\_id
- delivered\_commands
- runstatus
- average\_commands

- start\_time,time
- delivery\_rate
- duration,comments
- delivery\_latency
- xact\_seqno
- error\_id
- delivery\_time
- timestamp
- delivered\_transactions

### 22.1.15MSmerge\_contents

MSmerge\_contents 系统表位于发布的数据库之中，所包含的信息是：在为合并复制中使用的基于优先等级的缺省冲突解析器而发布数据库的过程中，数据库中修改过哪些行。其中的项目有：

- tablenick
- rowguid
- generation
- partchangeegen
- joinchangeegen
- lineage
- colv1

## 22.1.16MSmerge\_delete\_conflicts

MSmerge\_delete\_conflicts 系统表位于发布的数据库中(但是, 如果用户决定分散冲突记录, 该系统表就可能位于预订数据库中)。它包含被删除行的信息, 这些被删除行由于冲突而被卷回。其中的项目有:

- tablenick
- rowguid
- origin\_datasource
- conflict\_type
- reason\_code
- reason\_text
- pubid

## 22.1.17MSmerge\_genhistory

MSmerge\_genhistory 系统表位于分布式系统数据库中, 它包含订阅者了解的某一阶段或某几阶段的信息, 以防止共用阶段的交换, 并在恢复这些阶段后, 使订阅者再次保持同步。其中的项目有:

- guidsrc
- guidlocal
- pubid



- generation
- art\_nick
- nicknames
- coldate

## 22.1.18MSmerge\_history

MSmerge\_history 系统表位于分布式系统数据库中，它包含为订阅者更新的历史信息，其中的项目有：

- agent\_id
- delivery\_rate
- runstatus
- publisher\_insertcount
- start\_time
- publisher\_updatecount
- time
- publisher\_deletecount
- duration
- publisher\_conflictcount
- comments
- subscriber\_insertcount
- delivery\_time

- subscriber\_updatecount
- subscriber\_deletcount
- subscriber\_conflictcount
- error\_id · timestamp

## 22.1.19MSmerge\_replinfo

MSmerge\_replinfo 系统表位于发布的数据库中，包含由合并作业从服务器中创建的复制阶段信息。其中的项目有：

- repid
- replnickname
- recgen
- recguid
- sentgen
- sentguid
- schemaversion
- schemaguid
- merge\_jobid
- snapshot\_jobid

## 22.1.20MSmerge\_subscriptions

MSmerge\_subscriptions 系统表位于分布式数据库中，其中的阶段信息包含由 Subscriber 上的 Merge Agent 提供的每个预约服务对应的一行。该系统表包含的项目有：

- publisher\_id
- publisher\_db
- publication\_id
- subscriber\_id
- subscriber\_db
- subscription\_type
- sync\_type
- status
- subscription\_time

## 22.1.21MSmerge\_tombstone

MSmerge\_tombstone 系统表位于发布的数据库中，它包含被删除行的信息而对其它订阅者而言，是重复执行删除操作，该系统表包含的项目有：

- rowguid
- tablenick

- type
- lineage
- generation
- reason

### 22.1.22MSpublication\_access

MSpublication\_access 系统表位于分布式数据库中，它包含带有发布许可和 Publisher 的 SQL Server 登录的信息。其中的项目有：

- publisher\_id
- publication\_id
- login

### 22.1.23MSpublications

MSpublications 系统表位于分布式数据库中，它包含每个由 Publisher 复制的出版物的信息。其中的项目有：

- publisher\_id
- immediate\_sync
- publisher\_db
- allow\_push

- publication
- allow\_pull
- publication\_id
- allow\_anonymous
- publication\_type
- description
- thirdparty\_flag
- vendor\_name
- independent\_agent
- default\_access

## 22.1.24MSpublisher\_databases

MSpublisher\_databases 系统表位于分布式数据库中，它标识了 Distributor 上 Publisher 的数据库。其中的项目有：

- publisher\_id
- publisher\_db
- id(the id of the row)

## 22.1.25MSreplication\_objects

MSreplication\_objects 系统表位于订阅者数据库中，它标识了 Subscriber 数据库中 与副本相关的对象。其中的项目有：

- publisher
- publisher\_db
- publication
- object\_name
- object\_type(u=table,t=trigger)

## 22.1.26MSreplication\_subscriptions

MSreplication\_subscription 系统表位于订阅者数据库中，它包含了 Distribution Agent 复制到订阅者数据库中的复制信息，其中的项目有：

- publisher
- publisher\_db
- publication
- independent\_agent
- subscription\_type
- distribution\_agent(0=push,1=pull,2=anonymous)
- time

- description
- transaction\_timestamp
- update\_mode
- agent\_id
- subscription\_guid
- subid
- reinit

## 22.1.27MSrepl\_commands

MSrepl\_commands 系统表位于分布式数据库中，它包含被复制命令的复制信息，其中的项目有：

- publisher\_database\_id
- xact\_seqno
- type
- article\_id
- originator\_id
- command\_id
- partial\_command
- command

## 22.1.28MSrepl\_errors

MSrepl\_errors 系统表位于分布式数据库中，它包含扩展复制代理失败的复制信息，其中的项目有：

- id
- time
- error\_type\_id
- source\_type\_id
- source\_name
- error\_code
- error\_text

## 22.1.29MSrepl\_originators

MSrepl\_originators 系统表位于分布式数据库中，它包含已开始处理事务的可更新 Subscribers 的复制信息。其中的项目有：

- id
- publisher\_database\_id
- srvname
- dbname



### 22.1.30MSrepl\_transactions

MSrepl\_transactions 系统表位于分布式数据库中，它包含被复制事务的复制信息。其中的项目有：

- publisher\_database\_id
- xact\_id
- xact\_seqno
- entry\_time

### 22.1.31MSrepl\_version

MSrepl\_version 系统表位于分布式数据库中，它包含复制版本的复制信息。其中的项目有：

- major\_version
- minor\_version
- revision
- db\_existed

### 22.1.32MSsnapshot\_agents

MSsnapshot\_agents 系统表位于分布式数据库中，它包含与 Distributor 相关的 Snapshot Agents 的复制信息，其中的项目有：

- id
- name
- publisher\_id
- publisher\_db
- publication
- publication\_type
- local\_job
- job\_id
- profile\_id

### 22.1.33MSsnapshot\_history

MSsnapshot\_history 系统表位于分布式数据库中，它包含与 Distributor 相关的 Snapshot Agents 的历史复制信息，其中的项目有：

- agent\_id
- delivered\_transactions
- runstatus

- delivered\_commands
- start\_time
- delivery\_rate
- time
- error\_id
- duration
- timestamp
- comments

### 22.1.34MSsubscriber\_info

MSsubscriber\_info 系统表位于分布式数据库中,它包含 Publisher 和 Subscriber 以及他们从 Distributor 中被压入预约的复制信息,其中的项目有:

- publisher
- subscriber
- type
- login
- password
- description
- security\_mode

### 22.1.35 MSsubscriber\_schedule

MSsubscriber\_schedule 系统表位于分布式数据库中，它包含 Publisher 和 Subscriber 的缺省合并和事务同步时间表的复制信息。其中的项目有：

- publisher
- frequency\_subday
- subscriber
- frequency\_subday\_interval
- agent\_type
- active\_start\_time\_of\_day
- frequency\_type
- active\_end\_time\_of\_day
- frequency\_interval
- active\_start\_date
- frequency\_relative\_interval
- active\_end\_date
- frequency\_recurrence\_factor

### 22.1.36 MSsubscriptions

MSsubscriptions 系统表位于分布式数据库中，包含与 Distributor 相关的预约

复制信息，其中的项目有：

- publisher\_database\_id
- status
- publisher\_id
- subscription\_seqno
- publisher\_db
- snapshot\_seqno\_flag
- publication\_id
- independent\_agent
- article\_id
- subscription\_time
- subscriber\_id
- loopback\_detection
- subscriber\_db
- agent\_id
- subscription\_type
- update\_mode
- sync\_type
- publisher\_seqno

### 22.1.37MSsubscription\_properties

MSsubscription\_properties 系统表位于分布式数据库中，它包含出栈 Distribution Agents 的参数复制信息，其中的项目有：

- publisher
- publisher\_db
- publication
- publication\_type(0=snapshot,1=transactional)
- publisher\_login
- publisher\_password
- publisher\_security\_mode(0=SQL Server Authentication,1=Windows NTAuthentication)
- encrypted\_publisher\_password
- distributor\_login
- distributor\_password
- distributor\_security\_mode
- encrypted\_distributor\_password

### 22.1.38restorefile

restorefile 系统表位于 msdb 数据库中，它包含恢复文件的信息。其中的项

目有：

- restore\_history\_id
- file\_number
- destination\_phys\_driv
- destination\_phys\_name

### 22.1.39 restorefilegroup

restorefilegroup 系统表位于 msdb 数据库中，它包含恢复文件组的信息，其中的项目有：restore\_history\_id 和 filegroup\_name。

### 22.1.40 restorehistory

restorehistory 系统表位于 msdb 数据库中，它包含恢复对象的信息。其中的项目有：

- restore\_history\_id
- replace
- restore\_date
- recovery
- destination\_database\_name
- restart

- user\_name
- stop\_at
- backup\_set\_id
- device\_count
- restore\_type

## 22.1.41 sysalerts

sysalerts 系统表位于 msdb 数据库中，包含由服务器上的事件引发的警告信息(如 E-mail, pager 等)，其中的项目有：

- id
- enabled
- name
- delay\_between\_responses
- event\_source
- last\_occurrence\_date
- event\_category\_id
- last\_occurrence\_time
- event\_id
- last\_response\_date
- message\_id
- last\_response\_time



- severity
- notification\_message
- include\_event\_description
- job\_id
- database\_name
- has\_notification
- event\_description\_keyword
- flags
- occurrence\_count
- performance\_condition
- count\_reset\_date
- category\_id
- count\_reset\_time

## 22.1.42 sysallocations

sysallocations 系统表格位于每一个数据库中，它包含分配单位的信息，其中的项目有：

- objid
- indid
- fileid
- gaminterval

- iampage

### 22.1.43 sysaltfiles

sysaltfiles 系统表位于主数据库中，它包含数据库中的文件信息，其中的项目有：

- fileid
- status
- groupid
- perf
- size
- dbid
- maxsize
- name
- growth
- filename

### 22.1.44 sysarticles

sysarticles 系统表位于出版商数据库中，它包含数据库中各文章的复制信息。其中的项目有：

- artid
- description
- columns
- dest\_table
- creation\_script
- filter
- del\_cmd
- filter\_clause
- ins\_cmd
- sync\_objid
- name
- type
- objid
- upd\_cmd
- pubid
- schema\_option
- pre\_creation\_cmd
- dest\_owner
- status

## 22.1.45 sysarticleupdates

sysarticleupdates 系统表位于出版商数据库中，它包含可更新预约的文章复制信息。其中的项目有：

- artid
- pubid
- sync\_ins\_proc
- sync\_upd\_proc
- sync\_del\_proc
- autogen

## 22.1.46 syscategories

syscategories 系统表位于 msdb 数据库中，它包含用于为工作、警告和操作人员分类的类别信息。其中的项目有：

- category\_id
- category\_class
- category\_type
- name

## 22.1.47 syscharsets

syscharsets 系统表位于主数据库中，它包含在使用 Microsoft SQL Server 时可用的字符集和排序信息（实际上只使用一个），其中的项目有：

- type
- id
- csid
- status,name
- description
- binarydefinition
- definition

## 22.1.48 syscolumns

syscolmns 系统表位于每个数据库中，它包含所有表和视图中的列，以及存储过程中每个自变量的信息。其中的项目有：

- name · domain
- id
- number
- xtype
- colorder

- `typestat`
- `autoval`
- `xusertype`
- `offset`
- `length`
- `status`
- `xprec`
- `type`
- `xscale`
- `usertype`
- `colid`
- `printfmt`
- `xoffset`
- `prec`
- `bitpos`
- `scale`
- `reserved`
- `iscomputed`
- `colstat`
- `isoutparam`
- `cdefault`
- `isnullable`

## 22.1.49 syscomments

`syscomments` 系统表位于每个数据库中，它包含 CHECK 约束、DEFAULT 约束、缺省值、规则、存储过程、触发程序以及视图的信息，其中的项目有：

- `id`
- `texttype`
- `number`
- `language`
- `colid`
- `encrypted`
- `status`
- `compressed`
- `ctext`
- `text`

此版本中，不要从 `syscomments` 中删除信息，而需使用带 `ENCRYPTION` 参数的 `CREATE PROCEDURE` 语句。

## 22.1.50 sysconfigures

`sysconfigures` 系统表位于主数据库中，它包含配置选项的信息。其中的项目有：

- value
- config
- comment
- status

### 22.1.51 sysconstraints

sysconstraints 系统表位于每个数据库中，它包含约束及约束拥有者信息。其中的项目有：

- constid
- id
- colid
- spare1
- status
- actions
- error

### 22.1.52 syscurconfigs

syscurconfigs 系统表位于主数据库中，它包含当前配置选项的信息。其中的项目有：



- value
- config
- comment
- status

## 22.1.53 sysdatabases

sysdatabases 系统表位于主数据库中，它包含数据库的信息。其中的项目有：

- name
- sid
- dbid
- mode
- status
- category
- status2
- cmptlevel
- crdate
- filename
- reserved
- suid
- version

## 22.1.54 sysdepends

`sysdepends` 系统表格位于主数据库中，它包含不同对象之间的从属信息。其中的项目有：

- `id`
- `depdbid`
- `depid`
- `depsiteid`
- `number`
- `selall`
- `depnumber`
- `resultobj`
- `status`
- `readobj`

## 22.1.55 sysdevices

`sysdevices` 系统表位于主系统数据库中，它包含数据库文件、磁盘备份文件、磁带备份文件的信息。该系统表只具有向后的兼容性。参看 `sysfiles` 一节。其中的项目有：

- `name`

- size
- low
- high
- status
- cntrltype
- phyname

## 22.1.56 sysdownloadlist

sysdownloadlist 系统表位于 msdb 系统数据库中，它包含目标服务器的下载指令信息。其中的项目有：

- instance\_id
- object\_type
- source\_server
- object\_id
- operation\_code
- target\_server
- error\_message
- status
- date\_posted
- deleted\_object\_name
- date\_downloaded

## 22.1.57 sysfiles

`sysfiles` 虚拟系统表位于主系统数据库中，它包含数据库文件的信息。其中的项目有：

- `fileid`
- `groupid`
- `size`
- `maxsize`
- `growth`
- `status`
- `perf`
- `name`
- `filename`

## 22.1.58 sysfilegroups

`sysfilegroups` 系统表位于主系统数据库中，它包含数据文件组的信息。其中的项目有：

- `groupid`
- `allocpolicy`
- `status`

- `groupname`

## 22.1.59 `sysforeignkeys`

`sysforeignkeys` 系统表位于每个数据库中，它包含 FOREIGN KEY 约束的信息。其中的项目有：

- `constid`
- `fkeyid`
- `rkeyid`
- `fkey`
- `rkey`
- `keyno`

## 22.1.60 `sysfulltextcatalogs`

`sysfulltextcatalogs` 系统表位于每个数据库中，它包含全文本目录的信息，其中的项目有：

- `ftcatid`
- `name`
- `status`
- `path`

## 22.1.61 sysindexes

sysindexes 系统表位于每个数据库中，它包含索引和表的信息。其中的项目有：

- id
- xmaxlen
- status
- maxirow
- first
- OrigFillFactor
- indid
- reserved1
- root
- reserved2
- minlen
- FirstIAM
- keycnt
- impid
- groupid
- lockflags
- dpages

- pgmodctr
- reserved
- keys
- used
- name
- rowcnt
- statblob
- rowmodctr
- maxlen
- soid
- rows
- csid

## 22.1.62 sysindexkeys

sysindexkeys 系统表位于每个数据库中，它包含索引中的列信息。其中的项目有：

- id
- indid
- colid
- keyno

## 22.1.63 sysjobhistory

sysjobhistory 系统表位于 msdb 系统数据库中，它包含已安排的工作信息，其中的项目有：

- instance\_id
- run\_date
- job\_id
- run\_time
- step\_id
- run\_duration
- step\_name
- operator\_id\_emailed
- sql\_message\_id
- operator\_id\_netsent
- sql\_severity
- operator\_id\_paged
- message
- retries\_attempted
- run\_status
- server



## 22.1.64 sysjobschedules

sysjobschedules 系统表位于 msdb 系统数据库中，它包含工作时间的信息，其中的项目有：

- schedule\_id
- freq\_recurrence\_factor
- job\_id
- active\_start\_date
- name
- active\_end\_date
- enabled
- active\_start\_time
- freq\_type
- active\_end\_time
- freq\_interval
- next\_run\_date
- freq\_subday\_type
- next\_run\_time
- freq\_subday\_interval
- date\_created
- freq\_relative\_interval

## 22.1.65 sysjobs

sysjobs 系统表位于 msdb 系统数据库中，它包含已安排的工作信息。其中的项目有：

- job\_id
- description
- originating\_server
- start\_step\_id
- name
- category\_id
- enabled
- owner\_sid
- notify\_level\_eventlog
- notify\_page\_operator\_id
- notify\_level\_email
- delete\_level
- notify\_level\_netsend
- date\_created
- notify\_level\_page
- date\_modified
- notify\_email\_operator\_id

- version\_number
- notify\_netsend\_operator\_id

## 22.1.66 sysjobserver

sysjobserver 系统表位于 msdb 系统数据库中，它包含工作及其目标服务器的信息。其中的项目有：

- job\_id
- server\_id
- last\_run\_outcome
- last\_outcome\_message
- last\_run\_date
- last\_run\_time
- last\_run\_duration

## 22.1.67 sysjobsteps

sysjobsteps 系统表位于 msdb 系统数据库中，它包含工作步骤的信息。其中的项目有：

- job\_id
- server

- step\_id
- database\_name
- step\_name
- database\_user\_name
- subsystem
- retry\_attempts
- command
- retry\_interval
- flags
- os\_run\_priority
- additional\_parameters
- output\_file\_name
- cmdexec\_success\_code
- last\_run\_outcome
- on\_success\_action
- last\_run\_duration
- on\_success\_step\_id
- last\_run\_retries
- on\_fail\_action
- last\_run\_date
- on\_fail\_step\_id
- last\_run\_time

## 22.1.68syslanguages

syslanguages 系统表位于主系统数据库中，它包含 Microsoft SQL Server 可用的每种语言的信息。其中的项目有：

- langid
- months
- dateformat
- shortmonths
- datefirst
- days
- upgrade
- lcid
- name
- msglangid
- alias

## 22.1.69syslockinfo

syslockinfo 系统表位于主数据库中，它包含锁定需求的信息。其中的项目有：

- rsc\_text
- req\_mode

- rsc\_bin
- req\_status
- rsc\_valblk
- req\_refcnt
- rsc\_dbid
- req\_cryrefcnt
- rsc\_indid
- req\_lifetime
- rsc\_objid
- req\_spid
- rsc\_type
- req\_ecid
- rsc\_flag
- req\_ownertype

## 22.1.70 syslogins

syslogin 系统表位于主数据库中，它包含注册帐号的信息。其中的项目有：

- suid
- totio
- sid
- spacelimit

- status
- timelimit
- createdate
- resultlimit
- updatedate
- name
- accdate
- dbname
- totcpu
- password
- language
- securityadmin
- denylogin
- serveradmin
- hasaccess
- setupadmin
- isntname
- processadmin
- isntgroup
- diskadmin
- isntuser
- dbcreator

- sysadmin
- loginname

### 22.1.71 sysmembers

sysmembers 系统表位于每个数据库中，它包含每个数据库角色的成员信息。其中的项目有：memberuid 和 groupuid。

### 22.1.72 sysmergearticles

sysmergearticles 系统表位于出版商数据库中，它包含合并文章的信息，其中的项目有：

- name
- article\_resolver
- type
- ins\_conflict\_proc
- objid
- insert\_proc
- sync\_objid
- update\_proc
- view\_type



- select\_proc
- artid
- schema\_option
- description
- destination\_object
- pre\_creation\_command
- resolver\_clsid
- pubid
- subset\_filterclause
- nickname
- missing\_col\_count
- column\_tracking
- missing\_cols
- status
- columns
- conflict\_table
- resolver\_info
- creation\_script
- view\_sel\_proc
- conflict\_script

## 22.1.73 sysmergepublications

sysmergepublications 系统表位于出版商数据库中，它包含合并出版物的信息，其中的项目有：

- publisher
- allow\_push
- publisher\_db
- allow\_pull
- name
- allow\_anonymous
- description
- centralized\_conflicts
- retention
- status
- publication\_type
- snapshot\_ready
- pubid
- enabled\_for\_internet
- designmasterid
- default\_access
- parentid

- dynamic\_filters
- sync\_mode

## 22.1.74 sysmergeschemachange

sysmergeschemachange 系统表位于出版商数据库中，它包含模式更改的信息。其中的项目有：

- pubid
- artid
- schemaversion
- schemaguid
- schematype
- schematext

## 22.1.75 sysmergesubscriptions

sysmergesubscriptions 系统表位于出版商数据库中，它包含 Subscribers 的信息。其中的项目有：

- subid
- datasource\_path
- partnerid

- srvid
- datasource\_type
- db\_name
- pubid
- priority
- status
- sync\_type
- subscriber\_type
- description
- subscription\_type
- login\_name
- last\_validated

## 22.1.76sysmergesubsetfilters

sysmergesubsetfilters 系统表位于出版商数据库中，它包含被分割的文章的  
Contains 连接过滤器信息的信息。其中的项目有：

- filtername
- join\_articlename
- join\_filterid
- join\_nickname
- pubid

- join\_unique\_key
- artid
- expand\_proc
- art\_nickname
- join\_filterclause

### 22.1.77 sysmessages

sysmessages 系统表位于主数据库中，它包含系统错误和警告的信息。其中的项目有：

- error
- severity
- dlevel
- description
- msglangid

### 22.1.78 sysnotifications

sysnotifications 系统表位于 msdb 数据库中，包含有关通知的信息，其中的项目有：

- alert\_id

- operator\_id
- notification\_method

## 22.1.79 sysobjects

sysobjects 系统表位于每个数据库中，包含数据库中的对象信息。其中的项目有：

- name
- type
- id
- userstat
- xtype
- sysstat
- uid
- indexdel
- info
- refdate
- status
- version
- base\_schema\_ver
- deltrig
- replinfo

- instrig
- parent\_obj
- updtrig
- crdate
- seltrig
- ftcatid
- category
- schema\_ver
- cache
- stats\_schema\_ver

## 22.1.80 sysoledbusers

sysoledbusers 系统表位于主数据库中，它包含有关为连接服务器映射的用户和口令信息。其中的项目有：

- rmtsrvid
- rmtloginame
- rmtpassword
- loginsid
- status
- changedate

## 22.1.81 sysoperators

sysoperators 系统表位于 msdb 数据库中，它包含操作员的信息。其中的项目有：

- id
- enabled
- name
- email\_address
- last\_email\_date
- saturday\_pager\_end\_time
- last\_email\_time
- sunday\_pager\_start\_time
- pager\_address
- sunday\_pager\_end\_time
- last\_pager\_date
- pager\_days
- last\_pager\_time
- netsend\_address
- weekday\_pager\_start\_time
- last\_netsend\_date
- weekday\_pager\_end\_time



- last\_netsend\_time
- saturday\_pager\_start\_time
- category\_id

## 22.1.82 sysperfinfo

sysperfinfo 系统表位于主数据库中，它包含 Windows NT Performance Monitor 中的 SQL Server 性能计数器的信息。其中的项目有：

- object\_name
- counter\_name
- instance\_name
- cntr\_value
- cntr\_type

## 22.1.83 syspermissions

syspermissions 系统表位于每个数据库中，它包含数据库中的许可信息。其中的项目有：

- id
- selmod
- grantee

- updadd
- grantor
- updmod
- actadd
- refadd
- actmod
- refmod
- seladd

## 22.1.84 sysprocesses

sysprocesses 系统表位于主数据库中，它包含在 Microsoft SQL Server 上运行的过程信息。其中的项目有：

- spid
- status
- kpid
- sid
- blocked
- hostname
- waittype
- program\_name
- dbid

- hostprocess
- uid
- cmd
- cpu
- nt\_domain
- physical\_io
- nt\_username
- memusage
- net\_address
- login\_time
- net\_library
- last\_batch
- loginame
- ecid
- suid
- open\_tran

## 22.1.85 sysprotects

sysprotects 系统表位于每个数据库中，它包含安全帐户的许可信息。其中的项目有：

- id

- uid
- action
- protecttype
- columns
- grantor

## 22.1.86 syspublications

syspublications 系统表位于出版商数据库中，它包含数据库中的出版物信息。其中的项目有：

- description
- status
- name
- sync\_method
- pubid
- snapshot\_jobid
- repl\_freq
- independent\_agent
- immediate\_sync
- immediate\_sync\_ready
- enabled\_for\_internet
- allow\_sync\_tran

- allow\_push
- autogen\_sync\_procs
- allow\_pull
- retention
- allow\_anonymous
- default\_access

## 22.1.87 sysreferences

sysreferences 系统表位于每个数据库中，它包含引用列的 FOREIGN KEY 约束信息。其中的项目有：

- constid · fkey13
- fkeyid · fkey14
- rkeyid · fkey15
- rkeyindid · fkey16
- keycnt · rkey1
- forkeys · rkey2
- refkeys · rkey3
- fkeydbid · rkey4
- rkeydbid · rkey5
- fkey1 · rkey6
- fkey2 · rkey7

- fkey3 · rkey8
- fkey4 · rkey9
- fkey5 · rkey10
- fkey6 · rkey11
- fkey7 · rkey12
- fkey8 · rkey13
- fkey9 · rkey14
- fkey10 · rkey15
- fkey11 · rkey16
- fkey12

## 22.1.88 sysremotelogins

sysremotelogins 系统表位于主数据库中，它包含远程用户的信息。其中的项目有：

- remoteserverid
- remoteusername
- suid
- status
- sid
- changedate

## 22.1.89 sysreplicationalerts

sysreplicationalerts 系统表位于 msdo 系统数据库中，它包含复制警报的原因信息，其中的项目有：

- alert\_id
- publication
- status
- publication\_type
- agent\_type
- subscriber
- agent\_id
- subscriber\_db
- error\_id
- article
- alert\_error\_code
- destination\_object
- time
- source\_object
- publisher
- alert\_error\_text
- publisher\_db

## 22.1.90sys.servers

sys.servers 系统表位于主数据库中，它包含远程服务器和本地服务器的信息。其中的项目有：

- srvid
- providername
- srvstatus
- datasource
- srvname
- location
- srvproduct
- providerstring
- schemadate
- catalog
- topologyx
- srvnetname
- topologyy

## 22.1.91sys.subscriptions

sys.subscriptions 系统表位于出版商数据库中，它包含订阅信息，其中的项目



有：

- artid
- subscription\_type
- srvid
- distribution\_jobid
- dest\_db
- timestamp
- status
- update\_mode
- sync\_type
- loopback\_detection
- login\_name

#### 22.1.92 systargetservergroupmembers

systargetservergroupmembers 系统表位于 msdb 数据库中，它包含当前在 Multi Server 组中支持的目标服务器信息。其中的项目有 servergroup\_id 和 server\_id。

#### 22.1.93 systargetservergroups

systargetservergroup 系统表位于 msdb 数据库中，它包含在 Multi Server 情景中支持哪些目标服务器组的信息，其中的项目有 servergroup\_id 和 name。

## 22.1.94 systargetservers

systargetservers 系统表位于 msdb 数据库中，它包含当前在 Multi Server 情景中支持的目标服务器信息，其中的项目有：

- server\_id
- last\_poll\_date
- server\_name
- status
- location
- local\_time\_at\_last\_poll
- time\_zone\_adjustment
- enlisted\_by\_nt\_user
- enlist\_date
- poll\_interval

## 22.1.95 systaskids

systaskids 系统表位于 msdb 数据库中，它包含以前版本中创建的任务信息，以及 7.0 版本中的工作信息。其中的项目有：task\_id 和 job\_id。

## 22.1.96 systypes

systypes 系统表位于每个数据库中，它包含数据类型的信息，其中的项目有：

- name
- reserved
- xtype
- usertype
- status
- variable
- xusertype
- allownulls
- length
- type
- xprec
- printfmt
- xscale
- prec
- tdefault
- scale
- domain,uid

## 22.1.97 sysusers

sysusers 系统表位于每个数据库中，它包含数据库用户的信息。其中的项目有：

- uid
- environ
- status
- hasdbaccess
- name
- islogin
- sid
- isntname
- roles
- isntgroup
- createdate
- isntuser
- updatedate
- issqluser
- altuid
- isaliased
- password

- issqlrole
- suid
- isapprole
- gid

下一节中,将讨论系统表中用来查看 SQL Server 元数据的 Information Schema Views(信息模式视图)。

## 22.2 信息模式视图

Microsoft SQL Server 可通过存储过程或信息模式视图返回系统表的信息。要从这些视图中检索信息,可提供合法的名称 INFORMATION\_SCHEMA.view\_name。

举例如下:

```
select * from INFORMATION_SCHEMA.CHECK_CONSTRAINTS
```

在本例中,会返回 sysobjects 和 syscomments 系统表中的信息。

### 22.2.1 CHECK\_CONSTRAINTS

Information Schema View CHECK\_CONSTRAINTS 包含 sysobjects 和 syscomments 系统表中检查约束的信息。其中的项目有:

- CONSTRAINT\_CATALOG
- CONSTRAINT\_SCHEMA
- CONSTRAINT\_NAME
- CHECK\_CLAUSE

## 22.2.2 COLUMN\_DOMAIN\_USAGE

Information Schema View COLUMN\_DOMAIN\_USAGE 包含的信息是由当前数据库中的用户所拥有的用户自定义数据类型列，它们位于 sysobjects，syscolumns 和 systypes 系统表中。其中的项目有：

- DOMAIN\_CATALOG
- DOMAIN\_SCHEMA
- DOMAIN\_NAME
- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- COLUMN\_NAME

## 22.2.3 COLUMN\_PRIVILEGES

Information Schema view COLUMN\_PRIVILEGES 包含列的特权信息，这些

特权可以授予给当前使用的数据库中的用户，也可以由其授予给他人。它们位于 sysprotects、sysobjects，以及 syscolumns 系统表中，其中的项目有：

- GRANTOR
- GRANTEE
- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- COLUMN\_NAME
- PRIVILEGE\_TYPE
- IS\_GRANTABLE

## 22.2.4 COLUMNS

Information Schema View COLUMNS 包含当前数据库中用户的列的信息，它们位于 sysobjects，spt\_data type\_info，systypes，syscolumns，syscomments，sysconfigures 以及 syscharsets 系统表中。其中的项目有：

- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- COLUMN\_NAME
- ORDINAL\_POSITION
- COLUMN\_DEFAULT

- IS\_NULLABLE
- DATA\_TYPE
- CHARACTER\_MAXIMUM\_LENGTH
- CHARACTER\_OCTET\_LENGTH
- NUMERIC\_PRECISION
- NUMERIC\_PRECISION\_RADIX
- NUMERIC\_SCALE
- DATETIME\_PRECISION
- CHARACTER\_SET\_CATALOG
- CHARACTER\_SET\_SCHEMA
- CHARACTER\_SET\_NAME
- COLLATION\_CATALOG
- COLLATION\_SCHEMA
- COLLATION\_NAME
- DOMAIN\_CATALOG
- DOMAIN\_SCHEMA
- DOMAIN\_NAME

## 22.2.5 CONSTRAINT\_COLUMN\_USAGE

Information Schema View CONSTRAINT\_COLUMN\_USAGE 包含当前数据库中用户的受约束列的信息。它们位于 sysobjects , syscolumns , 以及 systypes



系统表中，其中的项目有：

- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- COLUMN\_NAME
- CONSTRAINT\_CATALOG
- CONSTRAINT\_SCHEMA
- CONSTRAINT\_NAME

## 22.2.6 CONSTRAINT\_TABLE\_USAGE

Information Schema View CONSTRAINT\_TABLE\_USAGE 包含当前数据库中的用户可访问的每个受约束的表信息，它们位于 sysobjects 系统表中。其中的项目有：

- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- CONSTRAINT\_CATALOG
- CONSTRAINT\_SCHEMA
- CONSTRAINT\_NAME

## 22.2.7 DOMAIN\_CONSTRAINTS

Information Schema View DOMAIN\_CONSTRAINTS 包含当前数据库中的数据类型和用户规则信息，它们位于 sysobjects 和 systypes 系统表中，其中的项目有：

- CONSTRAINT\_CATALOG
- CONSTRAINT\_SCHEMA
- CONSTRAINT\_NAME
- DOMAIN\_CATALOG
- DOMAIN\_SCHEMA
- DOMAIN\_NAME
- IS\_DEFERRABLE
- INITIALLY\_DEFERRED

## 22.2.8 DOMAINS

Information Schema View DOMAINS 包含当前数据库中用户的数据类型信息，它们位于 spt\_data type\_info, systypes, syscomments, sysconfigures 以及 syscharsets 系统表中。其中的项目有：

- DOMAIN\_CATALOG
- DOMAIN\_SCHEMA

- DOMAIN\_NAME
- DATA\_TYPE
- CHARACTER\_MAXIMUM\_LENGTH
- CHARACTER\_OCTET\_LENGTH
- COLLATION\_CATALOG
- COLLATION\_SCHEMA
- COLLATION\_NAME
- CHARACTER\_SET\_CATALOG
- CHARACTER\_SET\_SCHEMA
- CHARACTER\_SET\_NAME
- NUMERIC\_PRECISION
- NUMERIC\_PRECISION\_RADIX
- NUMERIC\_SCALE
- DATETIME\_PRECISION
- DOMAIN\_DEFAULT

## 22.2.9 KEY\_COLUMN\_USAGE

Information Schema View KEY\_COLUMN\_USAGE 包含由当前数据库中的用户拥有的列信息，它们位于 sysobjects，syscolumns，sysreferences，spt\_values 及 sysindexes 系统表中。其中的项目有：

- CONSTRAINT\_CATALOG

- CONSTRAINT\_SCHEMA
- CONSTRAINT\_NAME
- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- COLUMN\_NAME
- ORDINAL\_POSITION

## 22.2.10(2) REFERENTIAL\_CONSTRAINTS

Information Schema View REFERENTIAL\_CONSTRAINTS 包含当前数据库中的用户拥有的外部约束信息。它们位于 sysreferences, sysindexes 及 sysobjects 系统表中，其中的项目有：

- CONSTRAINT\_CATALOG
- CONSTRAINT\_SCHEMA
- CONSTRAINT\_NAME
- UNIQUE\_CONSTRAINT\_CATALOG
- UNIQUE\_CONSTRAINT\_SCHEMA
- UNIQUE\_CONSTRAINT\_NAME
- MATCH\_OPTION
- UPDATE\_RULE
- DELETE\_RULE

## 22.2.11 SCHEMATA

Information Schema View SCHEMATA 包含带有用户许可的每个数据库的信息。它们位于 sysdatabases , sysconfigures 及 syscharsets 系统表中，其中的项目有：

- CATALOG\_NAME
- SCHEMA\_NAME
- SCHEMA\_OWNERD
- DEFAULT\_CHARACTER\_SET\_CATALOG
- DEFAULT\_CHARACTER\_SET\_SCHEMA
- DEFAULT\_CHARACTER\_SET\_NAME

## 22.2.12 TABLE\_CONSTRAINTS

Information Schema View TABLE\_CONSTRAINTS 包含由当前数据库中的用户所拥有的表格约束信息。它们位于 sysobjects 系统表中。其中的项目有：

- CONSTRAINT\_CATALOG
- CONSTRAINT\_SCHEMA
- CONSTRAINT\_NAME
- TABLE\_CATALOG
- TABLE\_SCHEMA

- TABLE\_NAME
- CONSTRAINT\_TYPE
- IS\_DEFERRABLE
- INITIALLY\_DEFERRED

### 22.2.13 TABLE\_PRIVILEGES

Information Schema View TABLE\_PRIVILEGES 包含每个表的特权信息，这种特权可授予当前数据库中的用户，也可由该用户授予给他人。它们位于 sysprotects 和 sysobjects 系统表中，其中的项目有：

- GRANTOR
- GRANTEE
- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- PRIVILEGE\_TYPE
- IS\_GRANTABLE

### 22.2.14 TABLES

Information Schema View TABLES 包含带有当前数据库中用户许可的每个表

格的信息。它们位于 sysobjects 系统表中，其中的项目有：

- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- TABLE\_TYPE

## 22.2.15 VIEW\_COLUMN\_USAGE

Information Schema View VIEW\_COLUMN\_USAGE 包含当前数据库中的用户所拥有的每列的信息，它们位于 sysobjects 和 sysdepends 系统表中。其中的项目有：

- VIEW\_CATALOG
- VIEW\_SCHEMA
- VIEW\_NAME
- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- COLUMN\_NAME

## 22.2.16 VIEW\_TABLE\_USAGE

Information Schema View VIEW\_TABLE\_USAGE 包含由视图所使用并由当前数据库中的用户所拥有的每个表格的信息，它们位于 sysobjects 和 sysdepends 系统表中，其中的项目有：

- VIEW\_CATALOG
- VIEW\_SCHEMA
- VIEW\_NAME
- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME

## 22.2.17 VIEWS

Information Schema View VIEWS 包含当前数据库中的用户可访问的视图信息，它们位于 sysobjects 和 syscomments 系统表中，其中的项目有：

- TABLE\_CATALOG
- TABLE\_SCHEMA
- TABLE\_NAME
- VIEW\_DEFINITION
- CHECK\_OPTION



## · IS\_UPDATABLE

新增添的 Information Schema View 为浏览元数据提供了简便的途径，而在此之前，对元数据的浏览需要通过连接系统表来实现。下一章中，我们将讨论系统存储过程。存储过程是无需直接访问系统表，就能操作和浏览系统表的另一种方法。



## 返回总目录

<b>第 23 章 系统存储过程 .....</b>	<b>18</b>
sp_add_agent_parameter .....	19
sp_add_agent_profile .....	20
sp_add_alert .....	21
sp_addalias .....	24
sp_addapprole .....	25
sp_addarticle .....	25
sp_adddistpublisher .....	31
sp_adddistributiondb .....	33
sp_adddistributor .....	37
sp_addextendedproc .....	38
sp_addgroup .....	38
sp_add_category .....	39
sp_add_job .....	40
sp_add_jobschedule .....	42
sp_add_jobserver .....	45

sp_add_jobstep .....	46
sp_addlinkedserver .....	50
sp_addlinkedsrvlogin .....	55
sp_addlogin .....	57
sp_addmergearticle .....	58
sp_addmergefilter .....	62
sp_addmergepublication .....	63
sp_addmergepullsubscription .....	66
sp_addmergepullsubscription_agent.....	67
sp_addmergesubscription.....	74
sp_addmessage.....	78
sp_add_notification.....	80
sp_add_operator.....	81
sp_addpublication .....	83
sp_addpublication_snapshot .....	88
sp_addpublisher70 .....	90
sp_addpullsubscription .....	91
sp_addpullsubscription_agent.....	93
sp_addremotelogin .....	99
sp_addrole .....	100
sp_addrolemember.....	100
sp_addserver .....	101

sp_addsrvrolemember.....	102
sp_addsubscriber.....	102
sp_addsubscriber_schedule.....	107
sp_addsubscription.....	110
sp_addsynctriggers.....	115
sp_addtask.....	117
sp_addtype.....	117
sp_add_targetservergroup.....	119
sp_addumpdevice.....	119
sp_add_targetsvrgrp_member.....	120
sp_adduser.....	121
sp_altermessage.....	121
sp_apply_job_to_targets.....	122
sp_approlepassword.....	124
sp_article_validation.....	124
sp_articlecolumn.....	126
sp_articlefilter.....	127
sp_articlesynctranprocs.....	128
sp_articleview.....	129
sp_attach_db.....	130
sp_attach_single_file_db.....	130
sp_autostats.....	131

sp_bindefault.....	132
sp_bindrule.....	133
sp_bindsession .....	134
sp_browsereplcmds.....	134
sp_catalogs.....	136
sp_certify_removable.....	136
sp_change_agent_parameter.....	137
sp_change_agent_profile .....	137
sp_changearticle.....	138
sp_changedbowner.....	142
sp_changedistpublisher.....	142
sp_changedistributiondb .....	143
sp_changedistributor_property .....	144
sp_changedistributor_password.....	145
sp_changegroup .....	146
sp_changemergearticle.....	146
sp_changemergefilter.....	149
sp_changemergepublication.....	150
sp_changemergepullsubscription.....	153
sp_changemergesubscription .....	154
sp_changeobjectowner.....	156
sp_changepublication.....	156

sp_changesubscriber .....	159
sp_changesubscriber_schedule .....	164
sp_changesubstatus .....	167
sp_change_subscription_properties .....	171
sp_change_users_login .....	173
sp_column_privileges .....	174
sp_column_privileges_ex .....	175
sp_columns .....	176
sp_columns_ex.....	176
sp_configure.....	177
sp_createstats .....	178
sp_create_removable .....	179
sp_cursor_list.....	181
sp_cycle_errorlog.....	181
sp_databases.....	182
sp_datatype_info .....	182
sp_dbcmptlevel .....	183
sp_dbfixedrolepermission.....	183
sp_dbopiton.....	184
sp_dbremove .....	184
sp_defaultdb.....	185
sp_defaultlanguage .....	186

sp_delete_alert .....	187
sp_delete_category .....	187
sp_delete_job .....	188
sp_delete_jobschedule .....	189
sp_delete_jobserver .....	189
sp_delete_jobstep .....	190
sp_delete_notification .....	191
sp_delete_operator .....	191
sp_delete_targetserver .....	192
sp_delete_targetsevergroup .....	192
sp_delete_targetsvrgrp_member .....	193
sp_deletemergeconflictrow .....	193
sp_denylogin .....	195
sp_depends .....	195
sp_describe_cursor .....	195
sp_describe_cursor_columns .....	196
sp_describe_cursor_tables .....	197
sp_detach_db .....	198
sp_distcounters .....	199
sp_drop_agent_parameter .....	199
sp_drop_agent_profile .....	200
sp_dropalias .....	200

sp_dropapprole .....	201
sp_droparticle .....	201
sp_dropdevice .....	202
sp_dropdistpublisher .....	203
sp_dropdistributiondb .....	203
sp_dropdistributor .....	204
sp_dropextendedproc .....	204
sp_dropgroup .....	205
sp_droplinkedsrvlogin .....	205
sp_droplogin .....	206
sp_dropmergearticle .....	206
sp_dropmergefilter .....	207
sp_dropmergepublication .....	207
sp_dropmergepullsubscription .....	208
sp_dropmergesubscription .....	209
sp_dropmessage .....	210
sp_droppublication .....	211
sp_droppullsubscription .....	211
sp_dropremotelogin .....	212
sp_droprole .....	213
sp_droprolemember .....	214
sp_dropserver .....	214



sp_dropsrvrolemember .....	215
sp_dropsubscriber .....	216
sp_dropsubscription .....	217
sp_droptask .....	218
sp_droptype.....	219
sp_dropuser.....	219
sp_dropwebtask .....	219
sp_dsninfo .....	220
sp_enumcodepages .....	221
sp_enumcustomresolvers .....	222
sp_enumdsn .....	222
sp_enumfullsubscribers .....	222
sp_executesql.....	223
sp_fkeys .....	224
sp_foreignkeys .....	225
sp_fulltext_catalog.....	226
sp_fulltext_column .....	228
sp_fulltext_database.....	229
sp_fulltext_service .....	229
sp_fulltext_table.....	230
sp_generatefilters .....	232
sp_getbindtoken .....	232

sp_getmergedeletetype.....	233
sp_get_distributor .....	234
sp_grant_publication_access .....	234
sp_grantdbaccess .....	235
sp_grantlogin .....	236
sp_help .....	236
sp_help_agent_default .....	237
sp_help_agent_parameter .....	237
sp_help_agent_profile .....	238
sp_help_alert.....	238
sp_helparticle .....	239
sp_helparticlecolumns .....	240
sp_help_category .....	241
sp_helpconstraint .....	242
sp_helpdb .....	242
sp_helpdbfixedrole .....	243
sp_helpdevice.....	243
sp_helpdistpublisher .....	243
sp_helpdistributiondb.....	244
sp_helpdistributor .....	244
sp_help_downloadlist .....	245
sp_helpextendedproc .....	246

sp_helpfile .....	247
sp_helpfilegroup .....	247
sp_help_fulltext_catalogs .....	247
sp_help_fulltext_catalogs_cursor .....	248
sp_help_fulltext_columns .....	249
sp_help_fulltext_tables .....	249
sp_help_fulltext_tables_cursor .....	250
sp_help_fulltext_columns_cursor .....	251
sp_helpgroup .....	252
sp_helphistory .....	252
sp_help_job .....	253
sp_help_jobhistory .....	255
sp_help_jobschedule .....	257
sp_help_jobserver .....	258
sp_help_jobstep .....	259
sp_helpindex .....	260
sp_helplanguage .....	260
sp_helplinkedserver .....	260
sp_helplogins .....	261
sp_helpmergearticle .....	262
sp_helpmergearticleconflicts .....	262
sp_helpmergeconflictrows .....	263

sp_helpmergedeleteconflictrows .....	263
sp_helpmergefilter .....	264
sp_helpmergepublication .....	265
sp_helpmergepullsubscription .....	266
sp_helpmergesubscription .....	267
sp_help_notification.....	268
sp_helpntgroup.....	269
sp_help_operator.....	270
sp_helppublication .....	270
sp_help_publication_access.....	271
sp_helppublication_snapshot.....	272
sp_helppullsubscription .....	272
sp_helpremotelogin .....	273
sp_helpreplicationdb .....	274
sp_helpreplicationdboption.....	274
sp_helpreplicationoption.....	275
sp_helprole .....	275
sp_helprolemember.....	276
sp_helpprotect.....	276
sp_helpserver .....	277
sp_helpsort.....	278
sp_helpsrvrole .....	279

sp_helpsrvrolemember.....	279
sp_helpsubscriber.....	280
sp_helpsubscriberinfo .....	280
sp_helpsubscription .....	280
sp_helpsubscription_properties .....	282
sp_help_targetserver .....	283
sp_help_targetservergroup .....	283
sp_helptask.....	284
sp_helptext.....	284
sp_helptrigger .....	284
sp_helpuser .....	285
sp_indexes.....	285
sp_indexoption.....	286
sp_link_publication.....	287
sp_linkedservers.....	289
sp_makewebtask .....	289
sp_manage_jobs_by_login.....	297
sp_mergedummyupdate .....	298
sp_mergesubscription_cleanup .....	298
sp_montior .....	299
sp_msx_defect .....	299
sp_msx_enlist.....	300

sp_OACreate .....	301
sp_OADestroy .....	302
sp_OAGetErrorInfo .....	303
sp_OAGetProperty .....	304
sp_OAMethod .....	305
sp_OASetProperty .....	306
sp_OAStop .....	306
sp_password .....	307
sp_pkeys .....	307
sp_primarykeys .....	308
sp_post_msx_operation .....	309
sp_processmail .....	310
sp_procoption .....	311
sp_publication_validation .....	311
sp_purgehistory .....	313
sp_purge_jobhistory .....	313
sp_reassigntask .....	314
sp_refreshsubscriptions .....	314
sp_refreshview .....	315
sp_reinitmergepullsubscription .....	315
sp_reinitmergesubscription .....	316
sp_reinitpullsubscription .....	316

sp_reinitsubscription.....	317
sp_remoteoption.....	318
sp_removedbreplication.....	319
sp_remove_job_from_targets .....	319
sp_rename.....	320
sp_renamedb .....	321
sp_replcmds .....	322
sp_replcounters .....	322
sp_repldone.....	323
sp_replflush.....	324
sp_replicationdboption.....	324
sp_replicationoption.....	325
sp_replication_agent_checkup.....	327
sp_replshowcmds.....	327
sp_repltrans.....	328
sp_resync_targetserver.....	328
sp_revoke_publication_access.....	328
sp_revokedbaccess.....	329
sp_revokelogin .....	329
sp_runwebtask .....	330
sp_script_synctran_commands .....	330
sp_scriptdelproc.....	331

sp_scriptinsproc .....	331
sp_scriptupdproc .....	332
sp_server_info .....	332
sp_serveroption .....	333
sp_setapprole .....	334
sp_setnetname .....	335
sp_spaceused .....	336
sp_special_columns .....	336
sp_sproc_columns .....	337
sp_srvrolepermission .....	338
sp_start_job .....	339
sp_statistics .....	340
sp_stop_job .....	341
sp_stored_procedures .....	342
sp_subscription_cleanup .....	342
sp_table_privileges .....	343
sp_table_privileges_ex .....	344
sp_table_validation .....	344
sp_tableoption .....	346
sp_tables .....	347
sp_tables_ex .....	347
sp_unbinddefault .....	348



sp_unbindrule .....	349
sp_update_alert .....	349
sp_updatecategory .....	353
sp_update_job .....	354
sp_update_jobschedule .....	357
sp_update_jobstep .....	361
sp_update_notification .....	365
sp_update_operator .....	366
sp_updatestats .....	369
sp_updatetask .....	370
sp_validname .....	370
sp_validatelogins .....	371
sp_who .....	371
xp_cmdshell .....	371
xp_deletemail .....	372
xp_enumgroups .....	372
xp_findnextmsg .....	373
xp_grantlogin .....	374
xp_logevent .....	374
xp_loginconfig .....	375
xp_logininfo .....	375
xp_msver .....	376

xp_readmail .....	377
xp_revokelogin .....	380
xp_sendmail .....	380
xp_snmp_getstate .....	383
xp_snmp_raisetrap .....	384
xp_sprintf .....	385
xp_sqlinventory .....	386
xp_sqlmaint.....	387
xp_sscanf .....	387
xp_startmail .....	388
xp_stopmail.....	389

<b>附录 A 安装前要准备的检验表 .....</b>	<b>390</b>
检验表条目 .....	390

## 第 23 章 系统存储过程

SQL Server 系统存储过程能使用户不必直接进入系统表，即可执行所有的数据库管理任务。使用 Enterprise Manager，展开服务器和主数据库，然后双击存储过程项，即可得到系统存储过程列表。在缺省的情况下，主数据库并不在 Enterprise Manager 中显示。要打开这个功能以查看系统数据库，用户必须在 Registered SQL Server Properties 窗口下选择 Show System Database 和 System object。选择一个系统存储过程，并双击它，即可查看该系统存储过程的 Transact-SQL 编码。其它系统存储过程位于 msdb 系统数据库之下，可用同样的方法查看和访问它们。区别仅在于需展开 msdb 数据库，而非主数据库。

系统存储过程可以按其功能划分为以下几大类：

Catalog：数据词典功能程序。

Cursors：游标功能程序。

Distributed Queries：分布式查询功能程序。

SQL Server Agent：任务管理功能程序。

Replication：复制功能程序。

Security：安全管理功能程序。

System：SQL Server 维护功能程序。

Web Assistant：Web 功能程序。

Extended Procedures : 与外部程序的 SQL Server 接口。

OLE Automation : Transact-SQL 内部的 OLE 自动化。

Data Transformation Services : 数据仓库程序。

扩展存储过程可以进一步划分为 General , SQL Mail 以及 SQL Server Profiler。

下面 , 将按字母顺序依次讨论 Microsoft SQL Server 系统存储过程。

## sp\_add\_agent\_parameter

sp\_add\_agent\_parameter 系统存储过程用来在快照、合并和事务复制中将参数和该参数值添加到复制代理配置文件中。

语法

```
sp_add_agent_parameter [ @profile_id = ]          profile_id ,  
  [ @ parameter_name = ]  parameter_name ,  
  [ @ parameter_value = ] parameter_value
```

## 变量

[ @profile_id = ] profile_id	profile_id 是一个整数，表示 MSagent_configuration 系统表中的配置。
[ @parameter_name = ] parameter_name	parameter_name 是参数的系统名。
[ @parameter_value = ] parameter_value	parameter_value 是参数的 nvarchar(255)型数值。

## sp\_add\_agent\_profile

sp\_add\_agent\_profile 系统存储过程可为 MSagent\_profile 系统表添加一个复制代理配置文件，以用于快照、合并和事务复制中。

### 语法

```
sp_add_agent_profile [ [ @profile_id = ]          profile_id          OUTPUT ]  
{ [ @profile_name = ]      profile_name      [ [ @ agent_type = ]      agent_type  
}  
[ [ @ profile_type = ]      profile_type      ] { [ @ description = ]      description  
,  
[ , [ @default = ]          default          ]
```

## 变量 [ KH\*2 ]

[ @profile_id = ] profile_id	profile_id 是一个整数，表示 MSagent_configurations 系统表中的配置，并可用作 OUTPUT 参数。若未提供 profile_id，则返回 profile_id 的结果。
[ @profile_name = ] profile_name	profile 的系统名。
[ @agent_type = ] agent_type	agent_type 是一个整数，表示复制代理。其中，1=Snapshot Agent，2=Log Reader Agent，3=Distribution Agent，4=Merge Agent。
[ @profile_type = ] profile_type	profile_type 是一个整数。其中，0=系统配置文件（缺省），1=定制配置文件。
[ @description = ] description	description（描述）是 nvarchar(3000)，它描述配置文件。
[ @default= ] default	default（缺省）是位数据类型，它确定该配置文件是否是复制代理类型的缺省文件。0=非缺省配置文件，1=缺省配置文件。

## sp\_add\_alert

sp\_add\_alert 系统存储过程可添加新警告。

## 语法

```
sp_add_alert { name } [ , message_id ] [ , severity ] [ , enabled ]  
[ , delay_between_responses ] [ , notification_message ]  
[ , include_event_description_in ] [ , database ]  
[ , event_description_keyword_pattern ] [ , {job_id | job_name} ]  
[ , raise_snmp_trap ] [ , performance_condition ] [ , category ]
```

## 变量

name	name 是警告的 varchar(100)型名称。
message_id	message_id 是 sysmessages 系统表中的整型信息错误号。如果使用 severity, message_id 就必须为 0 或 NULL。
Severity	severity 也可用来表示警告, 其值为整数 9 至 25, 110, 120, 130 或 140, 缺省值为 0。如果使用 message_id, severity 必须是 0。
Enabled	enabled 是 tinyint 型数据, 其中 1=启用(缺省值), 0=不启用。
delay_between_responses	delay_between_responses 是一个整数, 表示发出下一个警告之前的延迟时间, 单位为秒(缺省值为 0)。此参数可用来防止短时间内多次报警。
notification_message	notification_message 是一个 varchar(255) 型数据, 缺省值为 NULL 它可用来给警告接收器发送附加信息。
include_event_description_in	include_event_description_in 是 tinyint 型数据(缺省值为 0), 它可确定 NT Event Log 中 SQL Server 错误的描述是否是发送给接收员的部分通知消息。0=None, 1=Email, 2=Pager, 4=NetSend, 7=All。



续表

database	database 是 varchar(64)型数据，若数据库名称使用缺省值 NULL，则表示将对服务器上任意数据库中的错误发出警告。
event_description_keyword	event_description_keyword 是 varchar(100)型数据，缺省值为 NULL，它可描述 Window NT 事件日志中的 SQL Server 错误。LIKE 表达式通配符可以用于表示对象名。
job_id	job_id 是作业的唯一标识符，缺省值为 NULL。 job_id 或 job_name 都是自变量
job_name	job_name 是 CmdExec 或 Transact-SQL varchar(100)型的作业名。它可在警告使用缺省值 NULL 时运行。

## sp\_addalias

sp\_addalias 是为向后兼容而提供的，使用 7.0 版本的程序员可用 Roles 来代替它。

## sp\_addapprole

sp\_addapprole 系统存储程序可创建用于应用程序中安全性的数据库角色。

### 语法

```
sp_addapprole [ @rolename = ] role , [ @password = ] password
```

### 变量

[ @rolename = ]	role	role 是新的数据库角色的系统名。
[ @password = ]	password	password 是系统名称，是访问授给数据库角色的许可的口令。

## sp\_addarticle

sp\_addarticle 系统存储过程可为出版物添加一篇新文章，以用于快照、合并和事务复制中。

### 语法

```
sp_addarticle [ @publication = ] publication ,  
[ @article = ] article , [ @source_table = ] source_table  
[ , [ @destination_table = ] destination_table ]  
[ , [ @vertical_partition = ] vertical_partition ]  
[ , [ @type = ] type ] [ , [ @filter = ] filter ] [ , [ @sync_object
```

```

= ]      sync_object      ]
      [ , [ ins_cmd = ]      ins_cmd      ] [ , [ @del_cmd = ]      del_cmd      ]
[ , [ upd_cmd = ]      upd_cmd      ] [ , [ @creation_script = ]
creation_script      ]
      [ , [ @description = ]      description      ]
      [ , [ @pre_creation_cmd = ]      pre_creation_cmd      ]
      [ , [ @filter_clause = ]      filter_clause      ] [ , [ @schema_option = ]
schema_option      ]
      [ , [ @destination_owner = ]      destination_owner      ] [ , [ @status = ]
status      ]
      [ , [ @source_owner = ]      source_owner      ] [ , [ @sync_object_owner = ]
source_owner      ]
      [ , [ @filter_owner = ]      filter_owner      ] [ , [ @source_object = ]      source_objec
]

```

## 变量

[ @publication = ] publication

[ @article = ] article

[ @source\_table = ]  
source\_table

[ @destination\_table = ]  
destination\_table

[ @vertical\_partition = ]  
vertical\_partition

[ @type = ] type

publication 是包含该文章的出版物的系统名

article 是新文章的系统名。

source\_table 是数据生成的表的 nvarchar(92)型名称。

destination\_table 是目标表的系统名。其缺省值表示使用目标表的 source\_table 名

vertical\_partition 是 nchar(5)型数据，包含的值有 true 或 false，以及缺省值。若该参数设定为 false，则发布所有的列；若设定为 true，则只使用主键。如果文章是垂直划分的，则可使用系统存储过程 sp\_articlecolumn 来增加列。

type 是文章类型的系统名。在该文章类型中，logbased manualfilter=带有手工过滤器的基于日志的文章，logbased manualview =带手动视图的基于日志的文章，logbased manualboth=带手动过滤器和手动视图的基于日志的文章，proc

exec=将存储过程的执行复制到文章的Subscriber中，serializable pro exec=复制存储文件的执行(如果存储过程是在可串行事务的内部)。Type 的缺省值为NULL。

[ @filter = ] filter

filter 是 FOR REPLICATION 存储过程的 nvarchar(92) 型名称，它可水平划分表格，缺省值是 NULL(过滤器程序已创建)。如果该参数有一个值，则必须在以前已运行了系统存储过程 sp\_articlefilter 和 sp\_articleview 来创建一个视图和该存储过程。

[ @sync\_object = ] sync\_object

sync\_object 是用于生成同步输出文件的表或视图的 nvarchar(92) 型名字(缺省值是 NULL)。如果该参数是 NULL，当使用 sp\_articlecolumn 系统存储过程增加列时，将自动创建使输出文件同步的视图。

[ @ins\_cmd = ] ins\_cmd

ins\_cmd 是 nvarchar(255) 型数据，是用来复制插入的方法。其中 NONE=无动作，CALL customproc=创建一个存储过程的调用，以便在 Subscriber 上应用该插入

[ @del\_cmd = ] del\_cmd

[ @upd\_cmd = ] upd\_cmd

[ @creation\_script = ]  
creation\_script

[ @description = ] description

[ @pre\_creation\_cmd = ]  
pre\_creation\_cmd

(可在目标数据库中创建该存储过程，以便传送插入中的每一列)。SQL(缺省值)或 NULL=用语法 insert into <table name> values (column1value , column2value , etc)来复制插入语句。

del\_cmd 是 nvarchar(255)型数据，是用以复制删除的方法。除了删除是通过主键值来进行的外，它与 ins\_cmd 相同。

upd\_cmd 是 nvarchar(255)型数据，是用来复制更新的方法。除了更新是通过传递主键值和文章的其他列来进行的之外，它与 ins\_cmd 相同。

creation\_script 是用以创建目标表的文章模式脚本的 nvarchar(127)型路径和名称，缺省值是 NULL。

description 是 nvarchar(255)型数据，它可描述文章，缺省值为 NULL。

pre\_creation\_cmd 是用以撤销表、删除表或截短表的 nvarchar(10)型预创建命令。其中 None=无命令，Delete=删除目标表，Drop(缺省)=撤销目标表，Truncate=截短目标表。

<pre>[ @filter_clause= ] filter_clause</pre>	<p>filter_clause 是用于水平划分的 ntext 型 WHERE 子句。缺省值为空字符串。</p>
<pre>[ @schema_option = ] schema_option</pre>	<p>schema_option 是 binary(8)型数据，其中 0x00=用 Snapshot 取消编辑脚本的功能，并使用所支持的 CreationScript；0x01=建立对象创建 CREATE TABLE ,CREATE PROCEDURE；0x10=创建成簇索引；0x20=将用户定义的数据类型转换为基本数据类型；0x40=创建非成簇索引；0x80=在主键上包含已声明的引用完整性；0x71=创建创建表、成簇索引和非成簇索引，将用户定义的数据类型转换为基本数据类型(缺省)；0x01=存储过程文章的缺省值。</p>
<pre>[ @destination_owner = ] destination_owner</pre>	<p>owner 是系统名称，代表目标对象的拥有者，缺省值为 NULL。如果出版物具有 ODBC Subscribers，则应使用 NULL。</p>
<pre>[ @status = ] status</pre>	<p>status 是文章选项的 tinyint 型掩码，其中，0，8=在插入语句中包含列名；16=使用参数化的语句；24=使用列名和参数化的语句。</p>
<pre>[ @source_owner = ]</pre>	<p>source_owner 是系统名称，是源对象集</p>

source\_owner

```
[ @sync_object_owner = ]  
sync_object_owner
```

```
[ @filter_owner = ]  
filter_owner
```

```
[ @source_object= ]  
source_object
```

的拥有者，在 Microsoft SQL Server 6.5 Publishers 中设置为 NULL。

sync\_object\_owner 是系统名称，是同步化的对象集的拥有者，在 Microsoft SQL Server 6.5 Publisher 中设置为 NULL。

filter 是系统名称，是过滤器集的拥有者，在 Microsoft SQL Server 6.5 Publishers 中设置为 NULL。

source\_object 是系统名，是源对象的拥有者。如果 source\_table 为 NULL，这个变量必须替换。

## sp\_adddistpublisher

sp\_adddistpublisher 系统存储过程可用本地 Distributor 添加远程 Publisher，以用于快照、合并以及事务复制中。

### 语法

```
sp_adddistpublisher { [ @publisher = ] publisher ,  
[ @distribution_db = ] distribution_db } [ , [ @security_mode = ]  
security_mode ]  
[ , [ @login = ] login ] [ , [ @password = ] password ]  
{ , [ @working_directory = ] working_directory },
```



[ , [ @trusted = ] trusted ] [ , [ @encrypted\_password = ] encrypted\_password ]

[ , [ @thirdparty\_flag = ] thirdparty\_flag ]

### 变量

[ @publisher = ] publisher

publisher 是 Publisher 的系统名。

[ @distribution\_db = ]

分布式数据库的系统名。

distribution\_db

[ @security\_mode = ]

security\_mode

security\_mode 是整型安全模式。其中，0=SQL Server Authentication，1=Distributor 在 Windows NT 上运行，2=Distributor 在 Windows 95 上运行，NULL=缺省值。

[ @login = ] login

login 是系统名，缺省值为 sa，当 security\_mode 项为 0=SQL Server Authentication 时，需要 Login。

[ @password = ] password

password 是登录的系统名口令，由复制代理使用以连接 Publisher。

[ @working\_directory = ]

working\_directory

working\_directory 是目录的 nvarchar(255) 型名称，其中，出版物的数据和模式文件以 UNC 格式保存。

[ @trusted = ] trusted

trusted 是一个 nchar(5) 型值，其为 True 时，表明远程 Publisher 是否与本地

Distributor 使用相同的口令。新增了两个委托登录映射：sa to sa，和 distributor\_admin to distributor\_admin。在连接远程 Publisher 与 Distributor 时，不需使用口令。当 trusted 值为 false 时，表明增加了一个非委托映射：distributor\_admin to distributor\_admin。在连接远程 Publisher 和 Distributor 时，需使用口令。

encrypted 口令是位数据类型，缺省值为 0，编辑脚本时使用。

thirdparty\_flag 是位数据类型，它确定出版物是否是除 Microsoft SQL Server 之外的数据源。其中，0=Microsoft SQL Server 数据库，1=除 Microsoft SQL Server 外的数据源。

```
[ @encrypted_password = ]  
encrypted_password  
[ @thirdparty_flag = ]  
thirdparty_flag
```

## sp\_adddistributiondb

sp\_adddistributiondb 用 Distributor 模式添加新 Distributor 数据库，以包含用

于快照、合并和事务复制中的分布程序及其历史。

语法

```
sp_adddistributiondb { [ @ database = ]      database      }
[ , [ @data_folder = ]      data_folder      ] [ , [ @data_file = ]      data_file
]
[ , [ @data_file_size = ]      data_file_size      ] [ , [ @log_folder = ]
log_folder      ]
[ , [ @log_file = ]      log_file      ] [ , [ @log_file_size = ]      log_file_size
]
[ , [ @min_distretention = ]      min_distretention      ]
[ , [ @max_distretention = ]      max_distretention      ]
[ , [ @history_retention = ]      history_retention      ]
[ , [ @security_mode = ]      security_mode      ] [ , [ @login= ]      login
]
[ , [ @password = ]      password      ]
[ , [ @encrypted_password = ]      encrypted_password      ]
[ , [ @createmode = ]      createmode      ]
```

变量

[ @database = ] database database 是数据库的系统名。  
[ @data\_folder = ] data\_folder data\_folder 是包含数据库文件的目录的 nvarchar(255)型名称。缺省值为安

[ @data\_file = ] data\_file

[ @data\_file\_size= ]  
data\_file\_size

[ @log\_folder = ] log\_folder

[ @log\_file = ] log\_file

[ @log\_file\_size = ]  
log\_file\_size

[ @min\_distretention = ]  
min\_distretention

[ @max\_distretention= ]  
max\_distretention

装 SQL Server 的路径 (若安装了 SQL Server 并包含缺省文件, 则该参数的缺省值为 \mssql17\data)。

data\_file 是数据库文件的 nvarchar(255) 型名称。如果没有提供该参数, 就用数据库名创建该名称。

data\_file\_size 是一个整数, 单位是 MB, 它表明起始数据文件的大小, 缺省值为 2MB。

log\_folder 是数据库日志文件所在目录的 nvarchar(255) 型名称。

log\_file 是日志文件的 nvarchar(255) 型名称。如果没有提供该参数, 则用数据库名创建 log\_file 名。

log\_file\_size 是一个整数, 单位是 MB, 它表明起始日志文件的大小, 缺省值 0MB 表示 512KB。

min\_distretention 是一个整数, 表明删除事务之前的最少小时数。缺省值为 0。

max\_distretention 是一个整数, 表明删除事务之前的最多小时数, 其缺省值

```
[ @history_retention = ]  
history_retention  
[ @security_mode = ]  
security_mode
```

```
[ @login = ] login
```

```
[ @password = ] password
```

```
[ @encrypted_password = ]  
encrypted_password
```

```
[ @createmode = ] createmode
```

为 24 小时。对每个失效的 Subscriptions 而言，在此之后的未激活的 Subscriptions 会通过 Raiserror 21011 而失效。

history\_retention 是一个整数，表明保存历史的小时数。缺省值为 48 小时。security\_mode 是一个整数，表明 Distributor 中的内容。其中，0=SQL Server Authentication，1=Windows NT Authentication，缺省值为 0。

login 是系统名，缺省值为 sa。如果 security\_mode 为 0=SQL Server Authentication，就需要该参数。

password 是登录的系统名口令，复制代理用它连接到 Distributor 上。

encrypted\_password 是位数据类型，缺省值为 0，编辑脚本时使用。

createmode 是一个整数，其中，0=创建用于附件的 db，1=创建 db 或使用已有但未附加上的 db(缺省值)，2=创建仅用于 instdist 和解除附属关系的 db。

## sp\_adddistributor

sp\_adddistributor 系统存储过程可在 sysservers 表中为带有属性数据的 Distributor 添加一个条目，用于快照、合并和事务复制中。

### 语法

```
sp_adddistributor { [ @distributor = ] distributor },
  [ , [ @heartbeat_interval = ] heartbeat_interval ] [ , [ @login = ]
login ]
  [ , [ @password = ] password ]
```

### 变量

[ @distributor = ] distributor distributor 是服务器名的系统名称。

[ @heartbeat\_interval = ] heartbeat\_interval heartbeat\_interval 是一个整数，它可创建检查复制代理的作业，并表示代理可在不记录信息的情况下运行的最大分钟数。缺省值为 10 分钟。

[ @login = ] login login 是 distributor\_admin 的登录系统名。缺省值为 sa。

[ @password = ] password password 是 distributor\_admin 登录的系统名称口令，缺省值为 NULL，用于发布者 RPC 连接的链接服务器条

目。如果发布器是本地发布器，distributor\_admin 的口令就设置为新值。

## sp\_addextendedproc

sp\_addextendedproc 系统存储过程可注册扩展存储过程的名称。它必须在主系统数据库环境下运行，并在 syscomments 和 sysobjects 系统表中创建行。

### 语法

```
sp_addextendedproc [ @funcname = ]      procedure      , [ @dllname = ]  
                    dll  
                    变量
```

[ @funcname = ] procedure	procedure 是 nvarchar(257)型数据，是在动态链接库(DLL)中调用的函数名。
[ @dllname = ] dll	DLL 是 DLL 的 varchar(255)型名称。

## sp\_addgroup

sp\_addgroup 系统存储过程在当前数据库中增加一个角色，它可以向后兼容。而使用 7.0 版本的程序员使用角色来代替它。

## 语法

```
sp_addgroup [ @rolename = ]          role
```

变量 `role` 是要添加的角色名。

## sp\_add\_category

`sp_add_category` 系统存储过程为服务器添加一个类别。

### 语法

```
sp_add_category [[ @class = ] class , ] [[ @type = ] type , ] { [ @name = ] name }
```

### 变量

[ @class = ] class	class 是类别的 varchar(8)型类，其中 JOB=增加一项作业(类型必须是 MULTISERVER)，ALERT=增加一个警告，OPERATOR=增加一个操作员。
[ @type = ] type	type 是要添加的类别的 varchar(12)类型，其中 LOCAL=本地计算机(缺省)，MULTISERVER=MSX 域的成员，NONE=无。
[ @name = ] name	name 是服务器的 varchar(100)型名称。



## sp\_add\_job

sp\_add\_job 系统存储过程可创建一项新作业。

### 语法

```
sp_add_job [ @job_name = ] job_name }
[ , [ @enabled = ] enabled ] [ , [ @description = ] description
]
[ , [ @start_step_id = ] step_id ] [ , [ @category_name = ] category
]
[ , [ @category_id = ] category_id ] [ , [ @owner_login_name = ]
login ]
[ , [ @notify_level_eventlog = ] eventlog_level ]
[ , [ @notify_level_email = ] email_level ]
[ , [ @notify_level_netsend = ] netsend_level ]
[ , [ @notify_level_page = ] page_level ]
[ , [ @notify_email_operator_name = ] email_name ]
[ , [ @notify_netsend_operator_name = ] netsend_name ]
[ , [ @notify_page_operator_name = ] page_name ]
[ , [ @delete_level = ] delete_level ] [ , [ @job_id = ] job_id
OUTPUT ]
```

## 变量

job_name	job_name 是作业的系统名称。
Enabled	enabled 决定作业是可执行还是不能执行。其中 1=可执行，0=不能执行，缺省值为 1。
description	discription 是作业的 nvarchar(512) 型描述，缺省值为“无描述项”。
step_id	step_id 是一个整数，表明作业的第一步骤的数目，缺省值为 1。
category	catelog 是作业类别的系统名，缺省值为 NULL。
Category_id	category_id 是一个整数，表示作业类别，缺省值为 NULL。
login	login 是登录的系统名。该登录许可用以运行作业，缺省值为 NULL。
Eventlog_level	eventlog_level 是一个整数，它确定何时添加创建作业的 NT 事件日志项，其中 0=不添加，1=创建成功时添加，2=创建失败时添加，3=成功失败均添加。缺省值为 2。
email_level	email_level 是一个决定何时发送 e-mail 的整数。其中 0=不发送，1=成功时发送，2=失败时发送，3=成功失败都发送。缺省值为 0。
Netsend_level	network_level 是一个决定何时发送网络信息

的整数。0=不发送，1=成功时发送，2=失败时发送，3=成功失败皆发送，缺省值为0。

page\_level

page\_level 是一个决定何时将寻呼发送到寻呼机上的整数。0=不发送，1=成功时发送，2=失败时发送，3=成功失败均发送。缺省值为0。

email\_name

e-mail\_name 是在 email\_level 上发送 e-mail 的人的 nvarchar(100) 型姓名，缺省值为 NULL。

net send\_name

net send\_name 是在 net send\_level 上发送网络信息的操作员的 nvarchar(100) 型名字。缺省值为 NULL。

page\_name

page\_name 是在 page\_level 上寻呼的人的 nvarchar(100) 型名字，缺省值为 NULL。

delete\_level

delete\_level 是一个决定何时删除作业的整数，其中 0=不删除，1=成功时删除，2=失败时删除，3=成功或失败均删除。缺省值为0。

job\_id

job\_id 是作业的唯一标识符，缺省值为 NULL。

## sp\_add\_jobschedule

sp\_add\_jobschedule 系统存储过程将作业列入计划内。

语法

```

sp_add_jobschedule { [ @job_id = ]      job_id      | [ @job_name = ]
job_name      }
    { [ @name = ]      name      } [ [ @enabled = ]      enabled      ] [ [ @freq_type
= ]      freq_type      ]
    [ , [ @freq_interval = ]      freq_interval      ]
    [ , [ @freq_subday_type = ]      freq_subday_type      ]
    [ , [ @freq_subday_interval = ]      freq_subday_interval      ]
    [ , [ @freq_relative_interval = ]      freq_relative_interval      ]
    [ , [ @freq_recurrence_factor = ]      freq_recurrence_factor      ]
    [ , [ @active_start_date = ]      active_start_date      ]
    [ , [ @active_end_time = ]      active_end_date      ]
    [ , [ @active_start_time = ]      active_start_time      ]
    [ , [ @active_end_time = ]      active_end_time      ]

```

### 变量

job_id	job_id 是作业的唯一标识符，缺省值为 NULL。可提供 job_id 或名称。
job_name	job_name 是要列入计划的作业的 varchar(100) 型名称。缺省值为 NULL。可提供 job_id 或名称。
name	name 是包含作业的计划的 varchar(100) 型名称。
enabled	enabled 是 tinyint 型数据，它决定

freq_type	<p>作业能否在计划中运行，其中 1=能运行，0=不能运行，缺省值为 1。</p> <p>freq_type 是决定作业何时运行的整数。缺省值为 0。其中 1=一次，4=每天，8=每周，16=每月，32=每月(根据 freq_interval 参数)。64=当 SQL Server Agent 启动时运行。</p>
freq_interval	<p>freq_interval 是决定作业运行日期的整数，缺省值为 0，其中 1=周日，2=周一，3=周二，4=周三，5=周四，6=周五，7=周六，8=白天，9=平日，10=每周末白天。</p>
freq_subday_type	<p>freq_subday_type 是一个整数，代表 freq_subday_interval 的单位，缺省值为 0。其中 1=在特定时间。2=秒，3=分钟，4=小时。</p>
freq_subday_interval	<p>freq_subday_interval 是一个整数，缺省值为 0，它决定每个作业运行的时间间隔，该间隔的单位是 freq_subday_type。</p>
freq_relative_interval	<p>freq_relative_interval 是一个整数，决定每个月中的 freq_interval。</p>

freq\_recurrence\_factor

缺省值为 0，1=第一个，2=第二个，4=第三个，8=第四个，16=最后一个。

freq\_recurrence\_factor 是代表作业运行间隔月数的整数。缺省值为 0。

active\_start\_date

active\_start\_date 是一个整数，代表作业开始运行的日期。缺省值为 NULL，格式为：YYYYMMDD。

active\_end\_date

active\_end\_date 是一个整数，代表作业结束运行的日期。缺省值为 99991231(9999 年 12 月 31 日)。

active\_start\_time

active\_start\_time 是一个整数，代表作业开始运行的时刻，缺省值为 000000(12 : 00 : 00AM)，格式为 HHMMSS。

active\_end\_time

active\_end\_time 是一个整数，代表作业结束运行的时刻，缺省值为 235959(11 : 59 : 59PM)。

sp\_add\_jobserver

sp\_add\_jobserver 系统存储过程将服务器添加到那些能够运行作业的服务器中。

## 语法

```
sp_add_jobserver [ { job_id | job_name }, ] { server }  
[ automatic_post ]
```

## 变量

job_id	job_id 是要添加到服务器中的作业的单一标识符，缺省值为 NULL，可提供 job_id 或名称。
job_name	job_name 是列入计划的作业的 varchar(100)型名称。缺省值为 NULL。可提供 job_id 或名称。
server	server 是为运行作业而添加的服务器的 varchar(30)型名称。它是 (本地) 或目标服务器名。

## sp\_add\_jobstep

sp\_add\_jobstep 系统存储过程可为作业创建一个步骤。可以用 sp\_add\_jobschedules 系统存储程序来给计划添加步骤。

## 语法

```
sp_add_jobstep { [ @job_id = ] job_id | [ @job_name = ] job_name  
                ] }  
                [ , [ @step_id = ] step_id ], { [ @step_name = ] step_name }  
                [ , [ @subsystem = ] subsystem ] [ , [ @command = ] command  
                }
```

```

[ , [ @additional_parameters = ]      parameters      ]
[ , [ @cmdexec_success_code = ]      code      ]
[ , [ @on_success_action = ]      success_action      ]
[ , [ @on_success_step_id = ]      success_step_id      ]
[ , [ @on_fail_action = ]      fail_action      ]
[ , [ @on_fail_step_id = ]      fail_step_id      ] [ , [ @server = ]      server
]
[ , [ @database_name = ]      database      ] [ , [ @database_user_name = ]
user      ]
[ , [ @retry_attempts = ]      retry_attempts      ] [ , [ @retry_interval = ]
retry_interval      ]
[ , [ @os_run_priority = ]      run_priority      ] [ , [ @output_file_name = ]
file_name      ]
[ , [ @flags = ]      flags      ]
变量

```

job\_id

job\_id 是作业的唯一标识符，缺省为 NULL。可提供 job\_id 或名称。

job\_name

job\_name 是作业的 varchar(100) 型名称，缺省值为 NULL。可提供 job\_id 或名称。

step\_id

step\_id 是表示作业步骤的整数。缺省值为 NULL。



step\_name

subsystem

command

step\_name 是步骤的系统名称。

subsystem 是 varchar(60)型数据，是 SQL Server Enterprise Manager 用以运行命令的子系统。缺省值为 TSQL。

command 是发送给 SQL Server Agent 来运行的 nvarchar(3200)型命令。缺省值为 NULL。命令可以带有下列代号：[ A-DBN ] =数据库名称；[ A-SVE ] =服务器名称；[ A-ERR ] =错误号；[ A-SEV ] =错误的严重级别；[ A-MSG ] =信息文本；[ DATE ] =格式为 YYYYMMDD 的当前日期；[ JOBID ] =作业 ID；[ MACH ] =计算机名称；[ MSSA ] =主 SQLServerAgent 名称；[ SQLDIR ] =SQL Server 的安装目录 (C : \ MSSQL7)；[ STEPCT ] =该步骤运行的时间。[ STEPID ] =步骤 ID；[ TIME ] =格式为 HHMMSS 的当前时间；[ STRTTM ] =格式为 HHMMSS 作业开始时间；[ STRDRT ] =格式为 YYYYMMDD 的作业开始日期。

code	code 是运行成功所返回的整数码，缺省值为 0。
success_action	success_action 是 tinyint 型数据，它决定返回成功码时应执行的操作。其中 1=成功后退出系统(缺省值)；2=失败后退出；3=继续进行下一步；4=转入步骤 success_step_id。
success_step_id	success_step_id 是该作业中要运行的步骤，其中成功码和 success_action 为 4，缺省值为 0。
fail_action	fail_action 是 tinyint 型数据，是非编码的返回值，其中 1=成功后退出；2(缺省值)=失败后退出；3=继续进行下一步；4=转入步骤 fail_step_id。
fail_step_id	fail_step_id 是当非编码和 success_action 为 4 时，该作业中应运行的步骤。缺省值为 0。
server	server 是运行该步骤的服务器的 nvarchar(30) 型名称，缺省值为 NULL。
database	database 是运行该步骤的数据库的系

<code>user</code>	系统名称，缺省值为 NULL。 <code>user</code> 是用以运行该步骤的用户帐号的系统名称。缺省值为 NULL。
<code>retry_attempts</code>	<code>retry_attempts</code> 是一个整数，它决定当步骤失败时重复运行的次数，缺省值为 0。
<code>retry_interval</code>	<code>retry_interval</code> 是一个整数，表示重复运行所间隔的秒数，缺省值为 0。
<code>run_priority</code>	<code>run_priority</code> 是一个整数，代表运行步骤所优先使用的 OS。其中 -15=空转，-1=低准；0=标准(缺省值)；1=高于标准；15=时间极限。
<code>file_name</code>	<code>file_name</code> 是输出文件的 <code>nvarchar(200)</code> 型名称，该输出文件中，该步骤的输出被省略，而可以包含命令代号。

## `sp_addlinkedserver`

`sp_addlinkedserver` 系统存储过程可定义一个链接服务器。该服务器允许通过 OLE DB 数据源访问分布式查询；它也允许远程存储过程在两个 SQL Server 之间调用。`DataSource`，`Location`，`ProviderString` 变量可标识链接服务器的数据

库。如果这些变量的任何一个为 NULL，就不能设置相应的 OLE DB 初始化属性。如果希望数据源是 Microsoft SQL Server，可使用 ProviderName, DataSource, Location 和 ProviderString 的 NULL 值。将使用 SQL Server OLE DB。只需提供 ProductName 和 Server 变量。

SQL Server OLE DB Provider Product\_name= SQL Server (缺省值) Provider\_name=N SQLOLEDB data\_source 是服务器 Microsoft OLE DB Provider for Jet Product\_name= Provider\_name= Microsoft.Jet.OLEDB.4.0 , data\_source=路径 \ Microsoft Access .mdf 数据库文件的路径 / 文件名 Microsoft OLE DB Provider for Oracle Product\_name= Oracle Provider\_name= MSDAORA , data\_source=在服务器 SQL Server 上创建的 Oracle 数据库范例的 SQL Net Alias OLE DB Provider for ODBC(使用 data\_source 参数) Provider\_name= Provider\_name= MSDASQL , data\_source= LocalServer OLE DB Provider for ODBC Product\_name= Product\_name= MSDASQL , provider\_string= DRIVER={SQL Server} ; SERVER=服务器名 ; UID=注册 ; PWD=口令 ;

### 语法

```
sp_addlinkedserver
{
[ @server = ] server ,
[ @srvproduct = ] product_name ,
[ @provider = ] provider_name ,
[ @datasrc = ] data_source ,
```

```
[ @ location = ]      location    ,  
[ @ provstr = ]      provider_string  ,  
[ @ catalog = ]      catalog  
}
```

变量

ServerName

ProductName

ProviderName

ServerName 是链接服务器的名称。

ProductName 是将作为 OLE DB 提供商使用的产品名称，缺省值为 NULL。它是一个 nvarchar 型数据，可含 128 个字符。数据类型 nvarchar 可存储 Unicode 字符数据

ProviderName 必须是一个唯一的名称，代表在 ProductName 中指定的 OLE DB 提供商，提供商名称是 nvarchar 型数据，可含 128 个字符。数据类型 nvarchar 可存储 Unicode 字符数据，缺省值为 NULL。ProviderName 必须在 NT 注册表中注册。数据提供者是 OLE DB 界面调用提供不同的数据源。Microsoft 中可用的 OLE DB 提供者可与 SQL Server 数据源，SQLOLEDB 内

部提供者以及 ODBC 的 OLE DB 提供者一同工作

DataSource

DataSource 是提供给由 OLE DB 提供商的数据源名称。提供商名称是 nvarchar 型数据，可含 4000 个字符

Location

Location 是由 OLE DB 提供商解释的数据库位置。提供商名称是 nvarchar 型数据，可含 4000 个字符。数据类型 nvarchar 将存储 Unicode 字符数据，缺省值为 NULL

ProviderString

ProviderString 指定唯一的数据库源。提供商名称是 nvarchar 型数据，可含 4000 个字符。数据类型 nvarchar 可存储 Unicode 字符数据，缺省值为 NULL

[ @catalog = ] catalog

catalog 是类别的系统名称，当与 OLE DB 提供商连接时使用该名称。缺省值为 NULL。

### 范例

以下是 sp\_addlinkedserver 存储过程的一个范例：

```
sp_addlinkedserver RemoteServerName , , MSDASQL ,  
LocalServerName ,NULL,NULL
```

在本例中，定义了一个使用 OLE DB ODBC 提供商和 Data Source DSN 变量的远程链接服务器。

范例

以下是 sp\_addlinkedserver 存储过程的另一个范例：

```
EXEC sp_addlinkedserver RemoteServerName , SQL Server ,  
NULL,NULL,NULL,NULL
```

在本例中，定义了一个使用 OLE DB 提供商的链接服务器。

范例

以下是 sp\_addlinkedserver 存储过程的另一个范例：

```
sp_addlinkedserver RemoteServerName , MSDASQL , NULL,NULL,  
DRIVER={SQL Server} ; SERVER=LocalServer ; UID=sa ; PWD= ;
```

在本例中，定义了一个使用 OLE DB ODBC 提供者 and ProviderString 变量的链接服务器。

范例

以下是 sp\_addlinkedserver 存储过程的另一个范例：

```
sp_addlinkedserver AccessRemoteServer ,  
Microsoft.Jet.OLEDB.4.0,
```

```
C : \MSOffice \ Access \ Samples \ Northwind.mdb ,NULL,NULL
```

在本例中，定义了一个使用 Microsoft Access OLE DB 提供商的链接服务器。必须在计算机中安装 Microsoft Access。

## sp\_addlinkedserverlogin

sp\_addlinkedserverlogin 存储过程可将本地服务器中的注册 ID 映射到另一台远程服务器中的注册 ID 上。在这种工作模式下，用户可以注册到本地数据库中，并运行一使用另一个服务器中的表格的查询。这将确定用户在把本地服务器与远程服务器相连接时使用谁的许可。

该存储过程有一些不同之处，因为如果 Useself 变量被设置为 true，那么，即使提供了 RemoteUser 和 RemotePassword 变量，该存储过程也不会使用这两个变量。如果 LocalLogin 变量被设置为 NULL，或根本未指明，则 RemoteUser 和 RemotePassword 的映射将应用到所有与远程服务器相连接的注册中。

值得注意的是，总存在一个缺省映射，该映射是通过运行新存储过程 sp\_addlinkedserver 而创建的。该缺省映射是在连接代表注册 ID 的链接服务器时使用的本地注册 ID 许可。新的存储过程 sp\_addlinkedserverlogin 可以用来改变这个缺省映射，并创建与每个注册 ID 相适应的映射。用户的许可取决于为远程服务器中的注册 ID 给定的许可内容，它们可以不同于本地服务器中给定的许可。

### 语法

```
sp_addlinkedserverlogin [ @rmtsrvname = ]          rmtsrvname
[ , [ @useself = ]          useself          [ , [ @locallogin = ]          locallogin
[ , [ @rmtuser = ]          rmtuser          [ , [ @rmtpassword = ]          rmtpassword
]]]]
```



## 变量

[ @rmtsrvname = ] rmtsrvname  
[ , [ @useself = ] useself

[ , [ @locallogin = ] locallogin

[ , [ @rmtuser = ] rmtuser

[ , [ @rmtpassword = ]

rmtsrvname 远程服务器的系统名称。  
useself 是 varchar(8)型数据，其值为 true 或 false。True 值表示登录将使用其 SQL Server 验证许可与远程服务器相连接 (True 值对 NT 验证用户无效)。如果使用 true 值，就不使用 Remote User 和 RemotePassword 变量。False 值表示 RemoteUser 和 RemotePassword 变量将用来连接远程服务器。如果使用 NULL，就将用 rmtuser 和 rmtpassword 连接远程服务器。

locallogin 是系统名称，缺省值为 NULL，它表示该参数将引用所有试图与远程服务器相连接的本地注册。如果该参数不是 NULL，该注册 ID 必须在本地服务器上的 syslogins 表中拥有一行。  
rmtuser 是一个系统名称，当变量 UseSelf 设置为 NULL 值时使用。它是将用来链接到远程服务器上的用户，缺省值为 NULL。

这是一个与远程用户相联系的远程口

rmtpassword

令。缺省值为 NULL。

sp\_addlogin

sp\_addlogin 系统存储过程可增加一 SQL Server 验证注册, 以与 Microsoft SQL Server 相连接。

语法

```
sp_addlogin [ @loginame = ] login [ , [ @passwd = ] password  
[ , [ @defdb = ] database [ , [ @deflanguage = ] language  
[ , [ @sid = ] sid [ , [ @encryptopt = ] encryption_option ]]]]
```

变量

[ @loginame = ] login

login 是注册的系统名称

[ @passwd = ] password

password 是口令的系统名称, 缺省值为 NULL。

[ @defdb = ] database

database 是注册在进入系统后所使用的数据库的缺省数据库系统名, 缺省值为主系统数据库。最好不要让用户使用缺省的主数据库注册, 而将该参数设置为用户数据库

[ @deflanguage = ] language

language 是用户登录到 SQL Server

```
[ @sid = ] sid
```

```
[ @encryptopt = ]  
encryption_option
```

## sp\_addmergearticle

sp\_addarticle 系统存储过程可为合并出版物添加一篇文章。

语法

```
sp_addmergearticle { @publication = } publication , [ @article = ]  
article ,  
[ @source_object = ] source_object }
```

后使用的缺省语言。其缺省值是由 sp\_configure 存储程序定义的语言。sid 是 varbinary(16)型安全标识号，该参数仅在数据库用户没有注册，是一个孤立的用户时才使用。所提供的安全标识号须是唯一的。提供该参数，SQL Server 将不生成它自己的安全标识号。

encryption\_option 决定是否加密口令。其中 NULL= 加密口令；skip\_encryption= 不加密口令，skip\_encryption\_old= 不加密口令，但 SQL Server 的以前版本中加密口令(只为满足升级需要)。

```

[ , [ @type = ]      type      ]
[ , [ @description = ]      description      ]
[ , [ @column_tracking = ]      column_tracking      ] [ , [ @status = ]
status      ]
[ , [ @pre_creation_cmd = ]      pre_creation_cmd      ]
[ , [ @creation_script = ]      creation_script      ]
[ , [ @schema_option = ]      schema_option      ]
[ , [ @subset_filterclause = ]      subset_filterclause      ]
[ , [ @article_resolver = ]      article_resolver      ]
[ , [ @resolver_info = ]      resolver_info      ]
[ , [ @source_owner = ]      source_owner      ]

```

### 变量

[ @publication = ]	publication	publication 是包含文章的出版物的系统名称。
[ @article = ]	article	article 是文章的系统名。它必须是一个表名(例如 MyTable)。
[ @source_object = ]	source_object	source_object 是要添加的文章所出自的源对象的 nvarchar(386)型名称(例如 bdo.MyTable)
[ { @type = ]	type	Type 是文章类型的系统名称, 现在只允许表类型。
[ @description = ]	description	description 是文章的 nvarchar(255)

```
[ @column_tracking = ]  
column_tracking
```

```
[ @status = ] status
```

```
[ @pre_creation_cmd = ]  
pre_creation_cmd
```

```
[ @creation_script = ]  
creation_script
```

型描述，缺省值为 NULL。

column\_tracking 是 nvarchar(10)型数据，表示列级跟踪的真假指示器，缺省值 false 表示关闭列级跟踪。如果已用合并复制发布了表格，就应使用相同的列级跟踪值，并将该变量设置为 true。

status 是文章的 nvarchar(10)型状态，缺省值 unsynced 表明初始数据已设定。如果状态是活动的，就运行初始处理脚本。

pre\_creation\_cmd 是 nvarchar(10)型的预先创建方式。其中，None=如果 Subscriber 中已存在该表，则对表不进行任何操作；Delete=按子集过滤器中的 WHERE 子句进行删除和过滤；Drop=在重建表之前将表撤消(缺省值)；Truncate=与 delete 相同，但删除的是页，而不是表(无 WHERE 子句)。

creation\_script 是 nvarchar(255)型的文章预建脚本(可选项，缺省值

```
[ @schema_option = ]  
schema_option
```

```
[ @subset_filterclause = ]  
subset_filterclause
```

```
[ @article_resolver = ]  
article_resolver
```

为 NULL)。

schema\_option 是生成文章模式的二进制(8)位图。其中,0x00=用 Snapshot 关闭编辑脚本功能,而使用所提供的 CreationScript; 0x01=生成对象创建 (CREATE TABLE; CREATE PROCEDURE); 0x10=生成成簇索引; 0x20=将用户定义的数据类型转化为基本数据类型; 0x40=生成非成簇索引; 0x80=在主键上提供已声明的引用完整性。0xf1(缺省)=制作创建表,创建成簇索引,创建非成簇索引,将用户定义的数据类型转化成基本数据类型; 0x01=存储过程文章的缺省值。

subset\_filterclause 是一个 nvarchar(2000)型的 WHERE 子句,它确定水平过滤。

article\_resolver 是文章的 varchar(255)型的定制解析程序。存储过程 sp\_enumcustomsolvers 列举出了的定制解析程序。如果该参数设置为 NULL 值,则使用系统提供的解

```
[ @resolver_info = ]  
resolver_info
```

```
[ , [ @source_owner = ]  
source_owner
```

### 范例

下面是 sp\_addmergearticle 存储过程的一个范例：

```
EXEC sp_addmergearticle @publication = mergeMyDB ,  
@ article = MyTable , @source_object = dbo.MyTable ,  
@ type = table , @description = null, @column_tracking = true ,  
@ status = unsynced , @pre_creation_cmd = drop ,  
@ creation_script = null, @ schema_option = 0x0000000000000071,  
@ article_resolver = null, @ subset_filterclause = null
```

在本例中，给出版物增加了一篇合并文章。

## sp\_addmergefilter

sp\_addmergefilter 通过连接另一个表来增加一个合并过滤器，以进行分区，它在带有外来键的文章中可用于带有过滤器的已发布的主键表。

### 语法

析程序。

resolver\_info 是用作定制解析程序的存储过程的系统名，缺省值为 NULL。

source\_owner 是一个系统名称，是源目标的拥有者。如果该参数设置为 NULL，则当前用户就是缺省值。

```

sp_addmergefilter { [ @publication = ] ,      publication ,
[ @ article = ]      article      , [ @ filtername = ]      filtername ,
[ @ join_articlename = ]      join_articlename  }
[ , [ @ join_filterclause = ]      join_filterclause  ]
[ , [ @ join_unique_key = ]      join_unique_key  ]

```

变量

[ @publication = ]	publication	publication 是出版物的系统名。
[ @article = ]	article	article 是该文章的系统名。
[ @filtername = ]	filtername	filtername 是过滤器的系统名。
[ @join_articlename = ]	join_articlename	join_articlename 是 nvarchar(258) 型的表名，该表与基础表连接在一起，并包含在出版物中。
[ @join_filterclause = ]	join_filterclause	join_filterclause 是能进行连接的 nvarchar(2000)型过滤器子句，它在该存储过程中仅定义布尔过滤器。
[ @join_unique_key = ]	join_unique_key	join_unique_key 是一个整数，缺省值为 0。

## sp\_addmergepublication

sp\_addmergepublication 系统存储过程增加一个合并出版物。



## 语法

```
sp_addmergepublication { [ @publication = ]      publication      }
[ , [ @description = ]      description      [ , [ @retention = ]      retention
]
[ , [ @sync_mode = ]      sync_mode      , [ @allow_push = ]      allow_push
]
[ , [ @allow_pull = ]      allow_pull      ] [ @allow_anonymous = ]
allow_anonymous ]
[ , [ @enabled_for_internet = ]      enabled_for_internet      ]
[ , [ @centralized_conflicts = ]      centralized_conflicts      ]
[ , [ @default_access = ]      default_access      ]
[ , [ @dynamic_filters = ]      dynamic_filters      ]
```

## 变量

[ @publication = ]	publication	publication 是新的合并出版物的系统名。
[ @description = ]	description	description 是 nvarchar(255)型对出版物的描述。
[ @retention = ]	retention	retention 是一个整数，代表保存给定的出版物的天数，缺省值为 60 天。在保留期间，订阅必须合并，否则它将不同步。
[ @sync_mode = ]	sync_mode	sync_mode 是 nvarchar(10)型同步模

[ @allow\_push = ] allow\_push

[ @allow\_pull = ] allow\_pull

[ @allow\_anonymous= ]  
allow\_anonymous

[ @enabled\_for\_internet = ]  
enabled\_for\_internet

[ @centralized\_conflicts = ]  
centralized\_conflicts

[ @default\_access = ]

式。缺省值为 native。Native =成批拷贝；Character =字符成批拷贝。

allow\_push 是 true 或 false 值 [ nvarchar(5)型 ]，它确定出版物是否允许推出订阅。

allow\_pull 是 true 或 false 值 [ nvarchar(5)型 ]，它决定出版物是否允许推入订阅。

allow\_anonymous 是 true 或 false 值 [ nvarchar(5) ] 型，决定出版物是否允许匿名订阅。

enabled\_for\_internet 是 nvarchar(5) 型数据，缺省值为 false，它确定出版物能否用于 Internet。FTP 用来向 Subscriber 发送快照。

centralized\_conflicts 是 nvarchar(5) 型数据，它确定 Publisher 中是否存有冲突记录。缺省值为 true。False 意味着在 Subscriber 上存储。

default\_access 是 nvarchar(5)型数

default\_access  
[ , [ @dynamic\_filters = ]  
dynamic\_filters

据，缺省值为 true。  
dynamic\_filters 是 nvarchar(5) 型数据，它确定是否在动态子句上过滤出版物，缺省值为 false。

## sp\_addmergepullsubscription

sp\_addmergepullsubscription 系统存储过程可创建推入合并订阅。

### 语法

```
sp_addmergepullsubscription { [ @publication = ]      publication }  
[ , [ @publisher = ]      publisher      ] [ , [ @publisher_db = ]  
publisher_db      ]  
[ , [ @subscriber_type = ]      subscriber_type      ]  
[ , [ @subscription_priority = ]      subscription_priority      ]  
[ , [ @sync_type = ]      sync_type      ] [ , [ @description = ]      description  
]
```

### 变量

[ @publication = ] publication  
[ @publisher = ] publisher

publication 是出版物的系统名称。  
publisher 是 Publisher 的系统名称  
(服务器名称)。

[ @publisher\_db = ]

publisher\_db 是 Publisher 数据库的

```

publisher_db
[ @subscriber_type = ]
subscriber_type

[ @subscription_priority = ]
subscription_priority

[ @sync_type = ] sync_type

[ @description = ] description

```

系统名称，缺省值为 NULL。

subscriber\_type 是一个 nvarchar(15) 型数据，缺省值为 local，其值可以是 global，local 或 anonymous。

subscription\_priority 是 real 型数据，缺省值为 NULL，它用来解决冲突。Local=0.0； Anonymous=0.0

sync\_type 是 nvarchar(15) 型订阅同步类型，缺省值为自动。Automatic=模式和数据转换，Nosync=无模式和数据转换

description 是 varchar(255) 型推入订阅描述者，缺省值为 NULL。

## sp\_addmergepullsubscription\_agent

sp\_addmergepullsubscription\_agent 系统存储过程用以在 Subscriber 中添加合并推入订阅。

### 语法

```

sp_addmergepullsubscription_agent [[ @name = ] name ]
{, [ @publisher = ] publisher , [ @publisher_db = ] publisher_db

```

```
,
[ @publication = ]      publication      }
[ , [ @publisher_security_mode = ]      publisher_security_mode      ]
[ , [ @publisher_login = ]      publisher_login      ]
[ , [ @publisher_password = ]      publisher_login      ]
[ , [ @publisher_distributor_password = ]      publisher_encrypted_login
]
[ , [ @encrypted_Password= ]      encrypted_password      ]
[ , [ @subscriber = ]      subscriber      ]
[ , [ @subscriber_db = ]      subscriber_db      ]
[ , [ @subscriber_security_mode = ]      subscriber_security_mode      ]
[ , [ @subscriber_login = ]      subscriber_login      ]
[ , [ @subscriber_password = ]      subscriber_password      ]
[ , [ @distributor = ]      distributor      ]
[ , [ @distributor_security_mode = ]      distributor_security_mode      ]
[ , [ @distributor_login = ]      distributor_login      ]
[ , [ @distributor_password = ]      distributor_password      ]
[ , [ @encrypted_password= ]      encrypted_password]
[ , [ @frequency_type = ]      frequency_type      ]
[ , [ @frequency_interval = ]      frequency_interval      ]
[ , [ @frequency_relative_interval = ]      frequency_relative_interval
]
```

```

[ , [ @frequency_recurrence_factor = ]      frequency_recurrence_factor
]
[ , [ @frequency_subday = ]      frequency_subday      ]
[ , [ @frequency_subday_interval = ]      frequency_subday_interval      ]
[ , [ @active_start_time_of_day = ]      active_start_time_of_day      ]
[ , [ @active_end_time_of_day = ]      active_end_time_of_day      ]
[ , [ @active_start_date = ]      active_start_date      ]
[ , [ @active_end_date = ]      active_end_date      ]
[ , [ @optional_command_line = ]      optional_command_line      ]
[ , [ @merge_jobid = ]      merge_jobid      ]
[ , [ @agent_mode = ]      agent_mode      ]

```

参数：

[ @name = ]	name	name 是代理的系统名称，缺省值为 NULL。
[ @publisher = ]	publisher	publisher 是 Publisher 的系统名(服务器名)。
[ @publisher_db = ]	publisher_db	publisher_db 是 Publisher 数据库的系统名。
[ @publication = ]	publication	publication 是出版物的系统名称。
[ @publisher_security_mode = ]	publisher_security_mode	publisher_security_mode 是一个整数，表示 Publisher 中的安全模式。

```
[ @publisher_login = ]
publisher_login
[ @publisher_password = ]
publisher_password

[ @publisher_encrypted_password=
] publisher_encrypted_password

[ @subscriber = ] subscriber
[ @subscriber_db = ]
subscriber_db
[ @subscriber_security_mode = ]
subscriber_security_mode

[ subscriber_login = ]
subscriber_login

[ @subscriber_password = ]
subscriber_password
```

缺省值为 1。

publisher\_login 是系统名称，缺省值为 NULL。

publisher\_password 是一个系统名称，缺省值是 NULL。

publisher\_encrypted\_password 是位数据类型，缺省值为 0。

subscriber 是订阅者的系统名称。

subscriber\_db 是订阅者数据库的系统名称。

subscriber\_security\_mode 是一个整数，代表 Subscriber 的安全模式，缺省值为 1。其中 0 = SQLServer Authentication，1 = Windows NTAuthentication

subscriber\_login 用于 SQL Server Authentication(订阅者安全模式)，缺省值为 NULL

subscriber\_password 是系统名称，用于 SQL Server Authentication (订

[ @distributor = ] distributor

[ @distributor\_security\_mode = ]  
distributor\_security\_mode

[ @distributor\_login = ]  
distributor\_login

[ @distributor\_password = ]  
distributor\_password

[ @encrypted\_password = ]  
encrypted\_password

[ @frequency\_type = ]  
frequency\_type

阅者安全模式为 0)，缺省值为 NULL。  
distributor 是 Distributor 的系统名，缺省值为 Publisher 服务器。

distributor\_security\_mode 是一个整数，代表 distributor 的安全模式，缺省值为 1。其中，0 = SQL Server Authentication，1 = Windows NT Authentication

distributor\_login 用于 SQL Server Authentication(订阅者安全模式)，缺省值为 sa。

distributor\_password 是用于 SQL Server Authentication(订阅者安全模式)的系统名称，缺省值 NULL。

encrypted\_password 是位数据类型，在编辑脚本时使用，以确定是否加密口令，缺省值为 0。

frequency\_type 是一个整数，决定如何安排 Distribution Agent 的时间表。其中 1=一次；2=根据需要；4=每天；8=每周；16=每月；32=相关的月份；64=自动启动；124=重复；NULL=



```
[ @frequency_interval = ]  
frequency_interval
```

```
[ @frequency_relative_interval  
= ] frequency_relative_interval
```

```
[ @frequency_recurrence_factor=  
] frequency_recurrence_factor  
[ @frequency_subday = ]  
frequency_subday
```

```
[ @frequency_subday_interval = ]  
frequency_subday_interval
```

```
[ @active_start_time_of_day= ]
```

缺省值。

frequency\_interval 是一个代表 Distributor Agent 运行日期的整数。1=周日；2=周一；3=周二；4=周三；5=周四；6=周五；7=周六；8=白天；9=平日；10=周末白天；NULL=缺省值。

frequency\_relative 是一个整数，当 frequency\_type 变量为相关的月份时使用。1=第一；2=第二；4=第三；8=第四；16=最后一个；NULL=缺省值。

frequency\_recurrence\_factor 是一个整数，与 frequency\_type 变量一起使用。缺省值为 NULL。

frequency\_subday 是一个整数，代表在选定的时间段内何时再安排时间表。其中，1=立刻；2=秒，4=分钟，8=小时；NULL=缺省值。

frequency\_subday\_interval 是一个整数，表示 frequency\_subday 的时间间隔。缺省值为 NULL。

active\_start\_time\_of\_day 是一个整

active_start_time_of_day	数，代表首次安排 Distribution Agent 的时刻。缺省值为 NULL(格式为 HHMMSS)。
[ @active_end_time_of_day = ] active_end_time_of_day	active_end_time_of_day 是一个整数，代表停止安排 Distribution Agent 的时刻，缺省值为 NULL(格式为 HHMMSS)。
[ @active_start_date = ] active_start_date	active_start_date 是一个整数，代表首次安排 Distribution Agent 的日期。缺省值为 NULL(格式为 YYYYMMDD)。
[ @active_end_date = ] active_end_date	active_end_date 是一个整数，代表停止安排 Distribution Agent 的日期。缺省值为 NULL(格式为 YYYYMMDD)。
[ @optional_command_line = ] optional_command_line	optional_command_line 是提供给 Distribution Agent 的 nvarchar(255) 型命令。缺省值为 NULL。
[ @merge_jobid = ] merge_jobid	merge_jobid 是二进制 (16) 型数据，是一个输出参数，缺省值为 NULL。
[ @agent_mode = ] agent_mode	agent_mode 是 nvarchar(200) 型数

据，是创建代理的模式。它可以是下列值之一：none = 没有代理，sqlagent = SQL Server Agent，mobilesync = 可移动的同步代理，both = SQL Server Agent 和可移动的同步代理。

## sp\_addmergesubscription

sp\_addmergesubscription 系统存储过程增加一个推出或推入合并订阅。

### 语法

```
sp_addmergesubscription { [ @publication = ]      publication      }
[ , [ @subscriber = ]      subscriber      ][ , [ @subscriber_db = ]
    subscriber_db      ]
[ , [ @subscription_type = ]      subscription_type      ]
[ , [ @subscriber_type = ]      subscriber_type      ]
[ [ @subscription_priority = ]      subscription-priority      ]
[ , [ @sync_type = ]      sync_type      ]
[ , [ @frequency_type = ]      frequency_type      ]
[ , [ @frequency_interval = ]      frequency_interval      ]
[ , [ @frequency_relative_interval = ]      frequency_relative_interval
    ]
[ , [ @frequency_recurrence_factor = ]      frequency_recurrence_factor
```

```

]
[ , [ @frequency_subday = ]      frequency_subday ]
[ , [ @frequency_subday_interval= ]      frequency_subday_interval ]
[ , [ @active_start_time_of_day = ]      active_start_time_of_day ]
[ , [ @active_end_time_of_day = ]      active_end_time_of_day ]
[ , [ @active_start_date = ]      active_start_date ]
[ , [ @active_end_date = ]      active_end_date ]
[ , [ @optional_command_line = ]      optional_command_line ]
[ , [ @description = ]      description ]

```

### 变量

[ @publication = ] publication publication 是出版物的系统名。

[ @subscriber = ] subscriber subscriber 是 Subscriber 的系统名称 (服务器名称)。

[ @subscriber\_db = ]  
subscriber\_db subscriber\_db 是 Subscriber 数据库的系统名称。

[ @subscriber\_type = ]  
subscriber\_type subscriber\_type 是 nvarchar(15)型数据，缺省值为 local，它可以是 global, local 或 anonymous。

[ @subscription\_type = ]  
subscription\_type subscription\_type 是订阅的 nvarchar(15) 型种类，缺省值为 push。其中 Push = 推出订阅和

[ @subscription\_priority = ]  
subscription\_priority

[ @sync\_type = ] sync\_type

[ @frequency\_type = ]  
frequency\_type

[ @frequency\_interval = ]  
frequency\_interval

[ @C = ]  
frequency\_relative\_interval

Distributor 上的 Merge Agent。Pull = 推入订阅，没有 Distributor 上的 Merge Agent。

subscription\_priority 是 real 型数据，用以解决冲突。缺省值为 NULL。Local = 0.0，Anonymous = 0.0。

sync\_type 是 nvarchar(15) 型订阅同步类型，缺省值为自动。Automatic = 模式和数据转换，Nosync = 无模式和数据转换

frequency\_type 是决定 Merge Agent 何时运行的整数。1=一次；2=根据需要；4=每天，8=每周；16=每月；32=相关的月份，64=自动启动；120=重复；NULL=缺省值。

frequency\_interval 是一个整数，代表 Merge Agent 的运行日期。1=周日，2=周一，3=周二，4=周三，5=周四，6=周五，7=周六，8=白天，9=平日，10=周末白天，NULL=缺省值。

frequency\_relative\_interval 是一个整数，当 frequency\_type 变量为

```
[ @frequency_recurrence_factor = ]
frequency_recurrence_factor
    [ @frequency_subday = ]
frequency_subday
```

```
[ @frequency_subday_interval = ]
frequency_subday_interval
```

```
[ @active_start_time_of_day = ]
active_start_time_of_day
```

```
[ @active_end_time_of_day = ]
active_end_time_of_day
```

```
[ @active_start_date = ]
active_start_date
```

相关的月份时使用，1=第一，2=第二，4=第三，8=第四，16=最后一个，NULL=缺省值。

frequency\_recurrence\_factor 是一个整数，与 frequency\_type 一起使用，缺省值为 NULL。

frequency\_subday 是一个整数，代表在选定的时间段内何时再安排时间表。其中，1=立刻；2=秒，4=分钟，8=小时；NULL=缺省值。

frequency\_subday\_interval 是一个整数，表示每次合并之间所发生的 frequency\_subday 的时间间隔。缺省值为 NULL。

active\_start\_time\_of\_day 是一个整数，代表首次安排 Merge Agent 的时刻。缺省值为 NULL (格式为 HHMMSS)。

active\_end\_time\_of\_day 是一个整数，代表停止安排 Merge Agent 的时刻。缺省值为 NULL (格式为 HHMMSS)。

active\_start\_date 是一个整数，代表首次安排 Merge Agent 的日期。缺

<pre>[ @active_end_date = ] active_end_date</pre>	<p>省值为 NULL (格式为 YYYYMMDD)。 active_end_date 是一个整数，代表停止安排 Merge Agent 的日期。缺省值为 NULL (格式为 YYYYMMDD)。</p>
<pre>[ @optional_command_line = ] optional_command_line</pre>	<p>optional_command_line 是 nvarchar(255)型命令 (输出结果是文件，配置文件或属性)。缺省值为 NULL。</p>
<pre>[ @description = ] description</pre>	<p>description 是 nvarchar(255)型合并订阅描述。缺省为 NULL。</p>

## sp\_addmessage

sp\_addmessage 系统存储过程可在 sysmessage 系统表中创建一个新的错误信息行。要以另一种语言添加信息，sysmessage 系统表中必须已有该信息的英文版。在以另一种语言添加相同的信息时，如果信息包含变量，这些变量必须与英文版信息中的变量具有相同的字符数，并在每个变量后带有感叹号 (!)。在以另一种语言添加信息之前，应使用 SET LANGUAGE 语句，否则系统将缺省使用当前连接的语言。使用 @@ 语言标识可查看当前连接语言。

### 语法

```
sp_addmessage {number, severity, msg } [ , language ] [ , with_log ]
```

[ , replace ]  
变量

number

number 是标识信息的整数。缺省值为 NULL。对于用户信息而不是系统信息而言，必须从 50001 开始。

severity

severity 是错误的严重性级别，其值为 smallint1 ~ 25。缺省值为 NULL。SA(系统管理员)必须添加严重性级别 19 至 25 的信息。

msg

msg 是 nvarchar(255)型错误信息，缺省值为 NULL。

language

language 是要增加的信息所使用的语言，缺省值为 NULL。

with\_log

with\_log 是 nvarchar(5)型数据，其值为 true 或 false，它确定信息是否写入错误日志中(NT 和 SQL Server)。true 值表示将信息写入日志；false 值表示信息也可以写入日志，但信息是否写入日志取决于引发信息的方式。

replace

replace 是用作 varchar(7)型的标



志，表示错误信息已存在，并取代错误信息和严重级别。其中，NULL=缺省值，Replace=取代错误信息和严重级别。

## sp\_add\_notification

sp\_add\_notification 系统存储过程可添加警告通知。

### 语法

```
sp_add_notification { alert } [ , operator ]  
[ , notification_method ]
```

### 变量

alert

是对用户的选择的 varchar(100) 通知警告描述。

operator

operator [ varchar(100) ] 是警告发生被通知的操作员。

notification\_method

notification\_method 是 tinyint 型数据，表示警告发生时通知操作员的方式。1=电子邮件(缺省值，必须已配置了 SQL Mail)。2=寻呼；3=NetSend，7=All(所有方式)。

## sp\_add\_operator

sp\_add\_operator 系统存储过程必须在 msdb 系统数据库中运行。它可增加警告发生时要通知的操作员。如果要向某个寻呼机发送通知，则电子邮件系统必须具备电子邮件到寻呼机的功能。

### 语法

```
_address ]  
[ , weekday_pager_start_time ][ , weekday_pager_end_time ]  
[ , saturday_pager_start_time ][ , saturday_pager_end_time ]  
[ , sunday_pager_start_time ][ , sunday_pager_end_time ][ , pager_days ]  
[ , netsend_address ][ , category ]
```

### 变量

name	name 是在发生警告时要通知的操作员的 varchar(100)名。
enabled	enabled 是 tinyint 型数据，代表操作员的 状态。1=能接收通知(缺省值)，0=不能接收通知。
email_address	email_address 是操作员的 varchar(100)型 e-mail 地址。
pager_address	pager_address 是操作员的 varchar(100)型

	寻呼地址，缺省值为 NULL。
<code>weekday_pager_start_time</code>	<code>weekday_pager_start_time</code> 是一个格式为 HHMMSS 的整数，缺省值为 090000，即 24 小时制的 9:00AM，代表周一至周五开始向操作员发送寻呼通知的时间。
<code>weekday_pager_end_time</code>	<code>weekday_pager_end_time</code> 是一个格式为 HHMMSS 的整数，缺省值为 180000，即 24 小时制的 6:00PM，代表周一至周五停止向操作员发送寻呼通知的时间。
<code>saturday_pager_start_time</code>	<code>saturday_pager_start_time</code> 是一个格式为 HHMMSS 的整数，缺省值为 090000，即 24 小时制的 9:00AM，代表周六开始向操作员发送寻呼通知的时间。
<code>saturday_pager_end_time</code>	<code>saturday_pager_end_time</code> 是一个格式为 HHMMSS 的整数，缺省值为 180000，即 24 小时制的 6:00PM，代表周六停止向操作员发送寻呼通知的时间。
<code>sunday_pager_start_time</code>	<code>sunday_pager_start_time</code> 是一个格式为 HHMMSS 的整数，缺省值为 090000，即 24 小时制的 9:00AM，代表周日开始向操作员发送寻呼通知的时间。
<code>sunday_pager_end_time</code>	<code>sunday_pager_end_time</code> 是一个格式为

HHMMSS 的整数，缺省值为 180000，即 24 小时制的 6:00PM，代表周日停止向操作员发送寻呼通知的时间。

pager\_days

pager\_days 是 tinyint 型数据，其值为 1~127。表示可以寻呼操作员的日期，缺省值 0=不能。把每天的值加在一起：1=周日，2=周一，4=周二，8=周三，16=周四，32=周五，64=周六。例如：周一至周五将是 2 + 4 + 8 + 16 + 32=62，或周六=64。

net\_send\_address

net\_send\_address 是通过计算机屏幕上弹出的网络信息通知的操作员的 varchar(100) 型 e-mail 地址，缺省值为 NULL。

category

category 是 varchar(100) 型类别名。缺省值为 NULL。

## sp\_addpublication

sp\_addpublication 系统存储过程可增加一个出版物，用于快照、事务性和合并复制，它只向后兼容。此版本中用 sp\_addpublication70 取代了 sp\_addpublication。

### 语法

```
sp_addpublication { [ @publication = ] publication } [ , [ @taskid = ]
```

```

    taskid      ]
[ , [ @restricted = ]      restricted      ] [ , [ @sync_method = ]
sync_method    ]
[ , [ @repl_freq = ]      repl_freq      ] [ , [ @description = ]      description
    ]
[ , [ @status = ]      status      ] [ , [ @independent_agent = ]
independent_agent    ]
[ , [ @immediate_sync = ]      immediate_sync      ]
[ , [ @enabled_for_internet = ]      enabled_for_internet      ]
[ , [ @allow_push = ]      allow_push, [ , [ @allow_pull = ]      allow_pull
    ]
[ , [ @allow_anonymous = ]      allow_anonymous      ]
[ , [ @allow_sync_tran = ]      allow_sync_tran      ]
[ , [ @autogen_sync_procs = ]      autogen_sync_procs      ]
[ , [ @retention = ]      retention      ]
[ , [ @default_access = ]      default_access

```

### 变量

[ @publication = ] publication publisher 是出版物的系统名称。

[ @taskid = ] taskid taskid 是一个整数，代表预先创建的调度任务。该变量只向后兼容。使用 sp\_addpublication\_snapshot 系统存

[ @restricted = ] restricted

储过程。

restricted 是 nvarchar(10)型数据，其值为 true 或 false，指定谁可以查看出版物，其中 False=任何人均可观看，True=只有已授权的 Subscribers 能查看。

[ @sync\_method = ] sync\_method

sync\_method 是 nvarchar(13)型的同步方式，缺省值为 native。其中，Native=成批拷贝，Character=字符成批拷贝。

[ @repl\_freq = ] repl\_freq

repl\_freq 是 nvarchar(10)型的复制频率类型。其中 Continuous=出版者发送所有基于日志的事务，Snapshot=出版者发送安排好的同步事件。

[ @description = ] description

description 是 nvarchar(255)型的出版物描述。缺省值为 NULL。

[ @status = ] status

status 是 nvarchar(8)型数据，它确定出版物首次创建时，订阅者能否使用出版物数据。缺省值为未激活。Inactive = 不能立即使用出版物数据，Active = 可以立即使用出版物数据。

[ @independent\_agent = ]  
independent\_agent

[ @immediate\_sync = ]  
immediate\_sync

[ @enabled\_for\_internet = ]  
enabled\_for\_internet

[ @allow\_push = ] allow\_push

independent\_agent 是 nvarchar(5) 型数据，其值为 true 或 false。True 代表独立的 Distributor Agent。

immediate\_sync 代理是 nvarchar(5) 数据，其值为 true 或 false。True = 在创建订阅且独立的 Distribution Agent 已存在后，创建同步文件，运行 Snapshot Agent，且 Subscribers 立即拥有同步文件。False = 只有在有新订阅的情况下，才创建同步文件。创建订阅后，只有当 Snapshot Agent 运行成功后，Subscriber 才拥有同步文件 (缺省值)。

enabled\_for\_internet 是 nvarchar(5) 型数据，其值为 true 或 false。其中 False = 缺省值，True = 出版物的同步文件放在以前手工创建的 \repldate\ftp 文件夹中。

allow\_push 是 nvarchar(5) 型数据，其值为 true 或 false。其中，True = 出版物允许推出订阅 (缺省)。False = 不允许。

[ @allow\_pull = ] allow\_pull

[ @allow\_anonymous = ]  
allow\_anonymous

[ @allow\_sync\_tran = ]  
allow\_sync\_tran

[ @autogen\_sync\_procs = ]  
autogen\_sync\_procs

[ @retention = ] retention

[ , [ @default\_access = ]  
default\_access

allow\_pull 是 nvarchar(5)型数据，其值为 true 或 false。其中，True=出版物允许推入订阅(缺省)。False=不允许。

allow\_anonymous 是 nvarchar(5)型数据，其值为 true 或 false。其中，True=如果 immediate\_sync 为 true，则允许匿名；False=缺省值。

allow\_sync\_tran 是 nvarchar(5)型数据，其值为 true 或 false。false 为缺省。

autogen\_sync\_procs 是 nvarchar(5)型数据，其值为 true 或 false，缺省为 true。

retention 是一个代表保留期间的整数。缺省值为 1。

default\_access 是 nvarchar(5)型数据，其值为 true 或 false。它确定是否使用 Publisher 的缺省出版物访问列表。缺省值为 NULL。



## sp\_addpublication\_snapshot

sp\_addpublication\_snapshot 系统存储过程可增加一个 Snapshot Agent。

### 语法

```
sp_addpublication_snapshot { [ @publication = ]      publication  }
[ , [ @frequency_type = ]      frequency_type  ]
[ , [ @frequency_interval = ]    frequency_interval  ]
[ , [ @frequency_subday = ]      frequency_subday  ]
[ , [ @frequency_subday_interval = ]    frequency_subday_interval  ]
[ , [ @frequency_relative_interval = ]    frequency_relative_interval
]
[ , [ @frequency_recurrence_factor = ]    frequency_recurrence_factor
]
[ , [ @active_start_date = ]    active_start_date  ]
[ , [ @active_end_date = ]      active_end_date  ]
[ , [ @active_start_time_of_day = ]    active_start_time_of_day  ]
[ , [ @active_end_time_of_day = ]    active_end_time_of_day  ]
```

### 参数：

[ @publication = ] publication publication 是出版物的系统名。

[ @frequency\_type = ] frequency\_type 是一个决定 Snapshot

frequency\_type

[ @frequency\_interval = ]  
frequency\_interval

[ @frequency\_subday = ]  
frequency\_subday

[ @frequency\_subday\_interval = ]  
frequency\_subday\_interval

[ @frequency\_relative\_interval = ]  
frequency\_relative\_interval

[ @frequency\_recurrence\_factor = ]  
frequency\_recurrence\_factor

[ @active\_start\_date = ]  
active\_start\_date

Agent 何时运行的整数。其中 1=立即；4=每天(缺省)，8=每周，10=每月，20=每月，与频率间隔有关，40=SQL Server Agent 启动时运行。

frequency\_interval 是一个整数，代表由 frequency\_type 设定的频率。缺省值为 1=每天。

frequency\_subday 是一个整数，代表变量 freq\_subday\_interval 的单位。其中，1=一次，2=秒，4=分钟(缺省)，8=小时。

frequency\_subday\_interval 是一个整数，代表 frequency\_subday 间隔，缺省值为 5=每分钟一次。

frequency\_relative\_interval 是一个整数，表示 Snapshot Agent 运行的日期，缺省为 1。

frequency\_recurrence\_factor 是一个整数，代表 frequency\_type 使用的重复因素，缺省值为 0。

active\_start\_date 是一个整数，代表安排 Snapshot Agent 首次运行

<pre>[ @active_end_date = ] active_end_date</pre>	<p>的日期。缺省值为 0(格式为 YYYYMMDD)。</p> <p>active_end_date 是一个整数，代表停止安排 Snapshot Agent 的日期。缺省值为 99991231(格式为 YYYYMMDD)。</p>
<pre>[ @active_start_time_of _day = ] active_start_time_of_day</pre>	<p>active_start_time_of_day 是一个整数，代表首次安排 Snapshot Agent 的时间。缺省值为 0(格式为 HHMMSS)。</p>
<pre>[ @active_end_time_of_day = ] active_end_time_of_day</pre>	<p>active_end_time_of_day 是一个整数，代表停止安排 Snapshot Agent 的时间。缺省值为 235959(格式为 HHMMSS)。</p>

## sp\_addpublisher70

sp\_addpublisher70 系统存储过程在 6.5 版的 Subscriber 的基础上增加了一个 7.0 版的 Publisher。用户必须在增添了 sp\_addpublisher70 存储过程的 6.5 版 SQL Server 中运行一个脚本。该脚本位于 \Mssql7 \ Install 目录下的 Replp 70.sql 文件中。

### 语法

```
sp_addpublisher70 { [ @publisher = ] publisher ,
```

[ @dist\_account = ] dist\_account }  
变量

[ @publisher = ] publisher publisher 是 Publisher 的  
varchar(3)型名称。

[ @dist\_account = ] dist\_account dist\_account 是 varchar(255)型 NT  
帐户。 Distribution Agent 在  
Distributor 中使用该帐户。它通常  
是 Distributor 中的 SQL Server  
Agent 帐户。

## sp\_addpullsubscription

sp\_addpullsubscription 系统存储过程可为 Subscriber 数据库增加一个推入或  
匿名订阅。

### 语法

```
sp_addpullsubscription { [ @publisher = ] publisher ,  
[ @publisher_db = ] publisher_db ] [ @publication = ] publication  
]  
[ , [ @independent_agent = ] independent_agent ]  
[ , [ @subscription_type = ] subscription_type ]  
[ , [ @description = ] description ] [ , @update_mode = ]
```

update_mode	]	
[ , [ @immediate_sync = ]	immediate_sync ]	
变量		
[ @publisher = ]	publisher	publisher 是 Publisher 的系统名称。
[ @publisher_id = ]		publisher_db 是 Publisher 数据库的系统名称。
publisher_id		
[ @publication = ]	publication	publication 是出版物的系统名称。
[ @independent_agent = ]		independent_agent 是 nvarchar(5) 型数据，其值为 true 或 false，表示 Distributor 是独立的，缺省值为 true。
independent_agent		
[ @subscription_type = ]		subscription_type 是 nvarchar(9) 型数据，出版物的订阅类型可以是 pull 或匿名(缺省)。
subscription_type		
[ @description = ]	description	description 是 nvarchar(100) 型的出版物描述。缺省值为 NULL。
[ @update_mode = ]	update_mode	update_mode 是 nvarchar(15) 型数据，它可以是只读或同步。
[ , [ @immediate_sync = ]		immediate_sync 是位数据类型(缺省值为 1)，它确定是否在 Snapshot
immediate_sync ]		

Agent 运行时创建同步文件，该变量必须与 sp\_addpublication 中的 immediate\_sync 变量具有相同的值。

## sp\_addpullsubscription\_agent

sp\_addpullsubscription\_agent 系统存储过程可为 Subscriber 的数据库增加一个新代理，该存储过程可用于快照、事务性以及合并复制。

### 语法

```
sp_addpullsubscription_agent { [ @publisher = ] publisher ,  
[ @publisher_db = ] publisher_db , [ @publication = ]  
publication } ,  
[ , [ @subscriber = ] subscriber ]  
[ , [ @subscriber_db = ] subscriber_db ]  
[ , [ @subscriber_security_mode = ] subscriber_security_mode ]  
[ , [ @subscriber_login = ] subscriber_login ]  
[ , [ @subscriber_password = ] subscriber_password ]  
[ , [ @distributor = ] distributor , [ , [ @distribution_db = ]  
distribution_db ]  
[ , [ @distributor_security_mode = ] distributor_security_mode ]  
[ , [ @distributor_login = ] distributor_login ]  
[ , [ @distributor_password = ] distributor_password ]
```

```

[ , [ @optional_command_line = ] optional_command_line ]
[ , [ @frequency_type = ] frequency_type ]
[ , [ @frequency_interval = ] frequency_interval ]
[ , [ @frequency_relative_interval = ] frequency_relative_interval ]
[ , [ @frequency_recurrence_factor = ] frequency_recurrence_factor ]
[ , [ @frequency_subday = ] frequency_subday ]
[ , [ @frequency_subday_interval = ] frequency_subday_interval ]
[ , [ @active_start_time_of_day = ] active_start_time_of_day ]
[ , [ @active_end_time_of_day = ] active_end_time_of_day ]
[ , [ @active_start_date = ] active_start_date ]
[ , [ @active_end_date = ] active_end_date ]
[ , [ @distribution_jobid = ] distribution_jobid ]
[ , [ @encrypted_distributor_password = ]
encrypted_distributor_password ]
[ , [ @agent_mode = ] agent_mode ]

```

## 变量

[ @publisher = ] publisher	publisher 是 Publisher 的系统名称 (服务器名称)。
[ @publisher_db = ] publisher_db	publisher_db 是 Publisher 数据库的系统名称。
[ @publication = ] publication	publication 是出版物的系统名称。

[ , [ @subscriber = ] subscriber	subscriber 是 Subscriber 数据库的系统名称。
[ , [ subscriber_db = ] subscriber_db	subscriber_db 是订阅数据库的系统名称。
[ @subscriber_security_mode = ] subscriber_security_mode	subscriber_security_mode 是一个代表 Subscriber 安全模式的整数。缺省值为 1。其中，0=SQL Server Authentication，1=Windows NT Authentication
[ @subscriber_login = ] subscriber_login	subscriber_login 用于 SQL Server Authentication(订阅者安全模式)，缺省值为 NULL。
[ @subscriber_password = ] subscriber_password	subscriber_password 是用于 SQL Server Authentication (订阅者安全模式)的系统名称，缺省值为 NULL。
[ @distributor = ] distributor	是 distributor 的系统名，缺省为 Publisher(服务器名)。
[ @distributor_security_mode = ] distributor_security_mode	distributor_security_mode 是一个代表 Distributor 安全模式的整数。缺省值为 1。其中，0 = SQL Server Authentication，1 = Windows NT



```

    [ @distributor_login = ]
distributor_login

[ @distributor_password = ]
distributor_password

    [ @optional_command_line = ]
option command_line

[ @frequency_type = ]
frequency_type

[ @frequency_interval = ]
frequency_interval

```

Authentication。

distributor\_login 用于 SQL Server Authentication (订阅者安全模式), 缺省值为 sa。

distributor\_password 是用于 SQL Server Authentication(订阅者安全模式)的系统名称, 缺省值为 NULL。

optional\_command\_line 是提供给 Distribution Agent 的 nvarchar(255) 型命令。缺省值为 NULL。

frequency\_type 是一个决定如何安排 Distribution Agent 的整数, 1=一次, 2=根据需要, 4=每天, 8=每周, 16=每月, 32=相关的月份, 64=自动启动, 124=重复, NULL=缺省值。

frequency\_interval 是一个代表 Distribution Agent 运行日期的整数。其中 1=周日, 2=周一, 3=周二, 4=周三, 5=周四, 6=周五, 7=周六, 8=白天, 9=平日, 10=周末白天, NULL=

[ @C = ] frequency\_relative\_interval

[ @frequency\_recurrence\_factor = ] frequency\_recurrence\_factor

[ @frequency\_subday = ] frequency\_subday

[ @frequency\_subday\_interval = ] frequency\_subday\_interval

[ @active\_start\_time\_of\_day = ] active\_start\_time\_of\_day

[ @active\_end\_time\_of\_day = ] active\_end\_time\_of\_day

缺省值。

frequency\_relative\_interval 是一个整数，在 frequency\_type 为相关的月份时使用。其中，1=第一，2=第二，3=第三，8=第四，16=最末，NULL=缺省。

frequency\_recurrence\_factor 是一个整数，与 frequency\_type 一起使用，缺省值为 NULL。

frequency\_subday 是一个整数，代表在选定的时间段内，何时再安排运行。其中 1=一次，2=秒，4=分钟，8=小时，NULL=缺省值。

frequency\_subday\_interval 是一个整数，表示 frequency\_subday 的间隔。缺省值为 NULL。

active\_start\_time\_of\_day 是一个整数，代表安排 Distribution Agent 首次运行的时刻。缺省值为 NULL(格式为 HHMMSS)。

active\_end\_time\_of\_day 是一个整数，代表停止安排 Distribution

[ @active\_start\_date = ]  
active\_start\_date

[ @active\_end\_date = ]  
active\_end\_date

[ @distribution\_jobid = ]  
distribution\_jobid

[ @encrypted\_distributor\_password = ]  
encrypted\_distributor\_password  
[ , [ @agent\_mode = ] agent\_mode

Agent 运行的时刻。缺省值为 NULL(格式为 HHMMSS)。

active\_start\_date 是一个整数，代表首次安排 Distribution Agent 的日期。缺省值为 NULL(格式为 YYYYMMDD)。

active\_end\_date 是一个整数，代表停止安排 Distribution Agent 的日期。缺省值为 NULL(格式为 YYYYMMDD)。

distribution\_jobid 是二进制(16)型数据，是一个输出参数，缺省值为 NULL。

encrypted\_distributor\_password 是一个位型数据，在编辑脚本时使用，它确定是否加密发布器密码，缺省值为 0。

agent\_mode 是 nvarchar(20)型数据，代表创建代理时的模式。它可以是以下值之一：none = 没有代理，sqlagent = SQL Server Agent，mobilesync = 可移动的同步代理，both = SQL Server

Agent 和可移动的同步代理。

## sp\_addremotelogin

sp\_addremotelogin 系统存储过程可在本地服务器上添加一个新的远程登录。

### 语法

```
sp_addremotelogin [ @remoteserver = ]      remoteserver      [ , [ @loginame  
= ]  
login      [ , [ @remotename = ]      remote_name      ]]
```

### 变量

[ @remoteserver = ]  
remoteserver

remoteserver 是远程服务器的系统名称。只有提供了其他变量，远程服务器上的用户才与本地服务器上的用户具有相同的登录。

[ @loginame = ] login

login 是本地 SQL Server 中用户的登录系统名。

[ @remotename = ] remote\_name

remote 是远程服务器上用户的登录系统名。

## sp\_addrole

sp\_addrole 系统存储过程在使用的数据库中增加一个数据库角色。

### 语法

```
sp_addrole [ @rolename = ]          role          [ , [ @ownername = ]          owner  
]
```

### 变量

```
[ @rolename = ]    role  
[ @ownername = ]  owner
```

role 是新角色的系统名。  
owner 是角色拥有者的系统名称，缺省为 dbo(数据库拥有者)。

## sp\_addrolemember

sp\_addrolemember 系统存储过程给使用的数据库中的数据库角色增加一个安全帐户。

### 语法

```
sp_addrolemember [ @rolename = ]          role          , [ @membername = ]  
security_account
```

### 变量

```
[ @rolename = ]    role          rolename 是数据库角色的系统名称。
```

```
[ @membername = ]
security_account
```

security\_account 是增加到数据库角色中的安全帐户的系统名。如果使用中的数据库没有该帐户，就添加该帐户。

## sp\_addserver

sp\_addserver 系统存储过程为一个远程服务器在 sys.servers 系统表中增加一行。它用于向后兼容。在 7.0 版本中，用系统存储过程 sp\_addlinkedserver 取代 sp\_addserver。

### 语法

```
sp_addserver [ @server = ] server [ , [ @local = ] local ]
[ , [ @duplicate_ok = ] duplicate_OK ]
```

### 变量

[ @server = ] server

[ @local = ] LOCAL

[ @duplicate\_ok = ]

duplicate\_ok

duplicate\_OK

server 是服务器的系统名称。

LOCAL 是 varchar(10)型数据，表明服务器的系统名是本地服务器，缺省值为 NULL。

duplicate\_ok 是 varchar(13)型数据，其值可以是 duplicate\_OK 或 NULL。

表示在无错误报告的情况下，可存在副本服务器名。

## sp\_addsrvrolemember

sp\_addsrvrolemember 系统存储过程为固定服务器角色添加一个注册。

### 语法

```
sp_addsrvrolemember [ @loginame = ] login [ , [ @rolename = ]  
role ]
```

### 变量

[ @loginame = ] login

login 是要添加到固定服务器角色中的注册系统名称。如果 SQL Server 中不存在该注册，则添加它。

[ @rolename = ] role

role 是固定服务器角色，sysadmin，securityadmin，serveradmin，setupadmin，processadmin，diskadmin 或 dbcreator 的系统名称。

## sp\_addsubscriber

sp\_addsubscriber 系统存储过程可创建新 Subscriber 以用于快照、事务性以及合并复制。

## 语法

```
sp_addsubscriber { [ @subscriber = ] subscriber } [ , [ @type = ]
type ]
[ , [ @login = ] login ] [ , [ @password = ] password ]
[ , [ @commit_batch_size = ] commit_batch_size ]
[ , [ @status_batch_size = ] status_batch_size ]
[ , [ @flush_frequency = ] flush_frequency ]
[ , [ @frequency_type = ] frequency_type ]
[ , [ @frequency_interval = ] frequency_interval ]
[ , [ @frequency_relative_interval = ] frequency_relative_interval
]
[ , [ @frequency_recurrence_factor = ] frequency_recurrence_factor
]
[ , [ @frequency_subday = ] frequency_subday ]
[ , [ @frequency_subday_interval = ] frequency_subday_interval ]
[ , [ @active_start_time_of_day = ] active_start_time_of_day ]
[ , [ @active_end_time_of_day = ] active_end_time_of_day ]
[ , [ @active_start_date = ] active_start_date ]
[ , [ @active_end_date = ] active_end_date ]
[ , [ @description = ] description ] [ , [ @security_mode = ]
security_mode ]
[ , [ @encrypted_password = ] encrypted_password ]
```



## 变量

[ @subscriber = ] subscriber

subscriber 是添加为 Subscriber 的服务器的系统名称。

[ @type = ] type

type 是 Subscriber 的 tinyint 类型。其中 0=Microsoft SQL Server Subscriber, 1=ODBC 数据源服务器。

[ @login = ] login

login 是 SQL Server 授权的登录的系统名称, 缺省值为 SA。

[ @password = ] password

password 是 SQL Server 授权的登录的系统名口令, 缺省值为 NULL。

[ @commit\_batch\_size = ]  
commit\_batch\_size

commit\_batch\_size 是一个整数, 它决定将多少个事务发送到 Subscriber 中, 并一起运作, 缺省值为 100。

[ @flush\_frequency = ]  
flush\_frequency

flush\_frequency 是一个整数, 决定事务在向 Subscriber 发布后, 从分布式数据库表中删除的频率。缺省为 0。

[ @frequency\_type = ]  
frequency\_type

Frequency\_type 是一个整数, 决定如何安排 Distribution Agent 的时间表。其中 1=一次, 2=根据需要, 4=每天, 8=每周, 16=每月, 32=相关的月份, 64=自动启动, 124=重复, NULL=

```
[ @frequency_interval = ]  
frequency_interval
```

```
[ @frequency_relative_interval = ]  
frequency_relative_interval
```

```
[ @frequency_recurrence_factor = ]  
frequency_recurrence_factor
```

```
[ @frequency_subday = ]  
frequency_subday
```

```
[ @frequency_subday_interval = ]  
frequency_subday_interval
```

缺省值。

frequency\_interval 是一个整数，代表 Distribution Agent 的运行日期。其中 1=周日，2=周一，3=周二，4=周三，5=周四，6=周五，7=周六，8=白天，9=平日，10=周末白天，NULL=缺省值。

frequency\_relative\_interval 是一个整数，当变量 frequency\_type 为相关的月份时使用。其中 1=第一，2=第二，4=第三，8=第四，16=最后一个，NULL=缺省值。

frequency\_recurrence\_factor 是一个整数，与 frequency\_type 一起使用，缺省值为 NULL。

frequency\_subday 是一个整数，代表在选定的时间段内，何时再次进行安排。1=一次，2=秒，4=分钟，8=小时，NULL=缺省值。

frequency\_subday\_interval 是一个整数，代表 frequency\_subday 的间隔。缺省值为 NULL。

[ @active_start_time_of_day = ]	active_start_time_of_day	active_start_time_of_day 是一个整数，代表首次安排 Distribution Agent 的时刻。缺省值为 NULL (格式为 HHMMSS)。
active_start_time_of_day		
[ @active_end_time_of_day = ]	active_end_time_of_day	active_end_time_of_day 是一个整数，代表停止安排 Distribution Agent 的时刻。缺省值为 NULL (格式为 HHMMSS)。
active_end_time_of_day		
[ @active_start_date = ]	active_start_date	active_start_date 是一个整数，代表首次安排 Distribution Agent 的日期。缺省值为 NULL (格式为 YYYYMMDD)。
active_start_date		
[ @active_end_date = ]	active_end_date	active_end_date 是一个整数，代表停止安排 Distribution Agent 的日期。缺省值为 NULL (格式为 YYYYMMDD)。
active_end_date		
[ @description= ]	description	description 是 Subscriber 的 nvarchar(255)型描述。缺省为 NULL。
[ @security_mode = ]	security_mode	security_mode 是一个代表安全模式的整数。0=SQL Server Authentication; 1=Windows NT
security_mode		

```
[ @encrypted_password = ]  
encrypted_password
```

Authentication(缺省)。

encrypted\_password 是位型数据，在编辑脚本时使用，它确定是否加密口令，缺省值为 0。

## sp\_addsubscriber\_schedule

sp\_addsubscriber\_schedule 系统存储过程可增加一个 Distribution Agent 或 Merge Agent 时间表，用以快照、事务性以及合并复制。

### 语法

```
sp_addsubscriber_schedule { [ @subscriber = ] subscriber }  
[ , [ @agent_type = ] agent_type ] [ , [ @frequency_type = ]  
frequency_type ]  
[ , [ @frequency_interval = ] frequency_interval ]  
[ , [ @frequency_relative_interval = ] frequency_relative_interval  
]  
[ , [ @frequency_recurrence_factor = ] frequency_recurrence_factor  
]  
[ , [ @frequency_subday = ] frequency_subday ]  
[ , [ @frequency_subday_interval = ] frequency_subday_interval ]  
[ , [ @active_start_time_of_day = ] active_start_time_of_day ]  
[ , [ @active_end_time_of_day = ] active_end_time_of_day ]
```

```
[ , [ @active_start_date = ] active_start_date ]
[ , [ @active_end_date = ] active_end_date ]
变量
```

```
[ @subscriber = ] subscriber
[ @agent_type = ] agent_type
```

subscriber 是 Subscriber 的系统名称。  
agent\_type 是代理的 smallint 类型，其中 0=Distribution Agent，1=Merge Agent，缺省值为 0。

```
[ @frequency_type= ] frequency_type
```

frequency\_type 是一个确定如何安排 Distribution Agent 运行的整数。其中，1=一次，2=根据需要，4=每天，8=每周，16=每月，32=相关的月份，64=自动启动，124=重复，NULL=缺省值。

```
[ @frequency_interval = ]
frequency_interval
```

frequency\_interval 是一个整数，表示 Distribution Agent 的运行日期。其中，1=周日，2=周一，3=周二，4=周三，5=周四，6=周五，7=周六，8=白天，9=平日，10=周末白天，NULL=缺省值。

```
[ @frequency_relative_interval = ]
frequency_relative_interval
```

frequency\_relative\_interval 是一个整数，在 frequency\_type 为相关的月份时使用。其中 1=第一，2=第二，

	4=第三，8=第四，16=最末，NULL=缺省。
[ @frequency_recurrence_factor = ] frequency_recurrence_factor	frequency_recurrence_factor 是一个整数，与 frequency_type 一起使用，缺省值为 NULL。
[ @frequency_subday = ] frequency_subday	frequency_subday 是一个整数，代表在选定的时间段内，何时再安排运行。其中 1=一次，2=秒，4=分钟，8=小时，NULL=缺省值。
[ @frequency_subday_interval = ] frequency_subday_interval	frequency_subday_interval 是一个整数，表示 frequency_subday 的间隔。缺省值为 NULL。
[ @active_start_time_of_day = ] active_start_time_of_day	active_start_time_of_day 是一个整数，代表首次安排 Distribution Agent 运行的时刻，缺省值为 NULL(格式为 HHMMSS)。
[ @active_end_time_of_day = ] active_end_time_of_day	active_end_time_of_day 是一个整数，代表停止安排 Distribution Agent 运行的时刻。缺省值为 NULL(格式为 HHMMSS)。
[ @active_start_date = ] active_start_date	active_start_date 是一个整数，代表首次安排 Distribution Agent

<pre>[ @active_end_date = ] active_end_date</pre>	<p>运行的日期。缺省值为 NULL(格式为 YYYYMMDD)。</p> <p>active_end_date 是一个整数，代表停止安排 Distribution Agent 运行的日期。缺省值为 NULL(格式为 YYYYMMDD)。</p>
---	---

## sp\_addsubscription

sp\_addsubscription 系统存储过程可为文章增加一个订阅。如果 Subscriber 要在限制出版的文章中浏览文章，则虚拟订阅必须在未激活的状态下添加。

### 语法

```
sp_addsubscription { [ @publication = ]      publication }
[ , [ @article = ]    article    ] [ , [ @subscriber = ]      subscriber    ]
[ , [ @destination_db = ]      destination_db    , [ @sync_type = ]
sync_type    ]
[ , [ @status = ]      status    [ @subscription_type = ]
subscription_type    ]
[ , [ @update_mode = ]    update_mode    ]
[ , [ @loopback_detection = ]    loopback_detection    ]
[ , [ @frequency_type = ]    frequency_type    ]
[ , [ @frequency_interval = ]    frequency_interval    ]
```

[ , [ @frequency\_relative\_interval = ] frequency\_relative\_interval ]  
[ , [ @frequency\_recurrence\_factor = ] frequency\_recurrence\_factor ]  
[ , [ @frequency\_subday = ] frequency\_subday ]  
[ , [ @frequency\_subday\_interval = ] frequency\_subday\_interval ]  
[ , [ @active\_start\_time\_of\_day = ] active\_start\_time\_of\_day ]  
[ , [ @active\_end\_time\_of\_day = ] active\_end\_time\_of\_day ]  
[ , [ @active\_start\_date = ] active\_start\_date ]  
[ , [ @active\_end\_date = ] active\_end\_date ]  
[ , [ @optional\_command\_line = ] optional\_command\_line ]]

### 变量

[ @publication = ] publication publication 是出版物的系统名。

[ @article = ] article article 是文章的系统名称。如果没有提供该变量，则表示订阅出版物中的所有文章。

[ @subscriber = ] subscriber subscriber 是 Subscriber 的系统名称。缺省值为 NULL。

[ @destination\_db = ] destination\_db destination\_db 是存放复制数据的目标数据库的系统名称（应与 Publication 数据库同名）。



[ @sync_type = ] sync_type	sync_type 是 nchar(5)型 订阅同步类型。其中，Automatic=通过分布式过程在 Subscriber 上完成的同步文件(缺省)。Manual=同步文件自动生成，且手动运用到 Subscriber 上。None=无初始同步。
[ @status = ] status	status 是 订阅状态的系统名称。sync_type = 没有 且 对其他 Subscribed 而言 状态是激活的。sync_type=自动，且状态是未激活。
[ @subscription_type = ] subscription_type	subscription_type 是 nvarchar 型数据，是订阅的类型，其值为 push(缺省)或 pull。
[ @update_mode = ] update_mode	update_mode 是 nvarchar(15)型数据，是更新的类型。其中 read_only(只读)是缺省值，synctrans 是其他类型。
[ , [ @loopback_detection = ] loopback_detection ]	loopback_detection 是 nvarchar(5)型数据，决定 Distribution Agent 是否将在 Subscriber 上生成的事务传送给 Subscriber。其中 True=Distribution Agent 不将事务传送回来。False=Distribution Agent

```
[ @frequency_type = ]  
frequency_type
```

```
[ @frequency_interval = ]  
frequency_interval
```

```
[ @frequency_relative_interval  
= ] frequency_relative_interval
```

```
[ @frequency_recurrence_factor
```

将事务传送回来。NULL= 当 update\_mode 取 synctran 值，且文章具有已出版的时间戳列时，就将该变量设置为 true (缺省值)。

frequency\_type 是一个整数，决定如何安排 Distribution Agent。其中，1=一次，2=根据需要，4=每天，8=每周，16=每月，32=相关的月份，64=自动启动，124=重复，NULL=缺省值。

frequency\_interval 是一个整数，代表 Distribution Agent 的运行日期。其中，1=周日，2=周一，3=周二，4=周三，5=周四，6=周五，7=周六，8=白天，9=平日，10=周末白天，NULL=缺省值。

frequency\_relative\_interval 是一个整数，当变量 frequency\_type 取值为相关的月份时使用。其中 1=第一，2=第二，4=第三，8=第四，16=末尾，NULL=缺省值。

frequency\_recurrence\_factor 是一

<code>= ] frequency_recurrence_factor</code>	个整数，与 <code>frequency_type</code> 一起使用，缺省值为 NULL。
<code>[ @frequency_subday = ] frequency_subday</code>	<code>frequency_subday</code> 是一个整数，代表在选定的时间段内，何时再安排。其中 1=一次，2=秒，4=分钟，8=小时，NULL=缺省值。
<code>[ @frequency_subday_interval = ] frequency_subday_interval</code>	<code>frequency_subday_interval</code> 是一个整数，代表 <code>frequency_subday</code> 的间隔。缺省值为 NULL。
<code>[ @active_start_time_of_day = ] active_start_time_of_day</code>	<code>active_start_time_of_day</code> 是一个整数，代表首次安排 Distribution Agent 的时刻。缺省值为 NULL (格式为 HHMMSS)。
<code>[ @active_end_time_of_day = ] active_end_time_of_day</code>	<code>active_end_time_of_day</code> 是一个整数，代表停止安排 Distribution Agent 运行的时刻。缺省值为 NULL (格式为 HHMMSS)。
<code>[ @active_start_date = ] active_start_date</code>	<code>active_start_date</code> 是一个整数，代表首次安排 Distribution Agent 的日期。缺省值为 NULL (格式为 YYYYMMDD)。
<code>[ @active_end_date = ]</code>	<code>active_end_date</code> 是一个整数，代表

`active_end_date`

停止安排 Distribution Agent 运行的日期。缺省值为 NULL(格式为 YYYYMMDD)。

[ `@optional_command_line =` ]  
`optional_command_line`

`optional_command_line` 是要运行的 `nvarchar(4000)` 型命令行。缺省值为 NULL。

[ `@reserved =` ] `reserved`  
[ `@agent_mode =` ] `agent_mode`

`reserved` 缺省值为 NULL。  
`agent_mode` 是 `nvarchar(200)` 型数据，代表创建代理的方式。它可以取下列值之一：None = 没有代理，`sqlagent` = SQL Server Agent，`mobilesync` = 可移动的同步代理，`both` = SQL Server Agent 和可移动的同步代理。

## `sp_addsynctriggers`

`sp_addsynctriggers` 系统存储过程将在 Subscriber 中增加同步事务触发程序。

语法

```
sp_addsynctriggers { [ @source_table = ] source_table ,  
[ @publisher = ] publisher , [ @publisher_db = ] publisher_db
```

,

```
[ @ins_proc = ]      ins_proc ,      [ @upd_proc = ]      upd_proc
[ , [ @del_proc = ]  del_proc      } [ , [ @identity_col = ]  identity_col
]
[ , [ @ts_col = ]    ts_col        ] [ , [ @filter_clause = ]  filter_clause
]
变量
```

[ @source_table = ]	source_table	source_table 是源表的 nvarchar(92)型名称。
[ @publisher = ]	publisher	publisher 是 Publisher 的系统名称(服务器名称)。
[ @publisher_db = ]	publisher_db	publisher_db 是正发布的数据库的系统名称。缺省为使用中的数据库。
[ @ins_proc = ]	ins_proc	ins_proc 是用于同步事务插入的文章存储过程的系统名称。
[ @upd_proc = ]	upd_proc	upd_proc 是用于同步事务更新的文章存储过程的系统名称。
[ @del_proc = ]	del_proc	del_proc 是用于同步事务删除的文章存储过程的系统名。
[ @identity_col = ]	identity_col	identity_col 是标识列的系统名称。
[ ts_col = ]	ts_col	ts_col 是时间戳列的系统名称。
[ @filter_clause = ]	filter_clause	filter_clause 是 nvarchar(4000)型数

据，包含用于过滤源表的 WHERE 子句。

## sp\_addtask

sp\_addtask 系统存储过程仅用于向后兼容。Microsoft SQL Server 7.0 版的程序员可使用 sp\_add\_job。

## sp\_addtype

sp\_addtype 系统存储过程可为应用程序增加一个由用户创建的数据类型。  
语法

```
sp_addtype [ @typename = ] type ,  
[ @phystype = ] system_data_type  
[ , [ @nulltype = ] null_type ]
```

变量

type

type 是用户创建的数据类型的系统名称。

[ @phystype = ] system\_data\_type

作为用户创建数据类型的基础的 SQL Server 系统数据类型。该变量可取以下值：  
binary(n) , image ,

smalldatetime, bit, int smallint,  
char(n), nchar(n), text,  
datetime, ntext, tinyint, decimal,  
numeric, uniqueidentifier,  
decimal[(p[,s])], numeric  
[(p[,s])], varbinary(n),  
Float, nvarchar(n),  
varchar(n), float(n), real.

N, P, S 在本列表的以下行中描述。

N

N 代表一个非零正整数，表示用户创建的数据类型的长度。

P

P 代表一个非零正整数，表示小数点左边或右边的位数。

S

S 代表一个非零正整数，代表小数点右边的小数位数；它必须小于或等于 P（上一个变量）。

[, [ @nulltype = ] null\_type ]

null\_type 是 varchar(8)型数据，其值为 NULL 或 NOT NULL（包含在单引号内）。

## sp\_add\_targetservergroup

sp\_add\_targetservergroup 系统存储过程可添加目标服务器组的名称。

语法

```
sp_add_targetservergroup { name }
```

变量 name 是目标服务器组的 varchar(100)型名称。

## sp\_addumpdevice

sp\_addumpdevice 系统存储过程可添加一个用以备份数据库不同部分的清空设备。

语法

```
sp_addumpdevice { device_type , logical_name , physical_name }  
[ , {{controller_type} | device_status } ]
```

变量

device\_type

device\_type 是 varchar(10)型数据，代表清空设备的类型。其中，Disk=磁盘文件，pipe=命名的管道，tape=磁带设备。

logical\_name

logical\_name 是用于 BACKUP 和



physical\_name

RESTORE 的备份设备的系统名称。

physical\_name 是清空设备的 nvarchar(260)型物理名称，它是一个操作系统路径 \ 文件。磁带设备的一个例子是 \ \ . \ \ TAPE0。

controller\_type

controller\_type 变量在 7.0 版本中被忽略。但其值为 5=磁带，6=管道。

device\_status

device\_status 是 varchar(40)型数据，表示磁带标签，其中 noskip=读取(缺省)，skip=忽略。

## sp\_add\_targetsvrgrp\_member

sp\_add\_targetsvrgrp\_member 系统存储过程可为目标服务器组增加一个服务器。

### 语法

```
sp_add_targetsvrgrp_member { group_name } [ , server_name ]
```

### 变量

group\_name

group\_name 是 varchar(100)型数据，代表目标服务器组的名称。

server\_name

server\_name 是 varchar(30)型数据，

代表要添加到目标服务器组中的服务器名。

## sp\_adduser

sp\_adduser 用于向后兼容。7.0 版本的程序员可使用 sp\_grantdbaccess 来取代它。

### 语法

```
sp_adduser [ @loginame = ] login [ , [ @name_in_db = ] user  
[ , [ @rolename = ] role ]]
```

### 变量

[ @loginame = ] login

login 是登录的系统名称。

[ @name\_in\_db = ] user

user 是新用户的系统名称，缺省值为 login。

[ @rolename = ] role

role 代表新用户所在的角色。

## sp\_altermessage

sp\_altermessage 系统存储过程可更改信息是否写入错误日志的状态。

### 语法

sp\_altermessage {message\_number} [ , write\_to\_log ][ , value ]  
 变量

message_number	message_number 是 sysmessage 错误的整型 ID。
[ , write_to_log ][ , value ]	如果 write to log 的值为 WITH_LOG，就将信息写入 NT 错误日志。值为 true 时，表示将信息写入 NT 错误日志。值为 false 时，表明错误不自动写入错误日志(值为 false 时，错误仍可写入错误日志，但是否写入取决于错误产生的原因)。

### 范例

以下是 sp\_altermessage 系统存储过程的一个例子：

```
sp_altermessage 255101, WITH_LOG, true
```

在本例中，将信息 255101 写入错误日志。

## sp\_apply\_job\_to\_targets

sp\_apply\_job\_to\_targets 系统存储过程为目标服务器或目标服务器组添加一项工作。

### 语法

```
sp_apply_job_to_targets {job_id | job_name }
```

```
[ , target_server_groups ]  
[ , target_servers ][ , operation ]
```

变量

job\_id

job\_id 是工作的唯一标识符，缺省值为 NULL。

job\_name

job\_name 是工作的 varchar(100)型名称，缺省值为 NULL。

target\_server\_groups

target\_server\_groups 是 varchar(1024)型数据，可以是一个或多个目标服务器组，各服务器组之间用逗号隔开。缺省值为 NULL。

target\_server

target\_server 是 varchar(1024)型数据，可以是一个或多个目标服务器，服务器之间用逗号隔开，缺省值为 NULL。

operation

operation 是 varchar(7)型数据，它确定如何从服务器中删除工作。其中，APPLY=从已应用的服务器删除(缺省)，REMOVE=从未应用中的服务器删除。

## sp\_approlepassword

sp\_approlepassword 系统存储过程可对使用的数据库中某个应用角色的口令进行修改。

### 语法

```
sp_approlepassword [ @rolename = ] role [ , [ @newpwd = ] password ]
```

### 变量

[ @rolename = ] role	role 是应用角色的系统名称。
[ @newpwd = ] password	password 是新口令的系统名称，缺省值为 NULL。

## sp\_article\_validation

sp\_article\_validation 系统存储过程可验证文章的数据。

### 语法

```
sp_article_validation [ @publication = ] publication [ , [ @article = ] article ]  
[ , [ @rowcount_only = ] rowcount_only ]  
[ , [ @full_or_fast = ] full_or_fast ]  
[ , [ @shutdown_agent = ] shutdown_agent ]
```

## 变量

[ @publication = ] publication	publication 是含有该文章的出版物的系统名称。
[ @article = ] article	article 是文章的系统名称。
[ , [ @rowcount_only = ] rowcount_only ]	rowcount_only 是位型数据，它确定是否只返回表的 rowcount。其值为： NULL= 仅返回 Rowcount。 1= 返回 Rowcount 和 Checksum。
[ , [ @full_or_fast = ] full_or_fast	full_or_fast 是 tinyint 型数据，它确定如何计算 rowcount。其中，0=使用 COUNT(*) 进行全运算。 1= 在 sysindexes 中进行快速运算(如统计数据不是当前值，结果可能不准确)。 2=如果系统检测到一个需求，通过快速计数和全运算来进行快速计算。
[ @shutdown_agent = ] shutdown_agent	shutdown_agent 是一个位型数据，决定当校验完成时是否关闭复制代理。其中，D=复制代理不关闭； 1=校验完成后复制代理关闭。

## sp\_articlecolumn

sp\_articlecolumn 系统存储过程可定义一篇文章中的列。

### 语法

```
sp_articlecolumn { [ @publication = ]      publication      , [ @article = ]  
                  article}  
[ , [ @column = ]      column      ][ , [ @operation = ]      operation      ]  
[ , [ @refresh_synctran_procs = ]      refresh_synctran_procs      ]
```

### 变量

[ @publication = ] publication pblication 是含有该文章的出版物的系统名称。

[ @article = ] article article 是该文章的系统名称。

[ @column = ] column column 是列的系统名称，缺省值为 NULL，对所有的列。

[ @operation = ] operation operation 是 nvarchar(4)型数据，代表列的状态。其中，Add = 复制列，Drop = 不复制列。

[ @refresh\_synctran\_procs = ] refresh\_synctran\_procs refresh\_synctran\_procs 是一个位型数据。其中，1=使支持同步事务的存储过程匹配复制的列。0=使支持同步事务的存储过程不匹配复制的列。

## sp\_articlefilter

sp\_articlefilter 系统存储过程可添加一个过滤器存储过程，该过滤器存储过程可水平划分已发布的表中的数据。

### 语法

```
sp_articlefilter { [ @publication = ]      publication      , [ @article = ]
    article      }
[ , [ @filter_name = ]      filter_name      ][ , [ @filter_clause = ]
filter_clause      ]
```

### 变量

[ @publication = ] publication

publication 是出版物的系统名称。

[ @article = ] article

article 是文章的系统名称。

[ @filter\_name = ] filter\_name

filter\_name 是要建立的存储过程的 nvarchar(46)型名称。该存储过程与过滤器使用相同的名称。缺省值为 NULL。

[ @filter\_clause = ] filter\_clause

filter\_clause 是一个文本，是水平划分数据的 WHERE 子句，输入时省略关键字 WHERE。



## sp\_articlesynctranprocs

sp\_articlesynctranprocs 系统存储过程可在 Publisher 中添加一个同步事务存储过程，用于 Subscriber 中的同步事务触发程序，以及用 sp\_addsyncstriggers 系统存储过程创建的触发程序。

### 语法

```
sp_articlesynctranprocs { [ @publication = ]      publication      ,  
[ @article = ]      article      , [ @ins_proc = ]      ins_proc      ,  
[ @upd_proc = ]      upd_proc      , [ @del_proc = ]      del_proc      }  
[ , [ @autogen = ]      autogen      ]
```

### 变量

[ @publication = ] publication publication 是出版物的系统名称。

[ @article = ] article article 是文章的系统名称。

[ @ins\_proc = ] ins\_proc ins\_proc 是用于同步事务插入的文章存储过程的系统名称。

[ @upd\_proc = ] upd\_proc upd\_proc 是用于同步事务更新的文章存储过程的系统名称。

[ @del\_proc = ] Del\_proc del\_proc 是用于同步事务删除的文章存储过程的系统名称。

[ @autogen = ] autogen autogen 是 nvarchar(5)型数据，其

值为 true 或 false，它确定是否自动创建同步事务存储程序。缺省值为 true。

## sp\_articleview

sp\_articleview 系统存储过程添加文章视图，该视图可垂直或水平划分表格，以便将表格中的数据传送到目标表格。

### 语法

```
sp_articleview { [ @publication = ]      publication      , [ @article = ]
                article                    }
[ , [ @view_name = ]      view_name        ][ , [ @filter_clause = ]
filter_clause            ]
```

### 变量

```
[ @publication = ]      publication
[ @article = ]      article
[ @view_name = ]      view_name
[ @filter_clause = ]      filter_clause
```

publication 是出版物的系统名称。

article 是文章的系统名称。

view\_name 是创建以划分文章(表)的视图名。缺省值为 NULL。

filter\_clause 是一个文本，是水平划分数据的 WHERE 子句。输入时省略关键字 WHERE。

## sp\_attach\_db

sp\_attach\_db 系统存储过程可将一个数据库附加给 SQL Server。当 sp\_detach\_db 系统存储过程将数据库文件从 SQL Server 中分解出来后，如果有必要，该系统存储过程可用于升级过程。

### 语法

```
sp_attach_db [ @dbname = ] dbname ,  
[ @filenamen = ] filename_n [ ,... n ]
```

### 变量

[ @dbname = ] dbname	dbname 是要附加给 SQL Server 的数据库的系统名称。
[ @filename_n = ] filename_n	filename_n 是数据库文件的 nvarchar(260) 型路径和物理名称。该变量最多可含 16 个文件名。

## sp\_attach\_single\_file\_db

sp\_attach\_single\_file\_db 系统存储过程可给 SQL Server 附加带有一个数据库文件的数据库。当 sp\_detach\_db 系统存储过程将数据库文件从 SQL Server 上分

解出来之后，如果有必要，该系统存储过程可用于升级过程。

### 语法

```
sp_attach_single_file_db [ @dbname = ]      dbname      ,  
[ @physname = ]      physical_name  
    变量
```

[ @dbname = ]      dbname	dbname 是要附加给 SQL Server 的数据库的系统名称。
[ @physname = ]      physical_name	filename_n 是数据库文件的 nvarchar(260)型路径和物理名称。

## sp\_autostats

sp\_autostats 系统存储过程可显示或修改一个设置，该设置决定 UPDATE STATISTICS 是否能够自动启动，应用于表格上的一个或多个索引。

### 语法

```
sp_autostats      table_name      [ , stats_flag  ] [ , index_name  ]
```

### 变量

table\_name

stats\_flag

table\_name 是表格的系统名称。

stats\_flag 是 varchar(10)型数据，缺省值为 NULL。其中，ON=UPDATE STATISTICS 设置可以自动启动，OFF=

不能自动启动。

index\_name

index\_name 是索引的系统名称，缺省值为 NULL。

## sp\_bindefault

sp\_bindefault 系统存储过程将 DEFAULT 绑定到一列或用户定义的数据类型上。

### 语法

```
sp_bindefault [ @defname = ]      default      ,  
[ @objname = ]      object_name  
[ , [ @futureonly = ]      futureonly_flag      ]  
      变量
```

default

default 是预先创建的 DEFAULT 的 nvarchar(386)型名称。

object\_name

object\_name 是表 / 列格式的 nvarchar(257)型数据，也可以是数据类型名。

futureonly\_flag

futureonly\_flag 是 varchar(15)型数据，当将 DEFAULT 绑定到用户定义的数据类型上时使用。其中，Futureonly= 禁止绑定到数据类型上的现有列继承

缺省值；然而，所有未来的列将具有 DEFAULT 的特征。NULL=缺省值。

## sp\_bindrule

sp\_bindrule 系统存储过程可将规则绑定到一列或用户定义的数据类型上。

### 语法

```
sp_bindrule { rule } [ , object_name [ , futureonly_flag ]
```

### 变量

rule

rule 是预先创建的 RULE 的 nvarchar(386)型名称。

object\_name

object\_name 是表 / 列格式的 nvarchar(257)型数据，也可以是数据类型名。

future\_only\_flag

future\_only\_flag 是 varchar(15)型数据，当将 DEFAULT 绑定到用户定义的数据类型上时使用。其中，Futureonly=禁止绑定到数据类型上的列继承缺省值；然而，所有未来的列都将具有 DEFAULT 的特征。NULL=缺省值。

## sp\_bindsession

sp\_bindsession 系统存储过程将链接绑定到其他链接上，或将链接从其他链接中解除。

### 语法

```
sp_bindsession { bind_token | NULL }
```

变量 bind\_token 是 varchar(255) 型数据，可用 sp\_getbindtoken 或 srv\_getbindtoken 系统存储程序获得。该系统存储过程可查找事务空间，以便允许链接使用相同的事务。

## sp\_browsereplcmds

sp\_browsereplcmds 系统存储过程可显示分布式数据库中已复制的命令。

### 语法

```
sp_browsereplcmds [ [ @xact_seqno_start = ] xact_seqno_start ]  
[ , [ @xact_seqno_end = ] xact_seqno_end ]  
[ , [ @originator_id = ] originator_id ]  
[ , [ @publisher_database_id = ] publisher_database_id ]  
[ , [ @article_id = ] article_id ]  
[ , [ @command_id = ] command_id ]
```

### 变量

[ @xact\_seqno\_start = ]  
xact\_seqno\_start

[ @xact\_seqno\_end = ]  
xact\_seqno\_end

[ @originator\_id = ] originator\_id

[ @publisher\_database\_id = ]  
publisher\_database\_id

[ @article\_id = ] article\_id

[ @command\_id = ] command\_id

xact\_seqno\_start 是 nchar(22)型数据，表示需要显示的最低精确序列号。缺省值为 0x000000000000000000000000。

xact\_seqno\_end 是 nchar(22)型数据，代表需要显示的最高精确序列号。缺省值为 0xFFFFFFFFFFFFFFFFFFFFFFFF。

originator\_id 是一个整数，显示带有特定 originator\_id 的命令。

publisher\_database\_id 是一个整数，显示带有特定 publisher\_database\_id 的命令。

article\_id 是一个整数，显示与特定 article\_id 有关的命令。

command\_id 是一个整数，代表 MSrepl\_commands 中命令的位置。需要使用所有其他的变量，且 xact\_seqno\_start 必须等于 xact\_seqno\_end。



## sp\_catalogs

sp\_catalogs 系统存储过程可显示链接服务器的类别列表。如果正在访问 SQL Server，它等效于数据库列表。

### 语法

```
sp_catalogs { linked_svr }
```

变量 linked\_server 是链接服务器的系统名称。

## sp\_certify\_removable

sp\_certify\_removable 系统存储过程可确认数据库能放在可拆介质上。

### 语法

```
sp_certify_removable [ @dbname = ] dbname [ , [ autofix = ]  
auto ]
```

### 变量

[ @dbname = ] dbname

[ @autofix = ] auto

dbname 是数据库的系统名称。

auto 改变数据库，并把数据库和对象的拥有关系传送给系统管理员，删除用户建立的数据库用户，包括 Microsoft SQL Server 7 中不包含的许可。

## sp\_change\_agent\_parameter

sp\_change\_agent\_parameter 系统存储过程更新复制代理配置文件的 MSAgent\_parameters 系统表中的变量。该系统存储过程可用于所有复制类型。

### 语法

```
sp_change_agent_parameter [ @profile_id = ] profile_id,  
[ @parameter_name = ] parameter_name ,  
[ @parameter_value = ] parameter_value
```

### 变量

[ @profile\_id = ] profile\_id

profile\_id 是一个代表配置文件标识符的整数。

[ @parameter\_name = ]  
parameter\_name

parameter\_name 是 MSAgent\_parameters 系统表中的参数的系统名。

[ @parameter\_value = ]  
parameter\_value

parameter\_value 是一个 nvarchar(255) 型数据，代表新参数值。

## sp\_change\_agent\_profile

sp\_change\_agent\_profile 系统存储过程更新复制代理配置文件的

MSAgent\_profiles 系统表中的变量。该系统存储过程可用于所有的复制类型。

#### 语法

```
sp_change_agent_profile [ @profile_id = ] profile_id,  
[ @property_name = ] property_name ,  
[ @property_value = ] property_value
```

#### 变量

[ @profile_id = ] profile_id	profile_id 是一个代表配置文件标识符的整数。
[ @property_name = ] property_name	property_name 是 MSAgent_profile 系统表中的属性的系统名。
[ @property_value = ] property_value	property_value 是 nvarchar(255)型数据，代表新的属性值。

## sp\_changearticle

sp\_changarticle 系统存储过程可修改未订阅的文章，用以快照及事务性复制。

#### 语法

```
sp_changearticle [[ @publication = ] publication ][ , [ @article = ]  
article ]  
[ , [ @property = ] property ][ , [ @value = ] value ]
```

## 变量

[ @publication = ] publication publication 是出版物的系统名称。

[ @article = ] article

article 是文章的系统名。

[ @property = ] property

property 是数据类型的系统名，代表要修改的文章属性。

[ @value = ] value

value 是 nvarchar(255) 型数据，包含文章属性的值。

要改变的文章属性和属性值可以是：

Name

name 是修改的文章名。

Description

description 是修改的出版工作描述。

sync\_object

sync\_object 是创建同步输出文件的表或视图的名称。

Type

文章类型可以是 Logbased=基于日志的文章。Logbased manualfilter=基于日志的文章，带手动过滤器。Logbased manualview=基于日志的文章，带手动视窗。Logbased manualboth=基于日志的文章，带手动过滤器和手动视图。

ins\_cmd ins\_cmd

ins\_cmd ins\_cmd 是要执行的 INSERT

del\_cmd del\_cmd

upd\_cmd upd\_cmd

Filter

dest\_table(dest\_object)

creation\_script creation\_script

pre\_creation\_cmd

Status

include column names

语句，否则由 SQL Server 生成它。

del\_cmd del\_cmd 是要执行的 DELETE 语句，否则由 SQL Server 生成它。

upd\_cmd upd\_cmd 是要执行的 UPDATE 语句，否则由 SQL Server 生成它。

Filter 是一个用于水平划分表格的存储过程，缺省值为 NULL。

dest\_table 是目标表的名称 (如果该名称与源表名称不同)。

creation\_script creation\_script 是建立目标表的文章模式脚本文件的路径 \ 名字。

pre\_creation\_cmd 是在同步之前撤消、删除或截短目标表的预设命令。其值为：None=没有命令，Drop=撤消目标表，Delete=删除目标表，Truncate=截短目标表。

Status 是修改的属性状态。

include column names 表明在 INSERT 语句中允许对列名进行的操作。其值可以是：No column names=不允许操作列名，Owner qualified=被授权的

schema option

destination\_owner

NULL

拥有者允许操作列名，Not owner qualified 不允许已授权的拥有者操作列名，String literals | parameters=确定日志读取器产生的命令使用 string\_literal 命令格式还是参数化的命令。

schema option 是二进制(8)位图，是文章的模式生成选项。其值可以是：  
0x00 = 使用用户提供的 CreationScript，0x01 = 生成 CreationScript，0x11 = 生成 CREATE TABLE，成簇索引，非成簇索引，以及将用户创建的数据类型改为基本类型的位图。0x10 = 创建成簇索引，0x20 = 将用户创建的数据类型改为基本类型，0x40 = 创建一个或多个非成簇索引，0x80 = 在主键上声明引用的完整性

destination\_owner 是目标对象拥有者的名称。

表示显示所有的属性值。

## sp\_changedbowner

sp\_changedbowner 系统存储过程可在正使用的数据库中运行，以改变该数据库的拥有者。新数据库的拥有者不必是正使用的数据库中的用户。如果没有提供 drop\_alias\_flag，缺省的情况下，将撤消旧数据库拥有者的别名。

### 语法

```
sp_changedbowner [ @loginame = ] login [ , [ @map = ]  
drop_alias_flag ]
```

### 变量

[ @loginame = ] login	login 是新数据库拥有者的登录 ID。
[ @map = ] drop_alias_flag	drop_alias_flag 是 varchar(5)型数据，其值为 TRUE 或 FALSE。True 值表示将撤消别名是旧新数据库拥有者的用户。如果没有提供 drop_alias_flag，缺省情况下将撤消旧数据库拥有者的别名。

## sp\_changedistpublisher

sp\_changedistpublisher 系统存储过程可修改分布式 Publisher 的一些特征。

### 语法

```
sp_changedistpublisher [ @publisher = ] publisher
    [ , [ @property = ] property ] [ , [ @value = ] value ]
```

## 变量

[ @publisher = ] publisher	publisher 是 Publisher 的系统名称。
[ @property = ] property	property 定义了 Publisher 的可修改的属性。其中，Active = 活动注册键的值。Distribution_db = 分布式数据库注册键中的值，Login = 登录注册键中的值，Password = 口令注册键中的值，Security_mode=安全模式注册键中的值，Working_directory=工作目录注册键中的值，NULL=现有的属性和值。
[ @value = ] value	value 是 nvarchar(255)型数据，包含 Publisher 属性的值。

## sp\_changedistributiondb

sp\_changedistributiondb 系统存储过程可修改分布式数据库的一些特征。

### 语法



```
sp_changedistributiondb [ @database = ]      database
[ , [ @property = ]      property      ] [ , [ @value = ]      value      ]
    变量
```

```
[ database = ]      database
[ @property = ]      property
```

database 是数据库的系统名。  
property 是数据类型的系统名称，代表 Publisher 中要更改的属性。其值可以是：History\_retention=历史表的保留阶段，Max\_distretention=发布的最长保留阶段，Min\_distretention = 发布的最短保留阶段，Security\_mode=安全模式注册键的值，NULL=显示现有的特性和值。

```
[ @value = ]      value
```

value 是 nvarchar(255)型数据，包含分布式数据库的特性值。

```
sp_changedistributor_property
```

sp\_changedistributor\_property 系统存储过程可为所有类型的复制修改 Distributor 的一些特征。

语法

```
sp_changedistributor_property [ [ @property = ]      property      ]
```

[ , [ @value = ] value ]  
变量

[ @property = ] property

property 是给出的 Distributor 的属性的系统名。其中，working\_directory=工作目录注册键的值，NULL=列出现有的特性和值。

[ @value = ] value

value 是 nvarchar(255)型数据，包含 Distributor 的特性值。

## sp\_changedistributor\_password

sp\_changedistributor\_password 系统存储过程可用于所有类型的复制，为 Distributor 创建新口令。如果 Distributor 是本地 Distributor，则该系统存储过程可用来修改 distributor\_admin 口令。

### 语法

sp\_changedistributor\_password [ @password = ] password  
变量 [ @password = ] password 是取代旧口令的新口令。

## sp\_changegroup

sp\_changegroup 系统存储过程可用于向后兼容。使用 Microsoft SQL Server 7 版本的程序员可用 sp\_addrolemember 取代 sp\_changegroup，来改变数据库用户的角色成员关系。使用 sp\_changegroup 将允许通过撤消其他角色成员关系，把数据库放在唯一的角色中；而版本 7 中新的角色变化表允许把数据库用户放在多个角色中。

### 语法

```
sp_changegroup [ @grpname = ] role , [ @username = ] user
```

### 变量

[ @grpname = ] role	role 是数据库用户被指定的角色，它必须是一个已有的数据库角色。
[ @username = ] user	user 是要改变为新角色的数据库用户的系统名称。

## sp\_changemergearticle

sp\_changemergearticle 系统存储过程可修改合并文章的一些特征。

### 语法

```
sp_changemergearticle [ @publication = ] publication [ , [ @article = ] article ]
```

[ , [ @property = ] property ] [ , [ @value = ] value ]  
变量

[ @publication = ] publication publication 是包含文章的出版物的系统名称。

[ @article = ] article article 是文章的系统名称，它必须是一个表名（例如，MyTable）。

[ @property = ] property property 是数据类型的系统名，代表要改变的文章特征。

[ @value = ] value value 是 nvarchar(255)型数据，含有文章的特性值。

合并文章中要修改的特性和特性值有：

namen ame 是合并文章的新名称。  
[ @description = ] description description 是文章的 nvarchar(255)型描述。缺省值为 NULL。

[ @column\_tracking = ] column\_tracking column\_tracking 是 nvarchar(10)型数据，其值为 true 或 false，是列级跟踪的指示器。缺省值 false 表示关闭列级跟踪。如果表格已用合并复制发布出去，就使用相同的列跟踪的值，并将该变量设置为 true。

[ @status = ] status status 是文章的 nvarchar(10)型状态，缺省值 unsynced 表明初始数据

```
[ @pre_creation_cmd = ]  
pre_creation_cmd
```

```
[ @creation_script = ]  
creation_script
```

```
[ @schema_option = ]  
schema_option
```

已在目标中指定。如状态设置为活动的，则运行初始过程脚本，否则可将状态设置为 inactive(不活动)。

pre\_creation\_cmd 是 nvarchar(10) 型预建方式，其中 None= 如果 Subscriber 中已存在该表，则不进行任何操作；Delete=按子集过滤器中的 WHERE 子句进行删除和过滤。Drop=在重建表之前将表撤消(缺省值)。Truncate=与 Delete 相同，但删除的是页(不需要 WHERE 子句)。

creation\_script 是文章的 nvarchar(255)型预建脚本(可选项，缺省值为 NULL)。

schema\_option 是文章的模式生成的二进制(8)位图，其中 0x00=通过 Snapshot 取消脚本的编辑，并使用所提供的 CreationScript；0x01=生成对象创建(CREATE TABLE;CREATE PROCEDURE);0x10=生成成簇索引；0x20=将用户定义的数据类型转换为基本数据类型；0x40=生成非成簇索

```
[ @subset_filterclause = ]  
subset_filterclause
```

```
[ @article_resolver = ]  
article_resolver
```

```
[ @resolver_info = ]  
resolver_info
```

## sp\_changemergefilter

sp\_changemergefilter 系统存储过程可修改过滤器的一些特征。

语法

引；0x80=在主键上提供已声明的引用完整性。0xf1(缺省)=制作创建表，创建成簇索引，非成簇索引，将用户定义的数据类型转换成基本数据类型；0x01=存储过程文章的缺省值。

subset\_filterclause 是一个 nvarchar(2000)型 WHERE 子句，它确定水平过滤。

article\_resolver 是文章的 varchar(255)型的定制解析程序。存储过程 sp\_enumcustomsolvers 列举了所有的定制解析程序。如果该参数为 NULL，则使用系统提供的解析程序。

resolver\_info 是用作定制解析程序的存储过程的系统名，缺省值为 NULL。

```

sp_changemergefilter [ @publication = ]      publication      [ , [ @ article = ]
article      ]
[ , [ @filtername = ]      filtername      [ , [ @property = ]      property
][ , [ @value= ]      value      ]
变量

```

[ @publication = ]      publication	publication 是出版物的系统名称。
[ @ article = ]      article	article 是文章的系统名称。
[ @ filtername = ]      filtername	filtername 是过滤器的系统名称。
[ @property = ]      property	property 是合并过滤器的特征。可修改以下特征：Filtername = 过滤器的名称。Join_filterclause = 省略字 WHERE 的 WHERE 子句。Join_table_name = 连接表的系统名。
[ @value = ]      value	value 是可修改的属性的 nvarchar(2000)型数值。

## sp\_changemergepublication

sp\_changemergepublication 系统存储过程可修改合并出版物的一些特征。  
语法

```
sp_changemergepublication [ @publication = ]      publication
[ , [ @property = ]      property      ][ , [ @value = ]      value      ]
    变量
```

[ @publication = ] publication publication 是出版物的系统名称。

[ @property = ] property property 是数据类型的系统名称，代表要改变的 Publication 特征。

[ @value = ] value value 是 nvarchar(255)型数据，包含 Publication 的特性值。

要修改的合并出版物属性和属性值：

[ @description = ] description description 是 nvarchar(255)型的出版物描述。

Status status 是 Publication 的状态，其值可以是 unsynced(不同步), active(活动)或 inactive(不活动)。

[ @retention = ] retention retention 是一个整数，代表保存给定的出版物的天数，缺省值为 60 天。在保留期间，出版物必须合并，否则它将不同步。

[ @sync\_mode = ] sync\_mode sync\_mode 是 nvarchar(10)型的同步模式。缺省值为 native。其中，Native = 成批拷贝，Character = 字符成批



[ @allow\_push = ] allow\_push

[ @allow\_pull = ] allow\_pull

[ @allow\_anonymous = ]  
allow\_anonymous

[ @enabled\_for\_internet = ]  
enabled\_for\_internet

[ @centralized\_conflicts = ]  
centralized\_conflicts

[ @default\_access = ]  
default\_access

拷贝。

allow\_push 的值是 true 或 false [ nvarchar(5)型 ], 它确定是否允许推出订阅出版物。

allow\_pull 的值是 true 或 false [ nvarchar(5)型 ], 它确定是否允许推入订阅出版物。

allow\_anonymous 的值是 true 或 false [ nvarchar(5) ] 型, 它确定是否允许匿名订阅出版物。

enabled\_for\_internet 是 nvarchar(5) 型数据, 缺省值为 false, 它确定出版物能否用于 Internet。FTP 用来向 Subscriber 发送快照。

centralized\_conflicts 是 nvarchar(5) 型数据, 它确定 Publisher 中是否存有冲突记录, 缺省值为 true。False 意味着 Subscriber 中出现存储记录。

default\_access 是 nvarchar(5) 型数据, 缺省值为 true。

## sp\_changemergepullsubscription

sp\_changemergepullsubscription 系统存储过程可修改合并推入订阅的一些特征。订阅器是正使用的服务器，subscriber\_db 是正使用的数据库。

### 语法

```
sp_changemergepullsubscription [[ @publication = ]      publication      ]
[ , [ @publisher = ]      publisher      ] [ , [ @publisher_db = ]
publisher_db      ]
[ , [ @property = ]      property      ] [ , [ @value = ]      value      ]
```

### 变量

[ @publication = ] publication  
[ @publisher = ] publisher

[ @publisher\_db = ]  
publisher\_db

[ @property = ] property

[ @value = ] value

publication 是出版物的系统名称。

publisher 是 Publisher 的系统名称 (服务器名称)。

publisher\_db 是 Publisher 数据库的系统名称，缺省值为 NULL。

property 是数据类型的系统名称，代表要改变的合并推入订阅特征。

value 是 nvarchar(255) 型数据，含有合并推入订阅的属性值。

要改变的合并推入订阅属性和属性值：

[ @priority = ] priority	subscription_priority 是 real 型数据，缺省值为 NULL，可用于解决冲突。其中，Local = 0.0，Anonymous = 0.0
[ @sync_type = ] sync_type	sync_type 是 nvarchar(15)型描述同步类型，缺省值为 automatic(自动)。其中，Automatic = 模式和数据转换。Nosync = 没有模式和数据转换。
[ @description = ] description	description 是 nvarchar(255)型推入订阅描述，缺省值为 NULL。

## sp\_changemergesubscription

sp\_changemergesubscription 系统存储过程可修改合并推出或推入订阅的一些特征。

### 语法

```
sp_changemergesubscription [[ @publication = ] publication ]  
[ , [ @subscriber = ] subscriber [ , [ @subscriber_db = ]  
subscriber_db ] [ , [ @property = ] property ] [ , [ @value  
= ] value ]
```

### 变量

[ @publication = ] publication publication 是出版物的系统名称。

[ @subscriber = ] subscriber

subscriber 是 Subscriber 的系统名称 (服务器名称)。

[ @subscriber\_db = ]  
subscriber\_db

subscriber\_db 是 Subscriber 数据库的系统名称。

[ @property = ] property

property 是数据类型的系统名称，代表要改变的合并订阅特征。

[ @value = ] value

value 是 nvarchar(255)型数据，带有合并订阅的属性值。

要改变的合并订阅属性和属性值：

[ @priority = ] priority

subscription\_priority 是一个 real 型数据，缺省值为 NULL，可用于解决冲突。其中，Local = 0.0，Anonymous = 0.0。

[ @sync\_type = ] sync\_type

sync\_type 是 nvarchar(255)型订阅同步类型，缺省为 automatic(自动)。其中，Automatic = 模式和数据转换，None = 没有模式 but 数据转换。

[ @description = ] description

discription 是 nvarchar(255)型合并订阅描述，缺省值为 NULL。

## sp\_changeobjectowner

sp\_changeobjectowner 系统存储过程可在当前数据库中运行，以改变对象的拥有者。

### 语法

```
sp_changeobjectowner [ @objname = ]          object          [ , [ @newowner = ]  
owner          ]
```

### 变量

[ @objname = ] object

object 是当前数据库中表格、视图或存储过程的已授权的拥有者  
nvarchar(517)型可选名称。

[ @newowner = ] owner

owner 是数据库用户(安全帐户)的系统名称。

## sp\_changepublication

sp\_changepublication 系统存储过程可修改出版物的一些特征。

### 语法

```
sp_changepublication [[ @publication = ]      publication      ]  
[ , [ @property = ]      property      ][ , [ @value = ]      value      ]
```

## 变量

[ @publication = ]	publication	publication 是出版物的系统名称。
[ @property = ]	property	property 是数据类型的系统名称，代表要改变的 Publication 特征。
[ @value = ]	value	value 是 nvarchar(255)型数据，包含 Publication 的属性值。
要改变的出版物属性和属性值：		
[ @description = ]	description	description 是 nvarchar(255)型出版物描述。缺省值为 NULL。
[ @taskid = ]	taskid	taskid 是一个整数，代表预先创建的调度任务。该变量仅向后兼容。使用 sp_addpublication_snapshot 系统存储过程。
[ @restricted = ]	restricted	restricted 是 nvarchar(10)型数据，其值为 true 或 false，它指定谁可以查看出版物，其中 False=任何人均可查看，True=只有已授权的 Subscribers 可查看。
[ @sync_method = ]	sync_method	sync_method 是 nvarchar(13)型同步方式，缺省值为 native。其中 Native=成批拷贝，Character=字符成批拷

[ @repl\_freq = ] repl\_freq

[ @status = ] status

immediate\_sync

independent\_agent

贝。

repl\_freq 是 nvarchar(10)型复制频率类型。其中 Continuous=Publisher 发送所有基于日志的事务。Snapshot=Publisher 发送安排好的同步事件。

status 是 nvarchar(8)型数据，它确定决定首次重建出版物时，出版数据是否可用于 Subscriber，缺省值为 inactive。其中，inactive=出版数据不能立即使用，active=出版数据可立即使用。

如果 immediate\_sync 设为 true，表示 Snapshot Agent 每次运行时都创建出版物的同步文件。如果设为 false，表示当检测到新订阅时，创建同步文件。

如果 independent\_agent 设为 true，表明出版物有一个独立的 Distribution Agent。如果设为 false，表明每个 Publisher 和 Subscriber 组合都有一个

enabled\_for\_internet

Distribution Agent。

如果 enabled\_for\_internet 设为 true，表示出版物可用于 Internet，出版物的同步文件位于 \Repldate\Ftp 中。

allow\_push

如果 allow\_push 设为 true，表示允许推出订阅出版物。

allow\_pull

如果 allow\_pull 设为 true，表示允许推入订阅出版物。

allow\_anonymous

如果 allow\_anonymous 设为 true，表示允许匿名订阅出版物。

Retention

retention 表示订阅操作的保留阶段，以天数为单位。

default\_access

如果 default\_access 设为 true，表示使用 Publisher 的缺省出版物访问列表。

sp\_changesubscriber

sp\_changesubscriber 系统存储过程可修改 Subscriber 的一些特征。

语法

```
sp_changesubscriber { [ @subscriber = ] subscriber } [ , [ @type = ]
```



```

type ]
[ , [ @login = ] login ][ , [ @password = ] password ]
[ , [ @commit_batch_size = ] commit_batch_size ]
[ , [ @status_batch_size = ] status_batch_size ]
[ , [ @flush_frequency = ] flush_frequency ]
[ , [ @frequency_type = ] frequency_type ]
[ , [ @frequency_interval = ] frequency_interval ]
[ , [ @frequency_relative_interval = ] frequency_relative_interval
]
[ , [ @frequency_recurrence_factor = ] frequency_recurrence_factor
]
[ , [ @frequency_subday = ] frequency_subday ]
[ , [ @frequency_subday_interval = ] frequency_subday_interval ]
[ , [ @active_start_time_of_day = ] active_start_time_of_day ]
[ , [ @active_end_time_of_day = ] active_end_time_of_day ]
[ , [ @active_start_date = ] active_start_date ]
[ , [ @active_end_date = ] active_end_date ]
[ , [ @description = ] description ][ , [ @security_mode = ]
security_mode ]

```

### 变量

[ @subscriber = ] subscriber      subscriber 是添加为 Subscriber 的  
服务器的系统名称。

[ @type = ] type	type 是 tinyint 型数据，代表 Subscriber 的类型。其中，0 = Microsoft SQL Server Subscriber，1 = ODBC 数据源服务器
[ @login = ] login	login 是 SQL Server Authenticated 登录的系统名称，缺省为 SA。
[ @password = ] password	password 是 SQL Server Authenticated 登录的系统名口令，缺省为 %，表示口令属性没有变化。
[ @commit_batch_size = ] commit_batch_size	commit_batch_size 是一个整数，确定要将多少事务传送到 Subscriber 上，并一起运作。缺省为 NULL。
[ @status_batch_size = ] status_batch_size	status_batch_size 被保留，以备将来使用。缺省为 NULL。
[ @flush_frequency = ] flush_frequency	flush_frequency 是一个整数，它确定事务发布到 Subscriber 上后，从分布式数据库表格中删除的频率，缺省为 NULL。
[ @frequency_type = ] frequency_type	frequency_type 是一个整数，它确定如何安排 Distribution Agent 的运行。其中 1=一次，2=根据需要，4=每天，8=每周，16=每月，32=相关的

```
[ @frequency_interval = ]  
frequency_interval
```

```
[ @frequency_relative_interval  
= ] frequency_relative_interval
```

```
[ @frequency_recurrence_factor  
= ] frequency_recurrence_factor
```

```
[ @frequency_subday = ]  
frequency_subday
```

```
[ @frequency_subday_interval  
= ] frequency_subday_interval
```

月份，64=自动启动，128=重复，NULL=缺省值。

frequency\_interval 是一个整数，代表 Distribution Agent 的运行日期，其中，1=周日，2=周一，3=周二，4=周三，5=周四，6=周五，7=周六，8=白天，9=平日，10=周末白天，NULL=缺省值。

frequency\_relative\_interval 是一个整数，当变量 frequency\_type 为相关的月份时使用。其中 1=第一，2=第二，4=第三，8=第四，16=末尾，NULL=缺省值。

frequency\_recurrence\_factor 是一个整数，与 frequency\_type 一起使用，缺省值为 NULL。

frequency\_subday\_interval 是一个整数，代表在选定的时间段内，何时再安排运行。1=一次，2=秒，4=分钟，8=小时，NULL=缺省值。

frequency\_subday\_interval 是一个整数，代表 frequency\_subday 的间

[ @active\_start\_time\_of\_day = ]  
active\_start\_time\_of\_day

[ @active\_end\_time\_of\_day = ]  
active\_end\_time\_of\_day

[ @active\_start\_date = ]  
active\_start\_date

[ @active\_end\_date = ]  
active\_end\_date

[ @description= ] description

[ @security\_mode = ]  
security\_mode

隔。缺省值为 NULL。

active\_start\_time\_of\_day 是一个整数，代表首次安排 Distribution Agent 远行的时刻。缺省值为 NULL (格式为 HHMMSS)。

active\_end\_time\_of\_day 是一个整数，代表停止安排 Distribution Agent 运行的时刻。缺省值为 NULL (格式为 HHMMSS)。

active\_start\_date 是一个整数，代表首次安排 Distribution Agent 的日期。缺省值为 NULL (格式为 YYYYMMDD)。

active\_end\_date 是一个整数，代表停止安排 Distribution Agent 的日期。缺省值为 NULL (格式为 YYYYMMDD)。

description 是 Subscriber 的 nvarchar(255) 型描述。缺省值为 NULL。

security\_mode 是一个代表安全模式的整数，其中 0=SQL Server 验证；

1=Windows NT 验证(缺省)。

## sp\_changesubscriber\_schedule

sp\_changesubscriber\_schedule 系统存储过程可修改 Distribution Agent 和 Merge Agent 的一些特征。用 sp\_changesubscriber 系统存储过程可修改 Distribution Agent 的时间表，或为 Merge Agent 创建缺省时间表。

### 语法

```
sp_changesubscriber_schedule [ @subscriber = ] subscriber ,
[ @agent_type = ] type [ , [ @frequency_type = ] frequency_type ]
[ , [ @frequency_interval = ] frequency_interval ]
[ , [ @frequency_relative_interval = ] frequency_relative_interval
]
[ , [ @frequency_recurrence_factor = ] frequency_recurrence_factor
]
[ , [ @frequency_subday = ] frequency_subday ]
[ , [ @frequency_subday_interval = ] frequency_subday_interval ]
[ , [ @active_start_time_of_day = ] active_start_time_of_day ]
[ , [ @active_end_time_of_day = ] active_end_time_of_day ]
[ , [ @active_start_date = ] active_start_date ]
[ , [ @active_end_date = ] active_end_date ]
```

### 变量

[ @ subscriber = ] subscriber  
[ @ agent\_type = ] agent\_type

[ @ frequency\_type = ] frequency\_type

[ @ frequency\_interval = ]  
frequency\_interval

[ @ frequency\_relative\_interval = ]  
frequency\_relative\_interval

[ @ frequency\_recurrence\_factor = ]

subscriber 是 Subscriber 的系统名称。  
agent\_type 是 smallint 型数据，代表代理的类型。其中 0=Distribution Agent，1=Merge Agent，缺省值为 0。  
frequency\_type 是一个整数，它确定如何安排 Distribution Agent。其中，1=一次，2=根据需要，4=每天，8=每周，16=每月，32=相关的月份，64=自动启动，124=重复，NULL=缺省值。

frequency\_interval 是一个整数，代表 Distribution Agent 的运行日期。其中，1=周日，2=周一，3=周二，4=周三，5=周四，6=周五，7=周六，8=白天，9=平日，10=周末白天，NULL=缺省值。

frequency\_relative\_interval 是一个整数，当变量 frequency\_type 为相关的月份时使用。1=第一，2=第二，4=第三，8=第四，16=末尾，NULL=缺省值。

frequency\_recurrence\_factor 是一个整

frequency_recurrence_factor		数，与 frequency_type 一起使用，缺省值为 NULL。
[ @frequency_subday = ] frequency_subday	=	frequency_subday 是一个整数，代表在选定的时间段内，何时再安排运行。其中 1=一次，2=秒，4=分钟，8=小时，NULL=缺省值。
[ @frequency_subday_interval = ] frequency_subday_interval	=	frequency_subday_interval 是一个整数，代表 frequency_subday 的间隔。缺省值为 NULL。
[ @active_start_time_of_day = ] active_start_time_of_day	=	active_start_time_of_day 是一个整数，代表首次安排 Distribution Agent 的时刻。缺省值为 NULL（格式为 HHMMSS）。
[ @active_end_time_of_day = ] active_end_time_of_day	=	active_end_time_of_day 是一个整数，代表停止安排 Distribution Agent 的时刻。缺省值为 NULL（格式为 HHMMSS）。
[ @active_start_date = ] active_start_date	=	active_start_date 是一个整数，代表首次安排 Distribution Agent 的日期。缺省值为 NULL（格式为 YYYYMMDD）。
[ @active_end_date = ] active_end_date	=	active_end_date 是一个整数，代表停

止安排 Distribution Agent 的日期。  
缺省值为 NULL(格式为  
YYYYMMDD)。

## sp\_changesubstatus

sp\_changesubstatus 系统存储过程可修改已有 Subscriber 的状态。

### 语法

```
sp_changesubstatus [[ @publication = ]      publication      ]
[ , [ @article = ]      article      ]
[ , [ @subscriber = ]  subscriber  ] { , [ @status = ]      status      }
[ , [ @previous_status = ]      previous_status      ]
[ , [ @destination_db = ]      destination_db      ]
[ , [ @frequency_type = ]      frequency_type      ]
[ , [ @frequency_interval = ]      frequency_interval      ]
[ , [ @frequency_relative_interval = ]      frequency_relative_interval
]
[ , [ @frequency_recurrence_factor = ]      frequency_recurrence_factor
]
[ , [ @frequency_subday = ]      frequency_subday      ]
[ , [ @frequency_subday_interval = ]      frequency_subday_interval      ]
[ , [ @active_start_time_of_day = ]      active_start_time_of_day      ]
```



```
[ , [ @active_end_time_of_day = ] active_end_time_of_day ]
[ , [ @active_start_date = ] active_start_date ]
[ , [ @active_end_date = ] active_end_date ]
[ , [ @optional_command_line = ] optional_command_line ]
[ , [ @distribution_jobid = ] distribution_jobid ]
[ , [ @from_auto_sync = ] from_auto_sync ]
[ , [ @ignore_distributor = ] ignore_distributor ]
```

### 变量

[ @publication = ]	publication	publication 是出版物的系统名称，缺省值 % 表示全部出版物。
[ @article = ]	article	article 是文章的系统名称。如果没有提供该参数，则订阅出版物中的所有文章。缺省值 % 指全部文章。
[ @subscriber = ]	subscriber	subscriber 是 Subscriber 的系统名称。缺省值 NULL 指全部 Subscriber。
[ @status = ]	status	status 是订阅状态的系统名称。Active = 订阅服务器同步，且正传送数据。Inactive = 订阅服务器条目，不带有限制出版物的订阅。Subscribed = 被订阅但不同步。
[ @previous_status = ]	previous_status	previous_status 是一个系统名称，代表订阅的以前状态。该变量允许用

```
[ @destination_db = ]  
destination_db
```

```
[ @frequency_type = ]  
frequency_type
```

```
[ @frequency_interval = ]  
frequency_interval
```

```
[ @frequency_relative_interval = ]  
frequency_relative_interval
```

```
[ @frequency_recurrence_factor
```

先前的状态将订阅的设置改变为其他状态。

destination\_db 是包含复制数据的目标数据库的系统名称 (该名称应与 Publication 数据库同名)。缺省值 % 意味着全部目标数据库。

frequency\_type 是一个整数, 它确定如何安排 Distribution Agent。其中 1=一次, 2=根据需要, 4=每天, 8=每周, 16=每月, 32=相关的月份, 64=自动启动, 124=重复, NULL=缺省值。

frequency\_relative\_interval 是一个整数, 代表 Distribution Agent 的运行日期。其中 1=周日, 2=周一, 3=周二, 4=周三, 5=周四, 6=周五, 7=周六, 8=白天, 9=平日, 10=周末白天, NULL=缺省值。

frequency\_interval 是一个整数, 当变量 frequency\_type 为相关的月份时使用。1=第一, 2=第二, 4=第三, 8=第四, 16=末尾, NULL=缺省值。

frequency\_recurrence\_factor 是一

<code>= ] frequency_recurrence_factor</code>	个整数，与 <code>frequency_type</code> 一起使用，缺省值为 <code>NULL</code> 。
<code>[ @frequency_subday = ] frequency_subday</code>	<code>frequency_subday_interval</code> 是一个整数，代表在选定的时间段内，何时再安排运行。其中 1=一次，2=秒，4=分钟，8=小时， <code>NULL</code> =缺省值。
<code>[ @frequency_subday_interval = ] frequency_subday_interval</code>	<code>frequency_subday_interval</code> 是一个整数，代表 <code>frequency_subday</code> 的间隔。缺省值为 <code>NULL</code> 。
<code>[ @active_start_time_of_day = ] active_start_time_of_day</code>	<code>active_start_time_of_day</code> 是一个整数，代表首次安排 <code>Distribution Agent</code> 的时刻。缺省值为 <code>NULL</code> (格式为 <code>HHMMSS</code> )。
<code>[ @active_end_time_of_day = ] active_end_time_of_day</code>	<code>active_end_time_of_day</code> 是一个整数，代表停止安排 <code>Distribution Agent</code> 的时刻。缺省值为 <code>NULL</code> (格式为 <code>HHMMSS</code> )。
<code>[ @active_start_date = ] active_start_date</code>	<code>active_start_date</code> 是一个整数，代表首次安排 <code>Distribution Agent</code> 的日期。缺省值为 <code>NULL</code> (格式为 <code>YYYYMMDD</code> )。
<code>[ @active_end_date = ]</code>	<code>active_end_date</code> 是一个整数，代表

<code>active_end_date</code>		停止安排 Distribution Agent 的日期。缺省值为 NULL(格式为 YYYYMMDD)。
<code>[ @optional_command_line = ] optional_command_line</code>		<code>optional_command_line</code> 是要运行的 <code>nvarchar(4000)</code> 型命令行，缺省值为 NULL。
<code>[ @distribution_jobid = ] distribution_jobid</code>		<code>distribution_jobid</code> 是二进制(16)型数据，缺省值为 NULL，是在发布器上用于订阅的分布式代理的工作标识符。该变量仅在将状态由 <code>active</code> 改变为 <code>inactive</code> 时使用。
<code>[ @from_auto_sync = ] from_auto_sync</code>		<code>from_auto_sync</code> 是位型数据，缺省值为 0。
<code>[ @ignore_distributor = ] ignore_distributor</code>		<code>ignore_distributor</code> 是位型数据，缺省值为 0。

## sp\_change\_subscription\_properties

`sp_change_subscription_properties` 系统存储过程可为订阅的 Publisher 和 Distributor

### 语法

```
sp_change_subscription_properties [ , [ @publisher = ] publisher ]
```

```
[ , [ @publisher_db = ] publisher_db ] [ , [ @publication = ]  
publication ] [ , [ @property = ] property ] [ , [ @value = ]  
value ]  
变量
```

```
[ @publisher = ] publisher  
[ @publisher_db = ] publisher_db
```

publisher 是 Publisher 的系统名称。  
publisher\_db 是 Publisher 数据库的系统名称。

```
[ @publication = ] publication  
[ @property = ] property
```

publication 是出版物的系统名称。  
property 是下列变量要修改的特征：  
publisher\_login , publisher\_password ,  
publisher\_security\_mode(其值为：  
0=SQL Server Authentication ,  
1=Windows NTAuthentication) ,  
encrypted\_publisher\_password ,  
distributor\_login ,  
distributor\_password ,  
distributor\_security\_mode(其值为：  
0=SQL Server Authentication ,  
1=Windows NT Authentication) ,  
encrypted\_distributor\_password。

```
[ @value = ] value
```

value 是 nvarchar(255)型数据，包含

属性值。

## sp\_change\_users\_login

sp\_change\_users\_login 系统存储过程可用于修改用户登录信息。

### 语法

```
sp_change_users_login [ @Action = ]          action          [ , [ @UserNamePattern  
= ]          user  
[ , [ @LoginName = ]          login          ]]
```

### 变量

[ @Action = ] action

action 是 varchar(10)型数据，是要执行的操作。Auto\_Fix=更新数据库用户(sysusers 系统表)，使对应于注册表(syslogins 系统表)中的同名登录。提供 UserNamePattern，但不提供 LoginName。Report=报告孤立且无相应登录的数据库用户。不为该报告提供 LoginName 或 UserNamePattern。Update\_One=提供 LoginName 和 UserNamePattern，当前数据库的 sysusers 表中的 UserName 将被更新为所提供的

```
[ @UserNamePattern = ] user
[ @LoginName = ] login
```

LoginName。  
user 是数据库用户的系统名称。  
login 是存在于 Microsoft SQL Server 上 syslogins 列表中的 SQL Server 登录的系统名称。

## sp\_column\_privileges

sp\_column\_privileges 系统存储过程可显示当前数据库中表的列特权。

### 语法

```
sp_column_privileges { table_name } [ , table_owner ] [ ,
table_qualifier ]
[ , column ]
```

### 变量

table\_name  
table\_owner  
table\_qualifier

column

table\_name 是表的系统名称。  
table\_owner 是表拥有者的系统名称。  
table\_qualifier 是 SQL Server 中数据库的系统名称。它可以用不同的源数据集名代表其他产品中的其他事物。  
column 是列的 nvarchar(384)型名称。  
如果省略该变量，则返回所有的列。

这是该系统存储程序中唯一允许使用通配符的变量。

## sp\_column\_privileges\_ex

sp\_column\_privileges\_ex 系统存储过程可显示链接服务器上某一表的列特权。

### 语法

```
sp_column_privileges_ex { table_server } [ , table_name ]  
[ , table_schema ]  
[ , table_catalog ] , column_name
```

### 变量

table\_server

table\_server 是链接服务器的系统名称。

table\_name

table\_name 是表格的系统名称。

table\_schema

table\_schema 是表格模式的系统名称，缺省值为 NULL。

table\_catalog

table\_catalog 是数据库的系统名称，缺省值为 NULL。

column\_name

column\_name 是列的 nvarchar(384) 型名称。



## sp\_columns

sp\_columns 系统存储可显示当前数据库中表格或视图的类别信息。

### 语法

```
sp_columns {object} [ ,owner ][ , qualifier ][ , column ][ , ODBCVer ]
```

### 变量

object	object 是表格或视图的 nvarchar(384)型名称。
Owner	owner 是表格或视图的拥有者的 nvarchar(384)型名称。
Qualifier	qualifier 是 SQL Server 中数据库的系统名称。它可以用不同的源数据集名代表其他产品的其他事物。
Column	column 是列的 nvarchar(384)型名称。
ODBCVer	ODBCVer 是一个代表当前 ODBC 版本的整数，其值可以是 2 或 3，缺省值为 2。

## sp\_columns\_ex

sp\_columns\_ex 系统存储过程可显示列的类别信息。

### 语法

```
sp_columns_ex { table_server } [ , table_name ][ , table_schema
```

]  
 [ , table\_catalog ][ , column ]  
 变量

table_server	table_server 是链接服务器的系统名称。
table_name	table_name 是表格的系统名称。
table_schema	table_schema 是表格模式的系统名称，缺省值为 NULL。
table_catalog	table_catalog 是数据库的系统名称，缺省值为 NULL。
column	column 是列的 nvarchar(384) 型名称。

## sp\_configure

sp\_configure 系统存储过程可显示或改变当前服务器的全局配置选项。参看第 10 章“简化数据库管理员的角色。”

### 语法

sp\_configure [ name ][ , value ]  
 变量

name	name 是 varchar(35)型数据，是配置选项，如果它没有在另一个选项中复制，就可以是名称子集的任一字符串。如未提供，则返回所有可用的选项。
value	value 是一个整数，是分配给配置选项的值，对于大多数配置选项，0=关闭或 FALSE，1=打开或 TRUE。

## sp\_createstats

sp\_createstats 系统存储过程可运行 UPDATE STATISTICS WITH COLUMNS 语句，而不包括位，文本，图像或者数据戳数据类型的列，以及带有统计信息的列。

### 语法

```
sp_createstats [ @indexonly = ] indexonly , [ @fullscan = ] fullscan
```

### 变量

[ @indexonly = ] indexonly indexonly 是字符(9)型数据，缺省值为 NO。提供该变量可将那些由 UPDATE STATISTICS WITH COLUMNS 标志的列进一步限定为只经过索引的列。

[ @fullscan = ] fullscan fullscan 是一个字符(9)型数据，如果要扫描整个表格，就必须提供该变量，否则，就只扫描样本。

## sp\_create\_removable

sp\_create\_removable 系统存储过程用以创建可拆卸介质数据库，该数据库可以放在 CD-ROM 上或其他可拆卸介质上。该系统存储过程可根据数据量，将数据库放在包含系统表的文件上，包含事务日志的文件上，以及 1 到 16 个数据文件上。该系统存储过程还可用来向另一个地点发送数据库。

### 语法

```
sp_create_removable [ @dbname = ]          dbname      ,
[ @syslogical = ]      syslogical      ,
[ @sysphysical = ]     sysphysical      ,
[ @sysssize = ]        syssize,
[ @loglogical = ]      loglogical      ,
[ @logphysical = ]     logphysical      ,
[ @logsize = ]         logsize,
[ @datalogical = ]     datalogical      ,
[ @dataphysical = ]    datalogical      ,
[ @datasize1 = ]       datasize1
[ ..., [ @datalogical16 = ]      datalogical16      ,
[ @dataphysical16 = ]      dataphysical16      ,
[ @datasize16 = ]       datasize16      ]
```

### 变量

dbname	dbname 是数据库的系统名称。
syslogical	syslogical 是将创建以存放系统表的文件的逻辑名 (系统名称数据类型)。
sysphysical	sysphysical 是将创建以存放系统表的文件的物理名称 (路径 / 文件格式的 nvarchar(260)型数据)。
Syssize	syssize 是一个整数，代表由该系统存储过程创建以存放系统表的文件的大小。该变量以 MB 为单位，最小值为 1MB。
loglogical	loglogical 是将创建以存放事物日志的文件的逻辑名称 (系统名称数据类型)。
logphysical	logphysical 是将创建以存放事务日志的文件的物理名称 (路径 / 文件格式的 nvarchar(260)型数据)。
Logsize	logsize 是一个整数，代表由该系统存储过程创建以存放事务日志的文件的大小。该变量以 MB 为单位，最小值为 1MB。
datalogical	datalogical 是将创建以存放数据表的文件的逻辑名称 (系统名称数据类型)。对大型数据库，最多可以有 16 个数据文件。
dataphysical	dataphysical 是将创建以存放数据表的文件的物理名称 (路径 / 文件格式的 nvarchar(260)型数据)。
Datasize	datasize 是一个整数，代表由该系统存储过程创建

以存放数据表的文件的大小。该变量以 MB 为单位，最小值为 1MB。
------------------------------------

## sp\_cursor\_list

sp\_cursor\_list 系统存储过程可滚动返回并更新用于连接的打开服务器游标的信息。

### 语法

```
sp_cursor_list { [ @cursor_return = ] cursor_variable_name OUTPUT }  
{, [ @cursor_scope = ] cursor_scope }
```

### 变量

cursor_variable_name	cursor_variable_name 是现有的游标变量的声明名称。
cursor_scope	cursor_scope 是一个整数，其中 1=局部游标，2=全局游标，3=局部和全局游标。

## sp\_cycle\_errorlog

sp\_cycle\_errorlog 系统存储过程可关闭当前的错误日志文件，并打开新的错误日志文件。其运行过程类似于关闭再重新启动 SQL Server。

## 语法

`sp_cycle_errorlog`

`sp_databases`

`sp_databases` 系统存储过程可通过数据库网关返回服务器上的 Microsoft SQL Server 数据库，以及可用的数据库。

## 语法

`sp_databases`

`sp_datatype_info`

如果没有给 `sp_datatype` 系统存储过程传递变量，该存储过程可列出在当前 Microsoft SQL Server 上创建的数据类型。也可以将数据类型作为变量传递给它。

## 语法

`sp_datatype_info` [ `data_type` ] [ , `odbc_version` ]

## 变量

`data_type`

`data_type` 是一个整数值，该值可以通过运行不带参数的同一个系统存储过程（`sp_datatype_info`）并检查输出列 `data_type` 中的值而获得。

`odbc_version` `odbc_version` 是正使用的 ODBC 版本，它是 `tinyint` 型数据，缺省值为 2。

## `sp_dbcmptlevel`

`sp_dbcmptlevel` 系统存储过程可决定服务器如何与 Microsoft SQL Server 的某个早期版本兼容。

### 语法

```
sp_dbcmptlevel [[ @dbname = ] name ] [ , [ @new_cmptlevel = ] version ]
```

### 变量

[ @dbname = ] name

dbname 是数据库的系统名称。

[ @new\_cmptlevel = ] version

version 是一个整数，是数据库的兼容级别。其值可以是 60，65 或 70。

## `sp_dbfixedrolepermission`

`sp_dbfixedrolepermission` 系统存储过程可返回所有固定数据库角色或选定的角色的许可。

### 语法

```
sp_dbfixedrolepermission [[ @rolename = ] role ]
```



变量 `role` 是固定数据库角色的名称。如果没有提供该变量，则返回所有角色的许可。

## `sp_dbopiton`

`sp_dboption` 系统数据库可列出所有的数据库选项，或变更一个选项。

语法

```
sp_dboption [ database ] [ , option_name ] [ , value ]
```

变量

`database` `database` 是数据库的系统名称，如果没有提供该变量，则列出所有可更改的数据库选项。

`option_name` `option_name` 是要更改的数据库选项的 `varchar(35)`型名称。如果没有提供该变量，而只提供了 `database` 变量，则返回当前数据库的选项。

`value` `value` 是数据库选项的 `varchar(10)`型新值。

## `sp_dbremove`

`sp_dbremove` 系统存储过程可用于向后兼容。Microsoft SQL Server 7 版本的

程序员可用 `sp_detach_db` 系统存储过程从 Microsoft SQL Server 中删除一个可拆卸介质数据库。`sp_dbremove` 系统存储过程可删除一个数据库及其文件。

#### 语法

```
sp_dbremove [ @dbname = ]          database          [ , [ @dropdev = ]          dropdev  
          ]
```

#### 变量

[ @dbname = ]          database	database 是数据库的系统名。
[ @dropdev = ]          dropdev	dropdev 是一个标志，其值为 dropdev，该变量在 7.0 版本中被忽略。

## sp\_defaultdb

`sp_defaultdb` 系统存储过程可为登录创建新的缺省数据库。只有在存在 `guest` 用户的情况下，运行该存储过程才会给登录提供缺省数据库的访问权，否则，`dbo` 必须运行 `sp_grantdbaccess` 系统存储过程，明确给登录提供数据库的访问权。

#### 语法

```
sp_defaultdb [ @loginame = ]        login            [ , [ @defdb = ]          database  
          ]
```

#### 变量

[ @loginame = ] login login 是登录的系统名称。如果登录是

```
[ @defdb = ] database
```

一个尚不存在的 Windows NT 用户或用户组，它将被增加进来。

database 是一个现有数据库的系统名称，该数据库将成为用户的缺省数据库。

## sp\_defaultlanguage

sp\_defaultlanguage 系统存储过程可改变登录的缺省语言。

语法

```
sp_defaultlanguage [ @loginame = ] login [ , [ @language = ]  
language ]
```

变量

```
[ @loginame = ] login
```

login 是登录的系统名称。若登录是一个尚不存在的 Windows NT 用户或用户组，它将被添加进来。

```
[ @language = ] language
```

language 是现有语言的系统名称，该语言将成为用户的缺省语言，使用系统存储过程 sp\_helplanguage 将得到可用的语言。

## sp\_delete\_alert

sp\_delete\_alert 系统存储过程可在 msdb 系统数据库中运行时删除 SQL Server 警告。

### 语法

```
sp_delete_alert { name }
```

变量 name 是要删除的警告的 varchar(100)型名称。

## sp\_delete\_category

sp\_delete\_category 系统存储过程可从服务器中删除一个类别。

### 语法

```
sp_delete_category { class } [ , name ]
```

### 变量

class	class 是要删除的 varchar(8)型类别。该变量可能的值为 JOB, ALERT, 及 OPERATOR。注意, 删除警告的同时会删除与类别有成员关系的作业、类别或操作。
-------	--

name	name 是服务器的系统名称。
------	-----------------

## sp\_delete\_job

sp\_delete\_job 系统存储过程可从服务器中删除一项工作。需提供 job\_id 或 job\_name 变量。

### 语法

```
sp_delete_job { [ @job_id = ]      job_id      | [ @job_name = ]      job_name
              }
[ , [ @originating_server = ]      server      ] [ , [ @delete_history = ]
delete_history ]
```

### 变量

job\_id

job\_id 是作业的唯一标识符。

job\_name

job\_name 是要删除的作业的 varchar(100) 型名称。

server

server 是产生该作业的服务器的 varchar(30) 型名称。

delete\_history

delete\_history 是位标志，其中，0=false，1=true，缺省值为 1。

## sp\_delete\_jobschedule

sp\_delete\_jobschedule 系统存储过程可在服务器上运行安排好的工作中删除一项，需提供 job\_id 或 job\_name 变量。

### 语法

```
sp_delete_jobschedule { [ @job_id = ]          job_id      | [ @job_name = ]  
job_name      }  
{, [ @name = ]          sched_job_name      }
```

### 变量

job\_id job\_id 是要从工作安排表中删除的工作的唯一标识符。

job\_name job\_name 是要从安排表中删除的工作的 varchar(100)型名称。

sched\_job\_name sched\_job\_name 是安排表中工作的 varchar(100)型名称。

## sp\_delete\_jobserver

sp\_delete\_jobserver 系统存储过程可从用作某一项工作的工作服务器中删除一个服务器。

## 语法

```
sp_delete_jobserver { job_id | job_name } {, server }
```

## 变量

job_id	job_id 是要从工作服务器中删除的工作的唯一标识符。
job_name	job_name 是要从工作服务器中删除的工作的 varchar(100)型名称。
server	server 是作为工作服务器被删除的服务器的 varchar(30)型名称。

## sp\_delete\_jobstep

sp\_delete\_jobstep 系统存储过程可从一项工作中删除一个工作步骤，需提供 job\_id 或 job\_name 变量。

## 语法

```
sp_delete_jobstep { [ job_id = ] job_id | [ @job_name = ] job_name }
```

```
{, [ @step_id = ] step_id }
```

## 变量

job_id	job_id 是包含要从工作服务器上删除的工作步骤的工作的唯一标识符。
job_name	job_name 是包含要从工作服务器上删

除的工作步骤的工作的 varchar(100)  
型名称。

step\_id

step\_id 是代表要删除的步骤的整数。

## sp\_delete\_notification

sp\_delete\_notification 系统存储过程可删除一个通过警告发送给操作员的通  
知。

语法

```
sp_delete_notification { alert , operator }
```

变量

alert

alter 是警告的 varchar(100)型名称。

operator

operator 是操作员的 varchar(100)型名称。

## sp\_delete\_operator

sp\_delete\_operator 系统存储过程可删除一个操作员。

语法

```
sp_delete_operator { name } [ , reassign_operator ]
```

变量



name	name 是要删除的操作员的 varchar(100)型名称。
reassign_operator	reassign_operator 是 varchar(100)型数据，允许将删除的操作员的警告重新分配给该操作员。

## sp\_delete\_targetserver

sp\_delete\_targetserver 系统存储过程可从工作服务器(目标服务器)列表中删除服务器。

### 语法

```
sp_delete_targetserver { server } [ , clear_downloadlist ] [ , post_defection ]
```

### 变量

server	server 是要作为目标服务器删除的服务器的 varchar(100)型名称。
clear_downloadlist	被保留用于 SQL Server。
post_defection	被保留用于 SQL Server。

## sp\_delete\_targetservergroup

sp\_delete\_targetservergroup 系统存储过程可删除一个目标服务器组。

## 语法

```
sp_delete_targetservergroup { name }
```

变量 `name` 是要删除的目标服务器组的 `varchar(100)`型名称。

```
sp_delete_targetsvrgrp_member
```

`sp_delete_targetsvrgrp_member` 可从目标服务器组中删除服务器。

## 语法

```
sp_delete_targetsvrgrp_member { group_name } [ , server_name ]
```

## 变量

<code>group_name</code>	<code>group_name</code> 是要删除服务器的目标服务器组的 <code>varchar(100)</code> 型名称。
<code>server_name</code>	<code>server_name</code> 是要从目标服务器组中删除的服务器的 <code>varchar(30)</code> 型名称。

```
sp_deletemergeconflictrow
```

`sp_deletemergeconflictrow` 系统存储过程可从合并复制中删除冲突的行，该行位于 `msmerge_delete_conflicts` 系统表中。

## 语法

```
sp_deletemergeconflictrow [[ @conflict_table = ] conflict_table ]  
[ , [ @source_object = ] source_object ] { , [ @rowguid = ] rowguid  
  , [ @origin_datasource = ] origin_datasource ] }  
[ , [ @drop_table_if_empty = ] drop_table_if_empty ]
```

## 变量

```
[ @conflict_table = ]  
conflict_table
```

conflict\_table 是冲突表的系统名称。可以将该变量设为 NULL，用行指定删除的冲突，其中，变量 rowguid，origin\_datasource 和 source\_object 的值与使用该系统存储过程从系统列表 msmerge\_delete\_conflicts 中删除的变量值相等。缺省值 % 代表所有的冲突表。

```
[ @source_object = ]  
source_object  
[ @rowguid = ] rowguid
```

source\_object 是源表的 nvarchar(386) 型名称。  
rowguid 是删除冲突的行的唯一标识符。

```
[ @origin_datasource = ]  
origin_datasource  
[ @drop_table_if_empty = ]  
drop_table_if_empty
```

origin\_datasource 是 varchar(255) 型数据，代表冲突的来源。  
drop\_table\_if\_empty 是 nvarchar(10) 型数据，如果该变量设

为 true，若冲突表不包含行，则撤消该表。缺省为 false。

## sp\_denylogin

sp\_denylogin 系统存储过程可阻止登录访问当前的 SQL Server。

语法

```
sp_denylogin [ @loginame = ] login
```

变量 login 的格式是域 / 用户，是 Windows NT 用户或组的系统名称。如果它不存在，则添加它。

## sp\_depends

sp\_depends 系统存储过程可返回对象从属信息。

语法

```
sp_depends { object }
```

变量 object 是要返回从属信息的对象。

## sp\_describe\_cursor

sp\_describe\_cursor 系统存储过程可返回服务器游标的信息。

## 语法

```
sp_describe_cursor { [ @cursor_return = ]      cursor_variable_name      |  
OUTPUT}  
{  
  [ , [ @cursor_source = ]      local      , [ @cursor_identity = ]  
      local_cursor_name      ] |  
  [ , [ @cursor_source = ]      global      , [ @cursor_identity = ]  
      global_cursor_name      ] |  
  [ , [ @cursor_source = ]      variable      , [ @cursor_identity = ]  
      cursor_variable_name      ]  
}
```

## 变量

cursor_variable_name	cursor_variable_name 是现有游标变量的名称。
local_cursor_name	local_cursor_name 是局部游标的名称。
global_cursor_name	global_cursor_name 是全局游标的名称。

## sp\_describe\_cursor\_columns

sp\_describe\_cursor\_columns 系统存储过程可在服务器游标的结果集中输出列

的属性。

#### 语法

```
sp_describe_cursor_columns { [ @cursor_return = ] cursor_variable_name
    OUTPUT}
{
    [ , [ @cursor_source = ] local , [ @cursor_identity = ]
    local_cursor_name ] | [ , [ @cursor_source = ] global ,
    [ @cursor_identity = ] global_cursor_name ] |
    [ , [ @cursor_source = ] variable , [ @cursor_identity = ]
    cursor_variable_name ]
}
```

#### 变量

cursor_variable_name	cursor_variable_name 是现有游标变量的名称。
local_cursor_name	local_cursor_name 是局部游标的名称。
global_cursor_name	global_cursor_name 是全局游标的名称。

## sp\_describe\_cursor\_tables

sp\_describe\_cursor\_tables 系统存储过程可返回服务器游标的基础表信息。

#### 语法

```
sp_describe_cursor_tables { [ @cursor_return = ] cursor_variable_name
```

```

OUTPUT}
{
[ , [ @cursor_source = ] local , [ @cursor_identity = ]
local_cursor_name ] | [ , [ @cursor_source = ] global ,
[ @cursor_identity = ] global_cursor_name ] | [ , [ @cursor_source
= ] variable , [ @cursor_identity = ]
cursor_variable_name ]
}

```

## 变量

cursor_variable_name	cursor_variable_name 是现有游标变量的名称。
local_cursor_name	local_cursor_name 是局部游标的名称。
global_cursor_name	global_cursor_name 是全局游标的名称。

## sp\_detach\_db

sp\_detach\_db 系统存储过程可从 SQL Server 上分离一个数据库，并可以在分离数据库之前，运行 UPDATE STATISTICS 语句。

## 语法

```
sp_detach_db [ @dbname = ]      dbname  
[ , [ @skipchecks = ]    skipchecks ]  
    变量
```

```
[ @dbname = ]      dbname  
[ @skipchecks = ]    skipchecks
```

dbname 是数据库的系统名称。  
skipchecks 是 nvarchar(10)型数据。  
TRUE = 不运行 UPDATE STATISTICS ,  
FALSE = 运行 UPDATE STATISTICS。

## sp\_distcounters

sp\_distcounters 系统存储过程可返回 Subscriber 发布信息。该信息也可以通过 Performance Monitor 获得。

语法

```
sp_distcounters
```

## sp\_drop\_agent\_parameter

sp\_drop\_agent\_parameter 系统存储过程可从复制代理中删除配置文件所使用的参数。

语法

```
sp_drop_agent_parameter [ @profile_id = ]      profile_id ,
```



[ @parameter\_name = ] parameter\_name  
变量

[ @profile\_id = ] profile\_id profile\_id 是一个整数，是要撤消的带参数的配置文件。

[ @parameter\_name = ] parameter\_name parameter\_name 是要撤消的参数的系统名称。% 表示全部参数。

## sp\_drop\_agent\_profile

sp\_drop\_agent\_profile 系统存储过程可撤消复制代理的配置文件。  
语法

sp\_MSdrop\_agent\_profile [ @profile\_id = ] profile\_id  
变量 profile 是配置文件的整数 ID。

## sp\_dropalias

sp\_dropalias 系统存储只向后兼容。Microsoft SQL Server 7.0 程序员可使用角色。

语法

sp\_dropalias [ @loginame = ] login

变量 login 是要撤消别名的注册或 Windows NT 用户或组。

## sp\_dropapprole

sp\_dropapprole 系统存储过程可撤消当前数据库中的应用角色。

语法

```
sp_dropapprole [ @rolename = ]      role
```

变量 role 是要撤消的应用角色的系统名称。

## sp\_droparticle

sp\_droparticle 系统存储过程可删除未订阅的文章。此程序用在复制环境中以修改出版物。

语法

```
sp_droparticle [ @publication = ]    publication      , [ @article = ]  
article
```

```
[ , [ @ignore_distributor = ]      ignore_distributor  ]
```

变量

[ @publication = ] publication publication 是出版物的系统名称。

[ @article = ] article artical 是文章的系统名称。可以使

```
[ @ignore_distributor = ]
ignore_distributor
```

用 “ all ” (全部)。

ignore\_distributor 指示器是个位型数据，缺省值为 0 或 false。如果想忽略发布器上的清理任务，可以使用该变量。

## sp\_dropdevice

sp\_dropdevice 系统存储过程可撤消数据库设备。

语法

```
sp_dropdevice [ @logicalname = ] device
[ , [ @delfile = ] delfile ]
```

变量

```
[ @logicalname = ] device
```

```
[ @delfile = ] delfile
```

device 是数据库设备的逻辑名称。

delfile 的值可删除物理备份文件 (varchar(7)型数据)。如果忽略该变量，则关闭物理文件，但不删除。

## sp\_dropdistpublisher

sp\_dropdistpublisher 系统存储过程可删除 Publisher, Publisher 也是它自己的 Distributor。

### 语法

```
sp_dropdistpublisher [ @publisher = ] publisher [ , [ @no_checks = ]  
no_checks ]
```

### 变量

[ @publisher = ] publisher	publisher 是 Publisher 的系统名称, 也可以使用 “ all ”。
[ @no_checks = ] no_checks	no_checks 指示符是位型数据, 其缺省值为 0。

## sp\_dropdistributiondb

sp\_dropdistributiondb 系统存储过程可删除 Distributor 数据库和相关的数据库文件。

### 语法

```
sp_dropdistributiondb [ @database = ] database
```

变量 database 是数据库的系统名称。

## sp\_dropdistributor

sp\_dropdistributor 系统存储过程从注册为 Distributor 的服务器 NT 注册中撤消 Distributor。

### 语法

```
sp_dropdistributor [ @no_checks = ] no_checks ]  
[ , [ @ignore_distributor = ] ignore_distributor  
变量
```

[ @no_checks = ] no_checks	no_checks 指示符是一个位型数据，缺省值为 0。
[ @ignore_distributor = ] ignore_distributor	ignore_distributor 指示符是一个位型数据，缺省值为 0 或 false。如果要忽略 Distributor 中的清理任务，可以使用该变量。

## sp\_dropextendedproc

sp\_dropextendedproc 系统存储过程可删除一个扩展存储过程。

### 语法

```
sp_dropextendedproc{ procedure }
```

变量 procedure 是扩展存储过程的 nvarchar(257)型名称。

## sp\_dropgroup

sp\_dropgroup 系统存储过程可从当前数据库中撤消一个角色。此存储过程只向后兼容。Microsoft SQL Server 7 的程序员可用角色代替组。

### 语法

```
sp_dropgroup [ @rolename = ] role
```

变量 role 是要撤消的角色的系统名。

## sp\_droplinkedsrvlogin

sp\_droplinkedsrvlogin 系统存储过程可撤消本地 SQL Server 上的注册与映射到它上面的链接服务器注册间的连接。

### 语法

```
sp_droplinkedsrvlogin [ @rmtsrvname = ] rmtsrvname ,
```

```
[ @locallogin = ] locallogin
```

### 变量

[ @rmtsrvname = ] rmtsrvname	rmtsrvname 是链接服务器的系统名称。
[ @locallogin = ] locallogin	locallogin 是本地服务器上的 SQL Server 注册的系统名称。

## sp\_droplogin

sp\_droplogin 系统存储过程可撤消 Microsoft SQL Server 注册。

语法

```
sp_droplogin [ @loginame = ] login
```

变量 login 是要撤消的 SQL Server 注册的系统名称。

## sp\_dropmergearticle

sp\_dropmergearticle 系统存储过程可撤消合并出版文章。

语法

```
sp_dropmergearticle [ @publication = ] publication , [ @article = ]  
article
```

```
[ , [ @ignore_distributor = ] ignore_distributor
```

变量

[ @publication = ] publication publication 为出版物的系统名称，可以使用“all”。

[ @article = ] article article 为文章的系统名称。可以使用“all”。

[ @ignore\_distributor = ] ignore\_distributor 指示符是一个位型

ignore\_distributor

数据，缺省值为 0 或 false。如果要忽略 Distributor 上的清理任务，可以使用该变量。

## sp\_dropmergefilter

sp\_dropmergefilter 系统存储过程可删除合并过滤器。

语法

```
sp_dropmergefilter [ @publication = ]      publication      [ , [ @article = ]  
      article  
[ , [ @filtername = ]      filtername      ]
```

变量

[ @publication = ] publication publication 是出版物的系统名称。

[ @article = ] article

article 是文章的系统名称。

[ @filtername = ] filtername

filter 是过滤器的系统名称。

## sp\_dropmergepublication

sp\_dropmergepublication 系统存储过程可撤消出版物及与其有关的 Snapshot Agent。



## 语法

```
sp_dropmergepublication [ @publication = ]      publication
[ , [ @ignore_distributor = ]      ignore_distributor ]
    变量
```

[ @publication = ] publication publication 为出版物的系统名称。可以使用“all”。

[ @ignore\_distributor = ] ignore\_distributor 指示符是一个位型数据，缺省值为 0 或 false。如果要忽略 Distributor 上的清理任务，可以使用该变量。

## sp\_dropmergepullsubscription

sp\_dropmergepullsubscription 系统存储过程可删除合并推出订阅。在 Subscriber 上执行，其中当前数据库是 Subscriber 数据库。

### 语法

```
sp_dropmergepullsubscription [[ @publication = ]      publication      ]
[ , [ @publisher = ]      publisher      ] [ , [ @publisher_db = ]
publisher_db      ]
    变量
```

[ @publication = ] publication publication 为出版物的系统名称，

```
[ @publisher = ] publisher
```

可以使用“all”。

publisher 是 Publisher 的系统名称。

```
[ @publisher_db = ]  
publisher_db
```

publisher\_db 为 Publisher 数据库的系统名称。

## sp\_dropmergesubscription

sp\_dropmergesubscription 系统存储过程可从其 Merge Agent 中删除合并出版订阅。

### 语法

```
sp_dropmergesubscription [ [ @publication = ] publication ]  
[ , [ @subscriber = ] subscriber [ , [ @subscriber_db = ]  
subscriber_db ]  
[ , [ @subscription_type = ] subscription_type ]  
[ , [ @ignore_distributor = ] ignore_distributor ]
```

### 变量

```
[ @publication = ] publication
```

publication 是出版物的系统名称，可以使用“all”。

```
[ @subscriber = ] subscriber
```

subscriber 是 Subscriber 的系统名称。

```
[ @subscriber_db = ]
```

subscriber\_db 是 Subscriber 数据库的

subscriber\_db  
[ @subscription\_type = ]  
subscription\_type

系统名称。

subscription\_type 是 nvarchar(15)型数据，其中 Push=推入订阅(缺省值)，Pull=推出订阅，Both=推入订阅与推出订阅。

[ @ignore\_distributor = ]  
ignore\_distributor

ignore\_distributor 指示符是一个位型数据，缺省值为 0 或 false。如果要忽略 Distributor 上的清理任务，可以使用该变量。

## sp\_dropmessage

sp\_dropmessage 系统存储过程可从主数据库的 sysmessages 系统表中删除错误信息。

### 语法

sp\_dropmessage [ message\_number ] [ , language ]

### 变量

message\_number

用户为要删除的信息建立的整数。

language

language 是与信息有关的语言的系统名称。

## sp\_droppublication

sp\_droppublication 系统存储过程可删除一个出版物及其所有文章。同时也删除其同步任务。

### 语法

```
sp_droppublication [ @publication = ]      publication  
[ , [ @ignore_distributor = ]      ignore_distributor ]
```

### 变量

[ @publication = ]      publication	要删除的出版物的系统名称，可以使用“all”。
[      @ignore_distributor      =      ] ignore_distributor	ignore_distributor 指示符是一个位型数据，缺省值为 0 或 false。如果要忽略 Distributor 上的清理任务，可以使用该变量。

## sp\_droppullsubscription

sp\_droppullsubscription 系统存储过程可撤消 Subscriber 的当前数据库中的订阅。该程序也可删除 Subscriber 中的 Distributor Agent。

### 语法

```

sp_droppullsubscription [ @publisher = ] publisher ,
[ @publisher_db = ] publisher_db , [ @publication = ]
publication

```

```

[ , [ @reserved = ] reserved ]

```

### 变量

[ @publisher = ] publisher publisher 是 Publisher 的系统名称。

[ @publisher\_db = ] publisher\_db publisher\_db 是 Publisher 数据库的系统名称。可以使用“all”。

[ @publication = ] publication publication 为出版物的系统名称，可以使用“all”。

[ @reserved = ] reserved reserved 是 nvarchar(8)型数据，是为 SQL Server 保留的。

## sp\_dropremotelogin

sp\_dropremotelogin 系统存储过程可从本地 SQL Server 中删除远程用户的注册。Microsoft SQL Server 7.0 程序员可用链接服务器代替远程服务器。

### 语法

```

sp_dropremotelogin [ @remoteserver = ] remoteserver [ , [ @loginame = ]
login

```

```

[ , [ @remotename = ] remote_name ]]

```

## 变量

[ @remoteserver = ]	remoteserver	remoteserver 为远程服务器的系统名称。若只提供这一个参数，将从本地服务器上删除远程服务器的所有远程注册。
[ @loginame = ]	login	login 是与远程服务器注册名称有关的本地服务器注册的系统名称（可选项）。
[ @remotename = ]	remote_name	remote_name 是映射到本地注册的远程注册的系统名称（可选项）。

## sp\_droprole

sp\_droprole 系统存储过程可从当前使用的数据库中撤消 Microsoft SQL Server 角色。

### 语法

sp\_droprole [ @rolename = ] role

变量 role 是要删除的角色的系统名称。

## sp\_droprolemember

sp\_droprolemember 系统存储过程从当前数据库的 SQL Server 角色中撤消 SQL Server 注册、角色、NT 用户或组(这些也称作安全帐户)。

### 语法

```
sp_droprolemember [ @rolename = ] role ,  
[ @membername = ] security_account
```

### 变量

role	role 是角色的系统名称。
security_account	security_account 是作为角色中的成员而删除的安全帐户的系统名称。

## sp\_dropserver

sp\_dropserver 系统存储过程可撤消在本地服务器上用作远程或链接服务器的服务器。

### 语法

```
sp_dropserver [ @server = ] server [ , { [ @droplogins = ] droplogins  
| NULL } ]
```

### 变量

```
[ @server = ] server
[ @droplogins = ] droplogins
NULL
```

server 是服务器的系统名称。  
droplogins 是字符(10)型数据，当其值为 droplogins 时，通知 SQL Server 也将删除与已撤消的服务器有关的远程注册。其值为 NULL 时表示不撤消远程注册。

## sp\_drop\_srvrolemember

sp\_drop\_srvrolemember 系统存储过程可从固定的服务器角色中撤消 SQL Server 注册、Windows NT 用户或 Windows NT 组。

### 语法

```
sp_drop_srvrolemember [ @loginame = ] login [ , [ @rolename = ]
role ]
```

### 变量

```
[ @loginame = ] login
```

login 是要从角色的成员关系中删除的 SQL Server 注册、Windows NT 用户或 Windows NT 组的系统名称。

```
[ @rolename = ] role
```

role 是服务器角色的系统名称，它可以是 NULL。角色必须是有效的固定服务器角色，可以是 sysadmin, securityadmin, serveradm



in,setupadmin,processadmin,diskadmin,dbcreator。

## sp\_dropsubscriber

sp\_dropsubscriber 系统存储过程将从注册为 Subscriber 的服务器 NT 注册中删除 Subscriber，该程序也可取消为 repl\_subscriber 注册映射的 sa(系统管理员)注册。

### 语法

```
sp_dropsubscriber [ @subscriber = ] subscriber [ , [ @reserved = ] reserved ]  
[ , [ @ignore_distributor = ] ignore_distributor ]
```

### 变量

[ @subscriber = ] subscriber

subscriber 是 Subscriber 的系统名称。

[ @reserved = ] reserved

为 SQL Server 保留的变量。

[ @ignore\_distributor = ]

ignore\_distributor

ignore\_distributor 指示符是一个位型数据，缺省值为 0 或 false。如果要忽略 Distributor 上的清理任务，可以使用该变量。

## sp\_dropsubscription

sp\_dropsubscription 系统存储过程用于 Publisher 上，以删除对出版物、文章或订阅集的订阅。如果在没有撤消出版物中所有文章上订阅的情况下要撤消该出版物，并读取所有文章，则将不能读取 immediate\_sync 出版物中某篇文章上的订阅。

### 语法

```
sp_dropsubscription [[ @publication = ]      publication      [, [ @article  
= ]      article      ]  
{, [ @subscriber = ]      subscriber      } [, [ @destination_db = ]  
destination_db      ]  
[, [ @ignore_distributor = ]      ignore_distributor      ][, [ @reserved = ]  
reserved      ]
```

### 变量

[ @publication = ] publication

publication 是出版物的系统名称，使用“all”（全部）将删除 Subscriber 的所有出版物的全部预订。

[ @article = ] article

article 是文章的系统名称。用“全部”可删除 immediate\_sync 出版物中 Publication 和 Subscriber 的所有文章

的所有预订。如果没有提供此变量，将撤消出版物中所有文章的所有预订。

[ @ subscriber = ] subscriber

subscriber 是 Subscriber 的系统名称，用“all”将撤消所有 Subscriber 的所有预定。

[ @ destination\_db = ]  
destination\_db

destination\_db 是目标数据库的系统名称。如果此变量的值为 NULL，将撤消指定 Subscriber 的所有预定。

[ @ ignore\_distributor = ]  
ignore\_distributor

ignore\_distributor 指示符是一个位型数据，缺省值为 0 或 false。如果要忽略 Distributor 的清理任务，可以使用该变量。

[ @ reserved = ] reserved

reserved 是 nvarchar(10)型数据，是为 SQL Server 保留的。

## sp\_droptask

sp\_droptask 系统存储过程可删除已安排的任务，它只向后兼容。SQL Server 7 程序员可用作作业代替任务。

## sp\_droptype

sp\_droptype 系统存储过程可撤消用户建立的数据类型。

语法

```
sp_droptype { type }
```

变量 type 是要撤消的用户建立的数据类型名。

## sp\_dropuser

sp\_dropuser 系统存储过程只向后兼容，可从当前使用的数据库中撤消 SQL Server 用户、Windows NT 用户或 Windows NT 组。Microsoft SQL Server 程序员可用 sp\_revokedbaccess 代替该程序。

语法

```
sp_dropuser [ @name_in_db = ] user
```

变量 user 是要撤消的用户的系统名称。

## sp\_dropwebtask

sp\_dropwebtask 系统存储过程可撤消已有的 Web 任务。需提供一个或两个变量。

## 语法

```
sp_dropwebtask [[ @procname = ]      procname      | NULL ][ ,      outputfile  
      ]
```

## 变量

procname                    procname 是为放置 Web 任务查询的存储过程名。  
outputfile                 outputfile 是要删除的 HTML 输出文件名。

## sp\_dsninfo

sp\_dsninfo 系统存储过程可显示 ODBC 数据源名称的信息，这些信息源自本地服务器中用户建立的复制 Distributor。该程序展示了可用于在网络上查询的数据库和可供复制的数据库。信息包括来自其他数据库厂商的数据库和数据源以及不同文件类型的数据库和数据源。

## 语法

```
sp_dsninfo {      dsn      } [ ,      info_type      ] [ ,      login      ] [ ,  
      password      ]
```

## 变量

dsn                                    dsn 是所需要的 varchar(128) 型的 ODBC 数据源名称。

info\_type                            info\_type 是 varchar(128) 型数据，是要查看的信息类型，其中

DBMS\_NAME= 数据库厂商名 ,  
DBMS\_VERSION= 版本 ,  
DATABASE\_NAME= 数据库名称 ,  
SQL\_SUBSCRIBER=Subscriber。如果没有给出此变量或其值为 NULL, 将显示所有类型。

login

login 是 varchar(128)型 ODBC 数据源名称的注册。DSN 可包括注册和口令, 此时应省略 login 和 passwd 变量。

password

password 是用于 ODBC 数据源名称的注册的 varchar(128)型口令。

## sp\_enumcodepages

sp\_enumcodepages 系统存储过程用于生成 Web 任务, 并想了解执行 sp\_makewebtask 时可用的字符集和代码页(操作系统必须预先安装了 .nls 文件)。

语法

sp\_enumcodepages

## sp\_enumcustomresolvers

sp\_enumcustomresolvers 系统存储过程可显示位于 Distributor 上由合并复制用来解决冲突的定制解析程序。

语法

```
sp_enumcustomresolvers [ distributor ]
```

变量 distributor 为分配器的系统名称。

## sp\_enumdsn

sp\_enumdsn 系统存储过程可显示运行该 SQL Server 的用户帐户可用的所有 ODBC 数据源名称。

语法

```
sp_enumdsn
```

## sp\_enumfullsubscribers

sp\_enumfullsubscribers 系统存储过程可显示订阅了某个出版物的所有文章的 Subscriber。

语法

```
sp_enumfullsubscribers [ publication ]
```

```
sp_executesql
```

sp\_executesql 系统存储过程用来执行 Transact-SQL 语句，并可以包含变量。  
语法

```
sp_executesql [ @stmt = ] stmt
```

```
[  
{, [ @params = ] N @parameter_name data_type [ ,...n ] }  
{, [ @param1 = ] value1 [ ,...n ] }  
]
```

变量

```
[ @stmt = ] stmt
```

stmt 是 Unicode 型字符串或变量 (不允许串联)。如果给出常量，必须加前缀 N。使用的每一参数都必须在 @params 参数和 @param1(sp\_executesql 的另一个变量)参数中有一个对应项。

```
[ @params = ] N @parameter_name  
data_type [ ,...n ]
```

@params 是 stmt 中所有参数的一个字符串。还必须给出 data\_type 或参数。该变量只用于 @stmt 中的嵌套参数。n 表示可重复，并用逗号隔开。



```
[ @param1 = ] value1
```

value1 是在 @params 字符串中定义的第一个参数的常量或变量。每个 parameter\_name 都必须有对应的值。

## sp\_fkeys

sp\_fkeys 系统存储过程是目录系统存储过程，它可返回外来键的关联信息。

### 语法

```
sp_fkeys { pktablename } [ , pktableowner ] [ , pktablequalifier ]  
{ , fktablename } [ , fktableowner ] [ , fktablequalifier ]
```

### 变量

pktablename

pktablename 是主键表的系统名称。必须提供此变量和 / 或 Fktablename 变量。

Pktableowner

pktableowner 是一个系统名称，表示要查看外来键信息的主键表的对象拥有者，这是一个可选变量，如果没有指定拥有者，系统将使用该顺序中可用作的当前用户或 DBO 的缺省可视

Pktablequalifier

拥有者。

pktablequalifier 是一个系统名称，表示用于查看外来键信息的主键表的数据库名称。

Fktablename

fktablename 是外来键表的系统名称，必须提供此变量和 / 或 pktablename 变量。

Fktableowner

fktableowner 表示外来键的对象拥有者的系统名称，这是一个可选变量，如果没有指定拥有者，系统将使用该顺序中可用作的当前用户或 DBO 的缺省可视拥有者。

Fktablequalifier

fktablequalifier 是外来键表的数据库的系统名。

## sp\_foreignkeys

sp\_foreignkeys 系统存储过程可显示服务器中表格的外来键，此外来键引用了链接服务器内远程表的主键。

语法

```
sp_foreignkeys { table_server } [ , pktab_name ] [ , pktab_schema ]
```

```
[ , pktab_catalog ] [ , fktab_name ] [ , fktab_schema ] [ , fktab_catalog ]
```

变量

table\_server

table\_server 为远程服务器的系统名称。

pktab\_name

pktab\_name 是带有主键的表的系统名称 (缺省值为 NULL)。

pktab\_schema

pktab\_schema 是带有主键的模式的系统名称 (缺省值为 NULL)。

pktab\_catalog

pktab\_catalog 是带有主键的目录的系统名称 (缺省值为 NULL)。

fktab\_name

fktab\_name 是带有外来键的表的系统名称 (缺省值为 NULL)。

fktab\_schema

fktab\_schema 是带有外来键的模式的系统名称 (缺省值为 NULL)。

fktab\_catalog

fktab\_catalog 是带有外来键的目录的系统名称 (缺省值为 NULL)。

## sp\_fulltext\_catalog

sp\_fulltext\_catalog 系统存储过程可创建全文目录，撤消全文目录，启动对目录的索引操作，停止对目录的索引操作。

语法

```
sp_fulltext_catalog [ @ftcat = ]          fulltext_catalog_name ,
[ @action = ]          action
[ , [ @path = ]          root_directory ]
变量
```

```
[          @ftcat          =          ]
fulltext_catalog_name
[ @action = ]          action
```

fulltext\_catalog\_name 是全文目录的系统名称。

action 是 varchar(20)型数据，其中 Create = 把全文目录作为操作系统文件，并在 sysfulltextcatalogs 系统表中插入一行。Drop = 撤消 fulltext\_catalog\_name。Start\_incremental = 启动 fulltext\_catalog\_name 的差分索引内容，且只修改由现有的受影响的时间戳列表表示的行。Start\_full = 启用 fulltext\_catalog\_name 全部内容。Stop = 停止 fulltext\_catalog\_name 的内容。Rebuild = 从头开始重建 fulltext\_catalog\_name，而没有重新加载其内容。

[ @path = ] root\_directory

root\_directory 是 varchar(100)型的根目录，其缺省值为 NULL，即 C:\mssg17\Ftdata。该变量对除创建外的每一操作都必须是 NULL。

## sp\_fulltext\_column

sp\_fulltext\_column 系统存储过程可确定表格列是否进行了全文索引。没有激活的表格要预先标记以便进行全文索引。

### 语法

```
sp_fulltext_column [ @tablename = ] qualified_table_name ,  
[ @colname = ] column_name ,  
[ @action = ] action
```

### 变量

[ @tablename = ] qualified_table_name	qualified_table_name 是格式为 database.Owner.table 或 Owner.table 的 nvarchar(517)型表格。
[ @colname = ] column_name	column_name 是表格列的系统名称。
[ @action = ] action	action 是要执行的 varchar(20)型的操作，该变量的可能值是：Add = 将一列添加到表格的未激活的索引中。

Drop = 从表格的未激活的全文索引中撤消一行。
---------------------------

## sp\_fulltext\_database

sp\_fulltext\_database 系统存储过程可根据在使用的数据库中提供的操作变量，启动数据库中的全文索引，从当前数据库中撤消所有的全文目录，使数据库中全文索引无效(不从 sysfulltextcatalogs 系统表中删除行)。要提供全文索引，必须执行 sp\_fulltext\_catalog，且提供每个目录上的 start\_full 或 start\_incremental 变量。

### 语法

```
sp_fulltext_database [ @action = ] action
```

变量 action 是 varchar(20)型变量，其中 Enable = 根据系统表中的定义，通过撤消现有的全文目录并重建所有的全文索引来启动全文索引，Disable = 撤消所有全文目录操作系统文件，并将数据库标记为禁用全文索引。

## sp\_fulltext\_service

sp\_fulltext\_service 系统存储过程用于修改 Microsoft Search Service 的全文特性。

### 语法

```
sp_fulltext_service [ @action = ]      action
[ , [ @value = ]      value      ]
    变量
```

```
[ @action = ]'action'
```

```
[ @value = ]      value
```

action 是属性的 varchar(20)型名称，其中 Resource\_usage = Microsoft Search Service 使用的资源量。Clean\_up = 删除在 sysfulltextcatalogs 系统表中没有条目的孤立的全文操作系统文件。Connect\_timeout=在加载失败之前为 Microsoft Search Service 初始化全文内容提供的秒数。value 是表示属性值的整数，其中 Connect\_timeout = 1 ~ 32767 之间的值。Resource\_usage = Microsoft Search Service 1(背景)至 5(专用)使用的资源，其缺省值为 3，Clean\_up=对清理设置为 NULL。

## sp\_fulltext\_table

sp\_fulltext\_table 系统存储过程为全文索引标记一个表。还必须使用带

start\_incremental 或 start\_full 变量的 sp\_fulltext\_catalogs 系统存储过程，以便具有全文搜索功能。

### 语法

```
sp_fulltext_table [ @tablename = ] qualified_table_name ,  
[ @action = ] action  
[ , [ @ftcat = ] fulltext_catalog_name ,  
[ @keyname = ] unique_index_name ]
```

### 变量

[ @tablename = ]	qualified_table_name	是 nvarchar(517) 型表格，其格式可以是数据库、拥有者的表格。
qualified_table_name		
[ @action = ] action	action	是 varchar(20) 型数据，其中： Create = 在系统表中建立全文索引。 Drop = 删除系统表中的行并关闭全文索引。 Activate = 启动但不提供全文索引。 Deactivate = 关闭全文索引，而在应有的位置保持元数据，但不用于全文搜索。
[ @ftcat = ]	fulltext_catalog_name	是以前建立的全文目录的系统名称。对除创建外的每个操作，都必须将该变量设置为 NULL。
fulltext_catalog_name		



```
[ @keyname = ]  
unique_index_name
```

unique\_index\_name 是以前单独索引的列，用于全文索引列的搜索。对除创建外的每个操作，都必须将该变量设置为 NULL。

## sp\_generatefilters

当对表进行复制时，sp\_generatefilters 系统存储过程可在外来键表上建立过滤器。

语法

```
sp_generatefilters [ @publication = ] publication  
变量 publication 是要建立过滤器的出版物的系统名称。
```

## sp\_getbindtoken

当客户计划使用同样的事务并共享同样的锁定空间时，sp\_getbindtoken 系统存储过程将返回一个要用作标记的字符串；该程序可与 sp\_bindsession 系统存储过程一起使用，以便将新对话与相同的事务绑定在一起。

语法

```
sp_getbindtoken{ return_value }OUTPUT  
[ ,@for_xp_flag ]
```

## 变量

return\_value

return\_value 是 nvarchar(255)型数据，它和变元一样作为一个用于传递的标志返回 sp\_bindsession 系统存储过程，此程序可启动连接以共享事务环境。OUTPUT 表示 return\_value 将放在输出变量中。

@ for\_xp\_flag

for\_xp\_flag 是一个标记，当其值为 1 时，表示创建了一个绑定标记，可以将该绑定标记传送给扩展存储过程，以便将它传送回 SQL Server。

## sp\_getmergedeletetype

sp\_getmergedeletetype 系统存储过程可显示用户删除、部分删除或系统删除的合并删除类型。

### 语法

```
sp_getmergedeletetype [ @source_object = ] source_object ,  
[ @rowguid = ] rowguid
```

### 变量

[ @source_object = ] source_object	source_object 是复制的源对象的 nvarchar(386)型名称。
[ @rowguid= ] rowguid	rowguid 是查看有删除类型的行的唯一标识符。

## sp\_get\_distributor

sp\_get\_distributor 系统存储过程可显示分配器中用户删除、部分删除或系统删除的合并删除类型。

### 语法

```
sp_get_distributor
```

## sp\_grant\_publication\_access

sp\_grant\_publication\_access 系统存储过程可为注册提供 Publication 访问权。

### 语法

```
sp_grant_publication_access [ @publication = ]      publication ,  
[ @login = ]      login  
    变量
```

[ @publication = ] publication

publication 是出版物的系统名称。如果此变量为 NULL，注册将放在服务器的全局访问列表中。SQL Server 确认的注册必须已在出版商和分配器的 SQL Server 上创建，Windows NT 确

认的保密帐户必须已在出版商 SQL Server 上创建。

login 为注册 ID 的系统名称。

[ @login = ] login

## sp\_grantdbaccess

sp\_grantdbaccess 系统存储过程可在当前数据库中为 SQL Server 确认的注册或 Windows NT 用户或组建立保密帐户 (sp\_revokedbaccess 与此相反)。

### 语法

```
sp_grantdbaccess [ @loginame = ] login [ , [ @name_in_db = ]  
name_in_db ]
```

### 变量

[ @loginame = ] login

login 是在当前数据库中建立保密帐户的注册的系统名称。Windows NT 组 and 用户必须格式化为 Domain \ user。

[ @name\_in\_db = ] name\_in\_db

name\_in\_db 是数据库中帐户的系统名称。如果该变量置于 NULL，就使用该注册。

## sp\_grantlogin

sp\_grantlogin 系统存储过程准许 Windows NT 用户或组的帐户作为 Windows NT 验证的用户与 Microsoft SQL Server 连接；该程序也可用来通过执行 sp\_denylogin 系统存储过程，以授予已被拒绝的访问权(为 SQL Server 验证的注册使用 sp\_addlogin)。

### 语法

```
sp_grantlogin [ @loginame = ]      login
```

变量 login 是格式为 Domain \ user 的 Windows NT 用户或组的系统名称。

## sp\_help

sp\_help 系统存储过程可提供除触发程序(使用 sp\_helptrigger)以外的数据库对象的信息，包括当前数据库中的数据类型。

语法表示如下：

```
sp_help [ name ]
```

变量 name 是 sysobject 系统表中任一对象的系统名称。如果没有给出此变量，将列出除触发程序外的所有数据库对象。

## sp\_help\_agent\_default

sp\_help\_agent\_default 系统存储过程可返回 Replication 代理类型的缺省配置的整数 ID。

### 语法

```
sp_help_agent_default [ @agent_type = ] agent_type
```

变量 agent\_type 是一个整数，其中 1=Snapshot Agent，2=Log Reader Agent，3=Distribution Agent，4=Merge Agent。

## sp\_help\_agent\_parameter

sp\_help\_agent\_parameter 系统存储过程可显示 Replication 代理的配置文件的参数。

### 语法

```
sp_help_agent_parameter [ @profile_id = ] profile_id
```

变量 profile\_id 是 MSagent\_configuration 系统表中配置文件的整数 ID，缺省值为 -1(所有参数)。

## sp\_help\_agent\_profile

sp\_help\_agent\_profile 系统存储过程可返回代理的配置文件。

### 语法

```
sp_help_agent_profile [ [ @agent_type = ] agent_type ]  
[ , [ @profile_id = ] profile_id ]
```

### 变量

[ @agent\_type = ] agent\_type agent\_type 是一个整数，其中 0=全部，1=Snapshot Agent，2=Log Reader Agent，3=Distribution Agent，4=Merge Agent。

[ @profile\_id = ] profile\_id profile\_id 是配置文件的整数，缺省值为 -1(所有的配置文件)。

## sp\_help\_alert

sp\_help\_alert 系统存储过程必须在 msdb 系统数据库中执行，以显示当前 SQL Server 中以前创建的警告信息。

### 语法

```
sp_help_alert [ alert_name ] [ , order_by ] [ , alert_id ] [ ,
```

category ]  
变量

alert_name	alert_name 是警告的系统名称。如果不提供此变量，将返回服务器上所有的警告信息。
order_by	order_by 是结果集的排列次序的系统名称，缺省值为名称。
alert_id	alert_id 是警告的整数 ID，其缺省值为 NULL。
category	category 是警告类别的系统名称，缺省值为 NULL。

sp\_helparticle

sp\_helparticle 系统存储过程可返回复制文章的信息。

语法

```
sp_helparticle [ @publication = ] publication [ , [ @article = ]  
article ]  
[ , [ @returnfilter = ] returnfilter ]
```

变量

[ @publication = ] publication publication 是 Publication 的系统名称。

[ @article = ] article article 是文章的系统名称。如果不提供此变量，将返回 Publication 的



[ @returnfiler = ] returnfilter

所有文章的信息。

returnfilter 是一个位型数据，其中 1=返回过滤器子句。0=不返回过滤器子句。

## sp\_helparticlecolumns

sp\_helparticlecolumns 系统存储过程可列出文章表内的所有列，用于检查与包含复制的纵向分区。

### 语法

```
sp_helparticlecolumns [ @publication = ]      publication      ,  
[ @article = ]      article
```

### 变量

[ @publication = ] publication publication 是 Publication 的系统名称。

[ @article = ] article article 是文章的系统名称。

## sp\_help\_category

sp\_help\_category 系统存储过程可返回关于作业、警告或操作员的类别信息。执行不带参数的此系统存储过程可以查看可用变量的值。

### 语法

```
sp_help_category [ class ] [ , type ] [ , name ] [ , suffix ]
```

### 变量

class	class 是 varchar(8)型的类别，其可能的值有 JOB(缺省值)，ALERT 或 OPERATOR。
type	type 是 varchar(12)型数据，表示要报告的信息类型。其值可以是 LOCAL, MULTI-SERVER, NONE 和 NULL(缺省值)。
name	name 是系统名称，NULL=缺省值。 suffixsuffix 是位标志，其中 1=命令 SQL Server 添加后缀以定义对本地和多重服务器类别名称的搜索。

## sp\_helpconstraint

sp\_helpconstraint 系统存储过程可提供表格上的约束信息。

### 语法

```
sp_helpconstraint { table } [ , no_message ]
```

### 变量

table table 是表格的 nvarchar(776)型名称。

no\_message no\_message 是 varchar(5)型数据，其中 nomsg = 取消表名，msg=不取消表名（缺省值）。

## sp\_helpdb

sp\_helpdb 系统存储过程可返回数据库的信息，如果没有提供变量，则显示所有数据库的信息。

### 语法

```
sp_helpdb [ name ]
```

变量 name 是数据库的系统名称。

## sp\_helpdbfixedrole

sp\_helpdbfixedrole 系统存储过程可显示固定数据库角色的信息。

语法

```
sp_helpdbfixedrole [ [ @rolename = ] role ]
```

变量 `role` 是固定数据库角色的系统名称。执行不带参数的此系统存储过程可以查看所有固定数据库角色的列表。

## sp\_helpdevice

sp\_helpdevice 系统存储过程只向后兼容，Microsoft SQL Server 7 的程序员可使用文件和文件组。

语法

```
sp_helpdevice [ name ]
```

变量 `name` 是设备的系统名称。

## sp\_helpdistpublisher

sp\_helpdistpublisher 系统存储过程显示出版商的信息，该出版商也是他自己的分配器。

## 语法

```
sp_helptdispublisher [ , [ @publisher = ] publisher ]
```

变量 publisher 是出版商的系统名称。

## sp\_helpdistributiondb

sp\_helpdistributiondb 系统存储过程可显示分布式数据库的信息。

## 语法

```
sp_helpdistributiondb [[ @ =database = ] database ]
```

变量 database 是数据库的系统名称。

## sp\_helpdistributor

sp\_helpdistributor 系统存储过程可显示分配器的信息。

## 语法

```
sp_helpdistributor [[ @publisher = ] publisher ] [ , [ @local = ]  
local ]
```

## 变量

[ @publisher = ] publisher

publisher 是出版商的系统名称。

[ @local = ] local

local 是 nvarchar(5)型的指示符，其值

为 true 或 false , 缺省值 NULL 表示为本地分配器。

## sp\_help\_downloadlist

sp\_help\_downloadlist 系统存储过程可根据作业或为作业步骤显示 sysdownloadlist 系统表中的内容。可不提供变量, 或提供 job\_id 或 job\_name, 或不提供这两个变量。

### 语法

```
sp_help_downloadlist [ { job_id | job_name } ] [ , operation ] [ ,  
object_type ]  
[ , object_name ] [ , target_server ] [ , has_error ] [ , status ]  
[ , date_posted ]
```

### 变量

job_id	job_id 是作业的唯一标识符。
job_name	job_name 是作业的 varchar(100) 型名称。
operation	operation 是作业的 varchar(64) 型操作, 其中: DEFECT=释放目标服务器, DELETE=删除作业, INSERT=插入新作业或刷新作业, RE-ENLIST=为多重服务器域再次发送注册信息, 再次从 MSXOperator 处将信息下载到目标服务器上, SET-POLL=为目标服务器设置秒

数，范围从 1 ~ 28,800，以轮询多重服务器域，START=开始执行作业，STOP=停止执行作业，SYNC-TIME=目标服务器将其时钟与多重服务器域的时钟同步，UPDATE=更新作业的系统作业信息。

object\_type object\_type 是 varchar(64)型数据，其值为 JOB 或 SERVER。

object\_name object\_name 是依 Object\_type 而定 JOB 或 SERVER 的 varchar(100)型名称。

target\_server target\_server 是目标服务器的 varchar(30)型名称。

has\_error has\_error 是 tinyint 型标志，其中 1=确认全部错误，NULL=确认没有错误。

Status status 是 tinyint 型的作业状态。

date\_posted data\_posted 是当时或之后结果集的切断点的日期。

## sp\_helpextendedproc

sp\_helpextendedproc 系统存储过程可返回扩展存储过程的信息。

语法

sp\_helpextendedproc [ procedure ]

变量 procedure 是扩展存储过程的系统名称。

## sp\_helpfile

sp\_helpfile 系统存储过程可显示当前数据库中数据库文件的信息。

### 语法

```
sp_helpfile [ @filename = ]          name
```

变量 name 是与当前数据库有关的数据库文件的逻辑系统名称。如果没有提供该变量，则显示所有数据库的文件。

## sp\_helpfilegroup

sp\_helpfilegroup 系统存储过程可显示属于当前数据库的文件组的信息。

### 语法

```
sp_helpfilegroup [ @filegroupname = ]      name
```

变量 name 是属于当前数据库的文件组的逻辑系统名称。如果没有提供此变量，将显示所有数据库中的文件组。

## sp\_help\_fulltext\_catalogs

sp\_help\_fulltext\_catalogs 系统存储过程可显示全文目录的信息，如果没有提供变量，则显示所有的全文目录。



## 语法

```
sp_help_fulltext_catalogs [ @fulltext_catalog_name = ]  
fulltext_catalog_name
```

变量 `fulltext_catalog_name` 是全文目录的系统名称。

## sp\_help\_fulltext\_catalogs\_cursor

`sp_help_fulltext_catalogs_cursor` 系统存储过程可利用游标显示全文目录的详细信息，如果没有提供变量，则显示当前数据库中所有的全文目录。

## 语法

```
sp_help_fulltext_catalogs [ @cursor_return = ] @cursor_variable  
OUTPUT,  
[ @fulltext_catalog_name = ] fulltext_catalog_name
```

## 变量

[ @cursor\_return = ] @cursor\_variable 是游标的输出变量。

[ @fulltext\_catalog\_name = ] fulltext\_catalog\_name 是全文目录的系统名称。

## sp\_help\_fulltext\_columns

sp\_help\_fulltext\_columns 系统存储过程可显示为全文所有标记的列的信息。

### 语法

```
sp_help_fulltext_columns [ @table_name = ]          table_name ,  
[ @column_name = ]          column_name
```

### 变量

[ @table_name = ]          table_name	table_name 是 nvarchar(517) 型数据，表示任选的拥有者的合法表名。如果不提供此变量，则包含为全文索引标记的所有表。
[ @column_name = ]          column_name	column_name 是列的系统名称。如果不提供此变量，将显示表中包含的所有列。

## sp\_help\_fulltext\_tables

sp\_help\_fulltext\_tables 系统存储过程可显示为全文索引标记的表的信息。

### 语法

```
sp_help_fulltext_tables [ @fulltext_catalog_name = ]
    fulltext_catalog_name ,
[ @table_name = ]      table_name
    变量
```

<pre>[ @fulltext_catalog_name = ] fulltext_catalog_name</pre>	<p>fulltext_catalog_name 是全文目录的系统名称。如果不提供此变量，或其值为 NULL，则返回数据库中所有的全文目录。</p>
<pre>[ @table_name = ]      table_name</pre>	<p>table_name 是 nvarchar(517)型数据，表示任选的拥有者的合法表名。</p>

sp\_help\_fulltext\_tables\_cursor

sp\_help\_fulltext\_tables\_cursor 系统存储过程可利用游标显示为全文索引注册的表格。

语法

```
sp_help_fulltext_tables_cursor [ @cursor_return = ]      @cursor_variable
    OUTPUT,
[ @fulltext_catalog_name = ]      fulltext_catalog_name ,
[ @table_name = ]      table_name
    变量
```

[ @cursor_return = ]	cursor_variable 是游标的输出变量。
@cursor_variable OUTPUT	
[ @fulltext_catalog_name = ]	fulltext_catalog_name 是全文目录的系统名称。
fulltext_catalog_name	
[ @table_name = ]	table_name 是 nvarchar(517) 型数据，表示任选的拥有者的合法表名。
table_name	

## sp\_help\_fulltext\_columns\_cursor

sp\_help\_fulltext\_columns\_cursor 系统存储过程可利用游标返回为全文标索引标记的列的信息。

### 语法

```
sp_help_fulltext_columns_cursor [ @cursor_return = ]
    @cursor_variable OUTPUT,
[ @table_name = ] table_name ,
[ @column_name = ] column_name
    变量
```

[ @cursor_return = ]	cursor_variable 是游标的输出变量。
@cursor_variable OUTPUT	
[ @table_name = ]	table_name 是 nvarchar(517) 型数据，表示任选的拥有者的合法表名。如果
table_name	

```
[ @column_name = ]  
column_name
```

不提供此变量，则包含为全文索引标记的所有表。

column\_name 是列的系统名称。如果不提供此变量，将显示表中包含的所有列。

## sp\_helpgroup

sp\_helpgroup 系统存储过程只向后兼容，可显示当前数据库中的角色信息。Microsoft SQL Server 7 程序员可使用数据库角色代替数据库组。

### 语法

```
sp_helpgroup [[ @grpname = ] role ]
```

变量 role 是数据库角色的系统名称。如果不提供此变量，将返回当前数据库中所有角色的信息。

## sp\_helphistory

sp\_helphistory 系统存储过程只向后兼容。Microsoft SQL Server 程序员可使用作业代替它(sp\_help\_jobhistory)。

## sp\_help\_job

sp\_help\_job 系统存储过程可显示作业的信息。需提供 job\_id 或 job\_name 变量，但是不必同时提供两者。

### 语法

```
sp_help_job { [ @job_id = ] job_id | [ @job_name = ] job_name }
[ , [ @job_aspect = ] job_aspect ] [ , [ @job_type = ] job_type
]
[ , [ @owner_login_name = ] login_name ] [ , [ @subsystem = ] subsystem
]
[ , [ @category_name = ] category ] [ , [ @enabled = ] enabled ]
[ , [ @execution_status = ] status ] [ , [ @date_comparator = ]
date_comparison ]
[ , [ @date_created = ] date_created ] [ , [ @date_last_modified = ]
date_modified ]
[ , [ @description = ] description_pattern ]
```

### 变量

job_id	job_id 指作业的唯一标识符。
job_name	job_name 是作业的 varchar(100) 型名称。
job_aspect	job_aspect 是要显示的 varchar(9)

	型作业特征，该变量可能的值为：ALL，JOB，SCHEDULES，STEP或者TARGETS。
job_type	job_type 是 varchar(12)型数据，其值为 LOCAL，MULTI_SERVER 或 NULL。
login_name	login_name 是作业拥有者的 varchar(64)型的注册名称。缺省值为 NULL。
subsystem	subsystem 是子系统的 varchar(60)型名称，缺省值为 NULL。
category	category 是类别的 varchar(100)型名称，缺省值为 NULL。
Enabled	enabled 是 tinyint 型数据，其中 1=启动作业，0=禁止作业，NULL=缺省值。
Status	status 是表示作业状态的整数，其中 0=非空闲或挂起的作业，1=执行作业，2=作业等待线索，3=重试期间的作业，4=空闲作业，5=挂起的作业，7=完成作业。
date_comparison	date_comparison 是 char(1)型数据，该变量可能的值是 =，<或>，用于比

date\_modified

较 data\_created 与 date\_modified。  
date\_createddate\_created 是作业建立的日期，缺省值为 NULL。

date\_modified 是作业修改的日期，缺省值为 NULL。

description\_pattern

description\_pattern 是 varchar(512) 型数据，是对作业的描述。可使用通配符。

## sp\_help\_jobhistory

sp\_help\_jobhistory 系统存储过程可显示多重服务器环境下服务器的作业信息。

### 语法

```
sp_help_jobhistory [ job_id ] [ , job_name ] [ , step_id ]  
[ , sql_message_id ]  
[ , sql_severity ] [ , start_run_date ] [ , end_run_date ] [ , start_run_time ]  
[ , end_run_time ] [ , minimum_run_duration ] [ , run_status ]  
[ , minimum_retries ] [ , oldest_first ] [ , server ] [ , mode ]
```



## 变量

<code>job_id</code>	<code>job_id</code> 是作业的唯一标识符。
<code>job_name</code>	<code>job_name</code> 是作业的系统名称。
<code>step_id</code>	<code>step_id</code> 是一个整数，缺省值为 NULL。
<code>sql_message_id</code>	<code>SQL_message_id</code> 是整型错误信息 ID，当执行作业时，返回该变量，缺省值为 NULL。
<code>sql_severity</code>	<code>SQL_severity</code> 是错误信息的整型严重性等级，缺省值为 NULL。
<code>start_run_date</code>	<code>start_run_date</code> 是一个整数，其格式为年、月、日 YYYYMMDD，表示作业开始的日期。
<code>end_run_date</code>	<code>end_run_date</code> 是一个整数，其格式为年、月、日 YYYYMMDD，表示作业结束的日期。
<code>start_run_time</code>	<code>start_run_time</code> 是一整数，其格式为时、分、秒 HHMMSS，表示作业开始的时间。
<code>end_run_time</code>	<code>end_run_time</code> 是一整数，其格式为时、分、秒 HHMMSS，表示作业结束的时间。
<code>minimum_run_duration</code>	<code>minimum_run_duration</code> 是一整数，其格式为时、分、秒 HHMMSS，表示完成作业所需的最短时间。
<code>run_status</code>	<code>run_status</code> 是一整数，表示作业执行的状态，其中 0=失败，1=成功，2=重试，3=取消，4=在进程中信息，5=未知。

minimum_retries	minimum_retries 是一整数，表示重复执行作业的最少次数。
oldest_first	oldest_first 是一个整数，其值为 1 时，表示先输出最旧的作业，0(缺省值)表示先输出最新的作业。
server	server 表示执行作业的服务器的 varchar(30)型名称。
mode	mode 是 varchar(7)型数据，其中 NULL=打印所有列，SUMMARY=打印摘要。

## sp\_help\_jobschedule

sp\_help\_jobschedule 系统存储过程可显示作业调度的信息。变量可互相配合使用。语法 @argument\_name=supplied\_value 用于此存储过程，需提供 job\_id 或 job\_name，但不需要同时提供两者。

### 语法

```
sp_help_jobschedule
{
  { [ @job_id = ]      job_id      | [ @job_name = ]      job_name      }
  [ , [ @schedule_name = ]      schedule_name      ]
  |
  [ , [ @schedule_name = ]      schedule_name      , ]
  { [ @schedule_id = ]      schedule_id      }
}
```

## 变量

job_id	job_id 是作业号的唯一标识符，此变量不能与 schedule_id 或 job_name 一起作用，但可和 schedule_name 一起使用。
job_name	job_name 是作业的系统名称，此变量不能与 schedule_id 或 job_id 一起作用，但可和 schedule_name 一起使用。
schedule_name	schedule_name 是作业调度的 varchar(100)型名称。
schedule_id	schedule_id 是表示作业调度的整数。

## sp\_help\_jobserver

sp\_help\_jobserver 系统存储过程可显示作业的信息。需提供 job\_id 或 job\_name，但不必两者都提供。

## 语法

```
sp_help_jobserver { job_id | job_name } [ ,show_last_run_details ]
```

## 变量

job_id	job_id 是作业号的唯一标识符
job_name	job_name 是作业的系统名称
show_last_run_details	show_last_run_details 是 tinyint 型数据，其中 1=显示此作业最后一次执行

的细节，0=不显示此作业最后一次执行的细节。

## sp\_help\_jobstep

sp\_help\_jobstep 系统存储过程可显示作业的步骤信息。需提供 job\_id 或 job\_name，但不必两者都提供。

### 语法

```
sp_help_jobstep { [ @job_id = ] job_id | [ @job_name = ] job_name }  
[ , [ @step_id = ] step_id ] [ , [ @step_name = ] step_name ] [ , [ @suffix  
= ] suffix ]
```

### 变量

job_id	job_id 是作业号的唯一标识符
job_name	job_name 是作业的系统名称
step_id	step_id 是整型作业步骤号。如果不提供此变量，将显示作业的所有步骤。
step_name	step_name 是作业步骤的系统名称。
Suffix	suffix 是一个位标志，其中 1=包括文字说明，0=不包括文字说明(缺省值)。

## sp\_helpindex

sp\_helpindex 系统存储过程可显示表格索引的信息。

语法

```
sp_helpindex { name }
```

变量 name 是 nvarchar(776)型数据，是当前数据库中的表名。

## sp\_helplanguage

sp\_helplanguage 系统存储过程可显示位于 syslanguages 系统表中的 Microsoft SQL Server 语言的信息。

语法

```
sp_helplanguage [ language ]
```

变量 language 是语言的系统名称。如果不提供此变量，将显示所有的语言。

## sp\_helplinkedserverlogin

sp\_helplinkedserverlogin 系统存储过程可显示为链接服务器映射的注册。

语法

```
sp_helplinkedserverlogin [ [ @ rmtsrvname = ] rmtsrvname , ]
```

[[ @locallogin = ] locallogin ]  
变量

[ @ rmtsrvname = ] rmtsrvname	rmtsrvname 是链接服务器的系统名称。
[ @ locallogin = ] locallogin	locallogin 是一个系统名称,是映射到链接服务器 rmtsrvname 的本地 SQL Server 注册。

## sp\_helplogins

sp\_helplogins 系统存储过程可返回 SQL Server 注册和注册的数据库帐户的信息。该存储程序显示 LoginName,SID(注册安全标识符), DefaultDBName,DefaultLanguageName,IsDatabaseUserYesOrNo 和 DefaultDatabaseName。报表还显示数据库,角色,注册有帐户的每个数据库中的数据库用户帐户。

### 语法

sp\_helplogins [[ @LoginNamePattern = ] login ]

变量 login 是注册的系统名称。

## sp\_helpmergearticle

sp\_helpmergearticle 系统存储过程可显示用于合并复制的文章信息。

### 语法

```
sp_helpmergearticle [[ @publication = ]      publication      ] [[ @article  
= ]      article      ]
```

### 变量

[ @publication = ] publication publication 为出版物的系统名称，  
如果不提供此变量，缺省值 % 表示全  
部的出版物。

[ @article = ] article article 是文章的系统名称。如果不  
提供此变量，缺省值 % 表示全部的文  
章。

## sp\_helpmergearticleconflicts

sp\_helpmergearticleconflicts 系统存储过程可列出出版物中有冲突的文章。

### 语法

```
sp_helpmergearticleconflicts [[ @publication = ]      publication      ]
```

变量 publication 是出版物的系统名称。如果不提供此变量，缺省值 % 表示全  
部的出版物。

## sp\_helpmergeconflictrows

sp\_helpmergeconflictrows 系统存储过程可返回冲突表中的合并复制出版及其冲突的信息。

### 语法

```
sp_helpmergeconflictrows [ [ @publication = ]      publication      ]  
{ , [ @conflict_table = ]      conflict_table      }
```

### 变量

[ @publication = ] publication publication 是出版物的系统名称。如果不提供此变量，缺省值 % 表示全部的出版物。

[ @conflict\_table = ] conflict\_table conflict\_table 是冲突表的系统名称。

## sp\_helpmergedeleteconflictrows

sp\_helpmergedeleteconflictrows 系统存储过程显示 msmerge\_delete\_conflicts 复制表中

### 语法

```
sp_helpmergedeleteconflictrows [ [ @publication = ]      publication      ]
```



[ , [ @source\_object = ] source\_object ]  
变量

[ @publication = ] publication	publication 是出版物的系统名称，如果不提供此变量，缺省值%表示全部的出版物。
[ @source_object = ] source_object	source_object 是源对象的系统名称。

## sp\_helpmergefilter

sp\_helpmergefilter 系统存储过程可显示用于合并复制的过滤器的信息。

### 语法

sp\_helpmergefilter [ @publication = ] publication , [ @article = ] article  
[ , [ @filtername = ] filtername ]  
变量

[ @publication = ] publication

publication 是出版物的系统名称，如果不提供此变量，缺省值%表示所有的出版物。

[ @article = ] article

article 是文章的系统名称，如果不提供此变量，缺省值%表示所有的文章。

[ @filename = ] filename ] filename 是过滤器的系统名称。如果不提供此变量，系统缺省值 % 表示所有的过滤器。

## sp\_helpmergepublication

sp\_helpmergepublication 系统存储过程可显示用于合并复制的出版信息。

### 语法

```
sp_helpmergepublication [[ @publication = ] publication ]  
[ , [ @found = ] found ] [ , [ @publication_id = ] publication_id  
]  
[ , [ @reserved = ] reserved ]
```

### 变量

[ @publication = ] publication publication 是出版物的系统名称，如果不提供此变量，缺省值 % 表示所有的出版物。

[ @found = ] found found 是一个整数，其中 1=找到出版物，2=没找到出版物。

[ @publication\_id = ] publication\_id publication\_id 是出版物的唯一标识符。

[ @reserved = ] reserved 为 Microsoft SQL Server 保留。

## sp\_helpmergepullsubscription

sp\_helpmergepullsubscription 系统存储过程可显示用于合并复制的推出订阅信息。

### 语法

```
sp_helpmergepullsubscription [ [ @publication = ] publication ]  
[ , [ @publisher = ] publisher ] [ , [ @publisher_db = ]  
publisher_db ]  
[ , [ @subscription_type = ] subscription_type ]
```

### 变量

[ @publication = ] publication

publication 是出版物的系统名称，如果不提供此变量，缺省值 % 表示所有的出版物。

[ @publisher = ] publisher

publisher 是出版商的系统名称，如果没有提供此变量，缺省值 % 表示所有的出版商。

[ @publisher\_db = ]  
publisher\_db

publisher\_db 是出版商数据库的系统名称，如果没有提供此变量，缺省值 % 表示所有的出版商数据库。

[ @subscription\_type = ]

subscription\_type 是 nvarchar(10)型

subscription\_type

数据，其值为推入、推出(缺省值)或两者兼有。

## sp\_helpmergesubscription

sp\_helpmergesubscription 系统存储过程可显示用于合并复制的推入订阅信息。

### 语法

```
sp_helpmergesubscription [ [ @publication = ] publication ]  
[ , [ @subscriber = ] subscriber ] [ , [ @subscriber_db = ]  
subscriber_db ]  
[ , [ @publisher = ] publisher ] [ , [ @publisher_db = ]  
publisher_db ]  
[ , [ @subscription_type = ] subscription_type ]
```

### 变量

[ @publication = ] publication publication 是出版物的系统名称，如果不提供此变量，缺省值%表示所有的出版物。

[ @subscriber = ] subscriber subscriber 是为订户的系统名称，如果没有提供此变量，缺省值%表示所

```
[ @subscriber_db = ]  
subscriber_db
```

```
[ @publisher = ] publisher
```

```
[ @publisher_db = ]  
publisher_db
```

```
[ @subscription_type = ]  
subscription_type
```

有的订户。

subscriber\_db 是订户数据库的系统名称，如果没有提供此变量，缺省值 % 表示所有的订户数据库。

publisher 是出版商的系统名称，如果没有提供此变量，缺省值 % 表示所有的出版商。

publisher\_db 是出版商数据库的系统名称，如果没有提供此变量，缺省值 % 表示所有的出版商数据库。

subscription\_type 是 nvarchar(10) 型数据，其值为推入、推出(缺省值)或两者兼有。

## sp\_help\_notification

sp\_help\_notification 系统存储过程可显示自动警告的通知信息。

语法

```
sp_help_notification { object_type } [ , name ] [ , enum_type ]  
[ , notification_method ] [ , target_name ]
```

变量

object_type	object_type 是字符(9)型数据，其中 ALERTS=操作员的警告,OPERATORS=警告操作符。
name	name 是字符(100)型数据，如果变量 object_type 为 ALERTS，该变量表示警告名，如果变量 object_type 为 OPERATORS，该变量表示操作符的名称。
enum_type	enum_type 是字符(10)型数据，表示返回的信息。其中 ACTUAL= name object_type，ALL=所有的 object_type (即使它们不属于“名称”)，TARGET= target_name object_type。
notification_method	notification_method 是 tinyint 型数据，其中 1=返回电子邮件信息，2=返回寻呼信息，4=返回 NetSend 信息，7=返回全部信息。
target_name	target_name 是字符(100)型数据；如果 enum_type 为 TARGET,则使用该变量。如果 object_type 为 ALERTS,变量 target_name 就为报警名称，如果 object_type 为 OPERATORS,则该变量为操作员名。

## sp\_helpntgroup

sp\_helpntgroup 系统存储过程可显示当前数据库中带有数据库帐户的 Microsoft Windows NT 组的信息。

## 语法

```
sp_helpntgroup [[ @ntname = ]      name      ]
```

变量 `name` 是 Microsoft Windows NT 组的系统名称。如果没有给出此变量，将显示当前数据库中所有带数据库帐户的 NT 组。

## sp\_help\_operator

`sp_help_operator` 系统存储过程可显示当前 Microsoft SQL Server 的操作员的信息。此存储程序必须在 `msdb` 系统数据库中运行。

### 语法

```
sp_help_operator [  operator_name  ][ ,operator_id ]
```

### 变量

<code>operator_name</code>	<code>operator_name</code> 是操作员的系统名称。如果没有给出此变量，将返回所有操作员的信息。
<code>operator_id</code>	<code>operator_id</code> 是一个整数，是操作员的标识号。

## sp\_helppublication

`sp_helppublication` 系统存储过程可显示用于复制的出版信息。

### 语法

```
sp_helppublication [[ @publication = ]      publication      ]
```

变量 `publication` 是出版的系统名称。如果没有给出此变量，将返回本地 SQL Server 中所有的出版信息。

## `sp_help_publication_access`

`sp_help_publication_access` 系统存储过程可显示出版物的所有授权注册。

### 语法

```
sp_help_publication_access [ @publication = ]      publication
[ @return_granted = ]      return_granted         ][ , [ @login = ]      login
]
```

### 变量

[ @publication = ] publication

`publication` 是出版物的系统名称。如果没有提供此变量，将返回全局访问注册。

[ @return\_granted = ]  
return\_granted

`return_granted` 是一个位型数据，其中 0=未授权的出版商的所有注册，1=授权出版商的所有注册。

[ @login = ] login

`login` 是 SQL Server 已验证的注册 ID 的系统名称。如果没有给出此变量，缺省值 % 表示所有注册。



## sp\_helppublication\_snapshot

sp\_helppublication\_snapshot 系统存储过程可显示出版快照的信息，该程序可用于复制环境中。

### 语法

```
sp_helppublication_snapshot [ @publication = ]          publication
```

变量 publication 是出版物的系统名称。

## sp\_helppullsubscription

sp\_helppullsubscription 系统存储过程可显示订户下的一个或多个订阅信息。

### 语法

```
sp_helppullsubscription [[ @publisher = ]          publisher          ]  
[ , [ @publisher_db = ]          publisher_db          ] [ , [ @publication = ]  
publication          ]  
[ , [ @show_push = ]          show_push
```

### 变量

[ @publisher = ] publisher publisher 是出版商的系统名称，如果没有提供此变量，缺省值 % 表示所有出版商。

<pre>[ @publisher_db = ] publisher_db</pre>	<pre>publisher_db 是出版商数据库的系统名称。如果没有提供此变量，缺省值%表示所有出版商数据库。</pre>
<pre>[ @publication = ] publication</pre>	<pre>publication 是出版物的系统名称，如果没有给出此变量，缺省值%表示所有出版物。</pre>
<pre>[ @show_push = ] show_push</pre>	<pre>show_push 是 nvarchar(5)型数据，其中 FALSE=不显示推入订阅，TRUE=显示推入订阅。</pre>

## sp\_helpremotelogin

sp\_helpremotelogin 系统存储过程可显示在本地服务器上定义的远程注册的信息。

### 语法

```
sp_helpremotelogin [[ @remoteserver = ]
[ , [ @remotename = ]
remote_name ] ]
变量
```

<pre>[ @remoteserver = ] remoteserver</pre>	<pre>remoteserver 是远程服务器的系统名称，如果没有提供此变量，缺省值%表示所有远程服务器。</pre>
---	--

[ @remotename = ] remote\_name remote\_name 是远程注册的系统名称，如果没有提供此变量，缺省值 % 表示所有远程注册。

## sp\_helpreplicationdb

sp\_helpreplicationdb 系统存储过程可显示本地服务器上出版数据库的信息。

### 语法

```
sp_helpreplicationdb [ [ @dbname = ] dbname ] [ , [ @type = ] type ]
```

### 变量

[ @dbname = ] dbname	dbname 是数据库的系统名称，如果没有提供此变量，缺省值 % 表示所有数据库。
[ @type = ] type	type 是为信息类型的系统名称。其中 Pub=出版数据库，Sub=订阅数据库。

## sp\_helpreplicationdboption

sp\_helpreplicationdboption 系统存储过程用于本地 SQL Server 上以列出带有复制的数据库。

## 语法

```
sp_helpreplicationdboption [[ @dbname = ]      dbname      ] [ , [ @type =  
type      ]
```

## 变量

[ @dbname = ] dbname dbname 是数据库的系统名称，如果没有给出此变量，缺省值 % 表示所有数据库。

[ @type = ] type type 是一个系统名称，它确定是否允许复制，缺省值为允许复制。

## sp\_helpreplicationoption

sp\_helpreplicationoption 系统存储过程可显示复制选项的信息。

## 语法

```
sp_helpreplicationoption [[ @optname = ]      optname      ]
```

变量 optname 是复制选项的系统名称，其中 Transactional=事务复制，Merge=合并复制，NULL=缺省值(显示所有的复制)。

## sp\_helprole

sp\_helprole 系统存储过程可显示当前数据库中角色的信息。

## 语法

```
sp_helprole [[ @rolename = ] role ]
```

变量 `role` 是当前数据库中角色的系统名称，如果没有提供此变量，缺省值 `%` 表示所有角色。

## sp\_helprolemember

`sp_helprolemember` 系统存储过程可显示应用数据库中数据库角色成员的信息。

## 语法

```
sp_helprolemember [[ @rolename = ] role ]
```

变量 `role` 是当前数据库中数据库角色的系统名称。如果没有提供此变量，将返回数据库中至少有一个成员的所有角色。

## sp\_helpprotect

`sp_helpprotect` 系统存储过程可显示当前数据库中用户许可与执行的信息。

## 语法

```
sp_helpprotect [[ @name = ] object_statement ]
```

```
[ , [ @username = ] security_account ]
```

```
[ , [ @grantorname = ] grantor ] [ , [ @permissionarea = ] type
```

]

变量

[ @name = ] object\_statement

[ @username = ]  
security\_account

[ @grantorname = ] grantor

[ @permissionarea = ] type

object\_statement 是 nvarchar(776) 型数据，是数据库中的一个对象。它也可以是所有的 CREATE 语句，BACKUP DATABASE 与 BACKUP LOG 语句。

security\_account 是安全帐户的系统名称，如果没有提供此变量，将返回当前数据库中所有安全帐户的权限。

grantor 是最初授予许可的安全帐户的系统名称。如果没有提供此变量，将显示所有的授予人。

type 是 varchar(10) 型数据，其中 o= 对象许可，s= 语句许可，可以使用 o 和 s 的任意组合用逗号隔开，也可不用逗号隔开。

sp\_helpserver

sp\_helpserver 系统存储过程将显示远程服务器或复制服务器的信息。

语法

sp\_helpserver [ server ] [ , option ] [ , show\_topology

] 变量

server	server 是服务器的系统名称。如果没有提供此变量，就报告 sysservers 系统表中所有的服务器。
option	option 是描述服务器类型的 varchar(35)型数据，其中 Dist= Distributor，dpub=本地 Distributor 的 Remote Publisher，dsn=DSN 服务器，dsub=本地 Distributor 的 Remote Subscriber，fallback=Fallback 服务器，pullpub=Pull Publisher，pub= Publisher，rpc=Remote 服务器，sub=Subscriber。
show_topology	show_topology 是 varchar(1)型数据，其中 1=显示服务器的相互关系及返回的其他信息，NULL=不显示服务器的相互关系及返回的其他信息。

## sp\_helpsort

sp\_helpsort 系统存储过程可返回本地服务器的排列次序和字符集。

语法

sp\_helpsort

## sp\_helpsrvrole

sp\_helpsrvrole 系统存储过程可显示固定服务器角色。

### 语法

```
sp_helpsrvrole [[ @srvrolename = ] role ]
```

变量 role 是固定服务器角色的系统名称。其中 sysadmin=系统管理员，securityadmin=安全管理员，serveradmin=服务器管理器，setupadmin=安装管理员，processadmin=过程管理员，diskadmin=磁盘管理员，dbcreator=数据库创建者。如果没有提供此变量，将返回所有服务器角色的信息。

## sp\_helpsrvrolemember

sp\_helpsrvrolemember 系统存储过程可显示固定的服务器角色成员的信息。

### 语法

```
sp_helpsrvrolemember [[ @srvrolename = ] role ]
```

变量 role 是固定服务器角色的系统名称，其中 sysadmin=系统管理员，securityadmin=安全管理员，serveradmin=服务器管理员，setupadmin=安装管理员，processadmin=进程管理员，diskadmin=磁盘管理员，dbcreator=数据库创建者。如果没有提供此变量，将返回所有服务器角色的信息。



## sp\_helpsubscriber

sp\_helpsubscriber 系统存储过程可显示订户的订阅信息。

SP 语法

```
sp_helpsubscriber [[ @subscriber = ] subscriber ]
```

变量 subscriber 是订户的系统名称，缺省值 % 表示所有订户。

## sp\_helpsubscriberinfo

sp\_helpsubscriberinfo 系统存储过程可显示本地 Microsoft SQL Server 上复制订户的信息。

语法

```
sp_helpsubscriberinfo [[ @subscriber = ] subscriber ]
```

变量，subscriber 是订户的系统名称，如果不提供此变量，缺省值 % 表示所有订户。

## sp\_helpsubscription

sp\_helpsubscription 系统存储过程可返回订阅信息。

语法

```
sp_helpsubscription[[ @publication = ]      publication      ][, [ @article  
= ]      article      ]  
[, [ @subscriber = ]      subscriber      ][, [ @destination_db = ]  
destination_db      ]
```

### 变量

[ @publication = ] publication publication 是出版物的系统名称，如果没有提供此变量，缺省值%表示所有出版物。

[ @article = ] article article 是文章的系统名称，如果没有提供此变量，缺省值%表示出版物和订户的全部预订信息。All=返回一行中某一出版物的所有订户。

[ @subscriber = ] subscriber subscriber 是查看预订的订户的系统名称。如果没有提供此变量，缺省值%表示出版物和文章的所有预订信息。

[ @destination\_db = ] destination\_db destination\_db 是目标数据库的系统名称，如果没有提供此变量，缺省值%表示所有目标数据库。

## sp\_helpsubscription\_properties

sp\_helpsubscription\_properties 系统存储过程可显示与安全有关的 MSsubscription\_properties 复制表中的信息。

### 语法

```
sp_helpsubscription_properties [[ @publisher = ] publisher ] [ ,  
[ @publisher_db = ] publisher_db ] [ , [ @publication = ] publication ]  
[ , [ @publication_type = ] publication_type ]
```

### 变量

[ @publisher = ] publisher	publisher 是出版商的系统名称，如果没有提供此变量，缺省值 % 表示所有出版商。
[ @publisher_db = ] publisher_db	publisher_db 是出版商数据库的系统名称，如果没有提供此变量，缺省值 % 表示所有出版商数据库。
[ @publication = ] publication	publication 是出版物的系统名称，如果没有提供此变量，缺省值 % 表示所有出版物。
[ @publication_type = ]	publication_type 是出版的类型。它是

publication\_type ]

一个整数，缺省值为 0=Snapshot，  
1=Transactional。

## sp\_help\_targetserver

sp\_help\_targetserver 系统存储过程可返回与目标服务器有关的信息。

语法

```
sp_help_targetserver [ server_name ]
```

变量 server\_name 是目标服务器的 nvarchar(30)型名称。如果没有提供此变量，将返回所有目标服务器的信息。

## sp\_help\_targetservergroup

sp\_help\_targetservergroup 系统存储过程可显示目标服务器组中的目标服务器。

语法

```
sp_help_targetservergroup [ name ]
```

变量 name 是目标服务器组的系统名称，如果没有提供此变量，将返回所有目标服务器组的信息。

## sp\_helptask

sp\_helptask 系统存储过程只向后兼容。Microsoft SQL Server 7 的程序员用作业代替任务。

## sp\_helptext

sp\_helptext 系统存储过程可返回当前数据库中存储在 syscomments 系统表中的规则、代缺省、存储过程、触发程序或视图的 Transact SQL 代码。如果创建了加密的对象，就读不出该对象的文本。

### 语法

```
sp_helptext { name }
```

变量 name 是当前数据库中数据库对象的 nvarchar(776)型名称。

## sp\_helptrigger

sp\_helptrigger 系统存储过程可显示位于当前数据库的表格中的触发程序的类型。

### 语法

```
sp_helptrigger { table } [ , type ]
```

变量

table

type

table 是 nvarchar(776)型的表名。

type 是字符(6)型的触发器类型，其中 DELETE=删除触发器，INSERT=插入触发器，UPDATE=更新触发器。缺省值为 NULL。

## sp\_helpuser

sp\_helpuser 系统存储过程可显示当前数据库中的数据库用户的信息。

语法

```
sp_helpuser [[ @name_in_db = ] security_account ]
```

变量 security\_account 是 SQL Server 用户, Windows NT 用户或数据库角色的系统名称。如果没有提供此变量，将显示当前数据库中所有安全帐户的信息。

## sp\_indexes

sp\_indexes 系统存储过程可显示远程表格中的索引信息，如果 table\_name 变量或 table\_schema 变量中有一个是 NULL，那么 sp\_indexes 只返回对应列的值为 NULL 的行。

语法

```
sp_indexes { table_server } [ , table_name ] [ , table_schema ] [ ,
```

```
table_db    ][ ,    index    ][ ,    is_unique    ]
```

变量

table_server	table_server 是远程表格服务器的系统名称。
table_name	table_name 是远程表格的系统名称。用“%”表示所有数据库表格。
table_schema	table_schema 是表格模式的系统名称。用“%”表示所有模式或所有数据库。
table_db	table_db 是数据库的名称。
index	index 是索引的系统名称。
is_unique	is_unique 是一个位标志，其中 1 只返回唯一索引的信息，0 返回全部索引信息。

## sp\_indexoption

sp\_indexoption 系统存储过程允许为用户建立的索引选择一种锁定类型。这可就高级用户而言，因为 Microsoft SQL Server 会自动执行上述操作。

语法

```
sp_indexoption { [ @IndexNamePattern = ]          index_name          }  
[ , [ @OptionName = ]          option_name          ]  
[ , [ @Option_Name = ]          value          ]
```

变量

index_name	index_name 是用户建立的表格或索引的 nvarchar(1035)型名称。如果选择用某个数据库名称限制索引名称,该数据库名称必须是当前数据库。给出没有索引名的表格名称,将设置表格中所有索引的选项。
option_name	option_name 是 varchar(35)型数据,其中 AllowRowLocks=当“value”为 TRUE 时,使用行锁定(若将“value”设置为 FALSE,就只使用页面锁定和表格锁定)。AllowPageLocks=当“value”为 FALSE 时,不使用页面锁定(对指定的索引,只使用行锁定和表格锁定)。
value	value 是 varchar(12)型的设置,它确定“option_name”的值为 TRUE 或为 FALSE。如果该变量为缺省值 NULL,将显示“index_name”的当前设置。

## sp\_link\_publication

当与出版商联系,在 MSsubscription\_properties 系统表中建立推入推出订阅的条目时,sp\_link\_publication 系统存储过程可定义同步触发器采用的安全措施。用存储过程 sp\_help-subscription\_properties 和 sp\_subscription\_cleanup 可查看并清理 MSsubscription\_properties 系统表中的条目。

### 语法

```
sp_link_publication [ @publisher = ] publisher ,
```



```
[ @publisher_db = ] publisher_db , [ @publication = ]  
publication ,  
[ @security_mode = ] security_mode [ , [ @login = ] login ]  
[ , [ @password = ] password ]
```

变量

[ @publisher = ] publisher publisher 是出版商的系统名称。

[ @publisher\_db = ] publisher\_db publisher\_db 是出版商数据库的系统名称。

[ @publication = ] publication publication 是出版物的系统名称。

[ @security\_mode = ] security\_mode security\_mode 是与出版商联系时的整型安全模式。0 表示同步触发器利用动态远程过程调用与出版商的联系,2 表示同步触发器需用 sys.servers 系统表项目进行远程过程的调用。(用 sp\_addlinkedserver 可将项目添加至 sys.servers)。

[ @login = ] login login 是注册的系统名称。

[ @password = ] password password 是口令的系统名称。

## sp\_linkedservers

sp\_linkedservers 系统存储过程可显示本地服务器的链接服务器。  
语法

## sp\_lock

sp\_lock 系统存储过程可显示锁定的信息。用 sp\_who 存储过程可获取 spid 号。

语法

```
sp_lock [ spid1 ] [ , spid2 ]
```

变量

Spid1

spid1 是 sysprocesses 系统表中的整型过程 ID 号，如果没有提供此变量，将显示所有过程的信息。

Spid2

spid2 是要返回锁定信息的另一个过程 ID 号。

## sp\_makewebtask

sp\_makewebtask 系统存储过程通过建立任务产生 HTML 文档，HTML 文档存储了

执行查询时建立的数据。

模板文件用下列控制查询结果在模板中的布局的方法来格式化 HTML 文档：

- 可以用“在此插入数据”标识符。%insert\_data\_here%表明要在 HTML 表中添加查询结果的位置，用 @fixedfont, @bold, @italic, @colheaders 和 @tabborders 格式化查询结果

- 可以定义更精确完整的行格式，该格式会使系统忽略 sp\_makewebtask 中的 @colheaders, @reftext, @URL, @fixedfont, @resultstitle, @url\_query, @HTMLHeader, @singlerow, @webpagetitle, @italic 与 @tabborder 变量

- 关键词 <%begindetail%>和<%enddetail%>包括行格式

- <TR>,</TR>,<TD>,</TD>HTML 标记出现在行格式关键字之间

- <% insert\_data\_here%>用于行格式关键字内,以将每一列放在结果集中

## 语法

```
sp_makewebtask { [ @outputfile = ]          outputfile          } [ , [ @query = ]
  query          ]
[ , [ @fixedfont = ]      fixedfont      ] [ , [ @bold = ]      bold      ] [ , [ @italic
= ] italic      ]
[ , [ @colheaders = ]    colheaders    ] [ , [ @lastupdated = ]    lastupdated
  ]
[ , [ @HTMLHeader = ]    HTMLHeader    ] [ , [ @username = ]    username    ]
[ , [ @dbname = ]      dbname      ] [ , [ @templatefile = ]    templatefile
```

```

]
[ , [ @webpagetitle = ]      webpagetitle      ][ , [ @resultstitle = ]
resultstitle      ]
[[ , [ @URL = ]      URL      , [ @reftext = ]      reftext      ]
| [ , [ @table_urls = ]      table_urls      , [ @url_query = ]      url_query
]
]
[ , [ @whentype = ]      whentype      ][ , [ @targetdate = ]      targetdate      ]
[ , [ @targettime = ]      targettime      ][ , [ @dayflags = ]      dayflags      ]
[ , [ @numunits = ]      numunits      ][ , [ @unittype = ]      unittype      ]
[ , [ @procname = ]      procname      ][ , [ @maketask = ]      maketask      ]
[ , [ @rowcnt = ]      rowcnt      ][ , [ @tabborder = ]      tabborder      ]
[ , [ @singlerow = ]      singlerow      ][ , [ @blobfmt = ]      blobfmt      ]
[ , [ @nrowsperpage = ]      n      ][ , [ @datachg = ]      table_column_list      ]
[ , [ @charset = ]      characteraset      ]]
[ , [ @codepage= ]      codepage      ]

```

## 变量

@outputflie

outputflie 是 HTML 文档的 varchar(255)型路径和文件名。如果文件存储在远程机器上，就使用 UNC 名。否则文件将放在本地 SQL Server 中。

@query

@query 是要执行查询的文本型数据。可以定义多个

SELECT 查询。

- @fixedfont      @fixedfont 是 tinyint 型数据，其中 1=固定字体，0=比例字体，缺省值为 1。
- @bold            @bold 是 tinyint 型数据，其中 1=粗体字，0=非粗体字。缺省值为 0。
- @italic          @italic 是 tinyint 型数据，其中 1=斜字体，0=非斜字体，缺省值为 0。
- @colheaders     @colheaders 是 tinyint 型数据，其中 1=列标题，0=没有列标题。缺省值为 1。
- @lastupdated    @lastupdated 是 tinyint 型数据，其中 1=显示恰好在查询结果前的 HTML 页上的最后一次更新的日期和时间，0=不显示最后一次修改的日期和时间。缺省值为 1。
- @HTMLHeader    @HTMLHeader 是 tinyint 型数据，它表明@resultstitle 变量中内容的 HTML 格式码。其中 1=H1,2=H2,3=H3,4=H4,5=H5,6=H6。
- @username      username 是在该用户的许可下执行查询的 nvarchar(128)型数据。缺省值为当前数据库用户。
- @dbname         dbname 是执行查询所在的 nvarchar(128)型数据库名。缺省值是当前数据库。
- @templatefile   templatefile 是 HTML 模板文件的路径，在该模板文件中包括格式化资料与<%insert\_data\_here%>标志。
- @webpagetitle   @webpagetitle 是 HTML 文档的 varchar(255)型标题。

<code>@resultstitle</code>	<code>@resultstitle</code> 是查询结果的 <code>varchar(255)</code> 型标题，缺省值为“Query Results”。
<code>@URL</code>	<code>@URL</code> 是与放在文档结尾的另一个 HTML 文档的 <code>varchar(255)</code> 型连接。变量 <code>@reftext</code> 必须和此变量一块使用。但是不能使用变量 <code>@table_urls</code> 和 <code>@url_query</code> 。可以使用 <code>@URL</code> 和 <code>@reftext</code> 变量，也可以使用 <code>@url_query</code> 和 <code>@table_urls</code> 。
<code>@reftext</code>	<code>@reftext</code> 是 <code>varchar(255)</code> 型超级连接文本，也是目的文本。其地址是在 <code>@URL</code> 变量中定义的 URL。
<code>@table_urls</code>	<code>@table_urls</code> 是 <code>tinyint</code> 型数据，其中 1=超级连接在 HTML 文档中，且来自 <code>SELECT</code> 语句，（ <code>@url_query</code> 建立超级连接列表，不需给出 <code>@URL</code> 和 <code>@reftext</code> 变量。可使用 <code>@url_query</code> 和 <code>@table_urls</code> ）；0=没有创建 HTML 的超级连接的查询（缺省值）。
<code>@url_query</code>	<code>@url_query</code> 是 <code>varchar(255)</code> 型 <code>SELECT</code> 语句，以便从 SQL Server 表中生成多个 URL 和超级连接文本。可使用 <code>@url_query</code> 和 <code>@table_urls</code> 。 <code>@url_query</code> 变量返回两列：超级连接的地址和超级连接的说明。
<code>@whentype</code>	<code>@whentype</code> 是 <code>tinyint</code> 型数据，表示何时执行创建 HTML 文档的任务，其中 1=现在（缺省值），2=以后用 <code>@targetdate</code> 和 <code>@targettime</code> 以及此变量决定执行时间，3=从 <code>@targetdate</code> 开始，用 <code>@dayflags</code> 和 <code>@targettime</code>

在一星期的每几天建立页面(用 `sp_dropwebtask` 删除任务), 4=从 `@targetdate` 和 `@targettime` 开始, 每几分钟, 几小时, 几天, 几星期生成 Web 页(对 4 用 `@numunits` 和 `@unittype` 变量以及 4), 5=根据需要(使用不带自动调度的 `sp_runtask`), 6=现在生成 Web 页, 以后再用 `@targetdate` 和 `@targettime` 以及此变量确定时间, 7=现在生成页面, 并用 `@dayflags` 和 `@targettime` 在星期的每几天建立页面, 8=现在生成页面, 从现在和 `@targettime` 开始, 以后每几分钟, 几小时, 几天, 几星期定期生成 Web 页(用 `@numunits` 和 `@unittype` 变量), 9=现在根据需要生成 Web 页, 10=当数据库中的数据改变时立刻生成 Web 页面(使用 `@datachg` 变量)。

`@targetdate`

`@targetdate` 是生成页面的日期(格式为 `YYYYMMDD`)。如果没有提供此变量但需要目标日期, 就使用现在的日期。`@whentype` 为 2, 3, 4, 6 时需要 `@targetdate` 变量。

`@targettime`

`@targettime` 是以 `HHMMSS` 格式建立 HTML 文档的时间, 缺省值为 `12:00AM`。

`@dayflags`

`@dayflags` 是修改 HTML 文档的日期。可将值加在一起, 以安排数天的计划, 其中 1=星期天(缺省值), 2=星期一, 4=星期二, 8=星期三, 16=星期四, 32=星期五, 64=星期六(`@whentype` 为 3 和 7 时需要该变量)。

`@numunits`

`@numunits` 是 `tinyint` 型数据, 表示更新 HTML 文档的

频率，该变量可能的值是 1 到 255，缺省值为 1。此变量和 @whentype 为 4 和 8 以及 @unitttype 一起使用。

@unitttype

@unitttype 是 tinyint 型数据，表示应多长时间修改一次 HTML 文档，其中 1=小时(缺省值)，2=天，3=星期，4=分钟。此变量和 @whentype 为 4 和 8 以及 @numunits 一起使用。

@procname

@procname 是 HTML 文档的程序或任务的 varchar(128) 型名称，缺省值为 Web\_YYMMDDHHMMSS<spid>.varchar(28)。

@maketask

@maketask 是一整数，它确定是否创建生产 HTML 文档的任务，其中 0=建立不加密的存储过程，但不创建任务，1=建立加密的存储过程和任务，2=建立不加密的存储过程和任务(缺省值)。

@rowcnt

@rowcnt 是一整数，表示要在 HTML 文档中显示的最大行数，缺省值 0 表示所有的行。

@tabborder

@tabborder 是 tinyint 型数据，表示是否在结果表四周加上边界，1=是(缺省值)，0=否。

@singlerow

@singlerow 是 tinyint 型数据，表示结果是否以每页一行的形式出现，其中 1=是，0=否(缺省值)。

@blobfmt

@blobfmt 是 ntext 型数据，确定文本或图像数据类型的所有列是否应都放在同一结果页中，其中 NULL=是(缺省值)或将这些列保存到另一页，并用另一个 URT 以下述



格式将其连接至 HTML 文档：“ %n%FILE=output\_filename use TPLT=template\_filename URL=url\_link\_name... ”，其中 n=结果列表中与文本字段对应的列号，n+1 表示对另一文本或对图像 HTML 文档的超级连接。没有 template\_filename 时用 <%insert\_data\_here%>可确定放置数据的位置。url\_link\_name 是对文件的 http:// 连接。在 @blobfmt(%n%FILE=...) 中多个文本列或图像列可以重复使用同样的语法。不要把文件名放在问号中。

@nrowsperpage

@nrowsperpage 是一个整数，表示与 NEXT 和 PREVIOUS URLS 连接的每页中的行数。缺省值 0 表示所有页，此变量与 @singlerow 相互排斥。

@datachg

@datachg 是文本数据类型，当 @whentype 为 10 时使用该变量。它是重建 HTML 文档的表格和列的名称。语法为：{ TABLE=name [ COLUMN=name ] } [ ,... ] UPDATE, INSERT 和 DELETE 触发器建立在执行 Web 任务的表格中。sp\_runwebtask 可添加到任一已有的触发器的结尾（如果触发器还没有用密码编译，不让其作修改）。

@charset

@charset 是 nvarchar(25)型数据，表示浏览器使用的字符集别名代码。缺省值为 utf-8。

@codepage

@codepage 是与字符集有关的数值，缺省值是与缺省字符集 UTF-8 有关的 65001。使用 sp\_enumcodepages 可获

得代码页的列表。

## sp\_manage\_jobs\_by\_login

sp\_manage\_jobs\_by\_login 系统存储过程可更新注册拥有的作业。

### 语法

```
sp_manage_jobs_by_login { action } [ , current_owner_login_name ]  
[ , new_owner_login_name ]
```

### 变量

action

action 是要执行的 varchar(10) 型操作，其中 DELETE 删除所有 current\_owner\_login\_name 的作业，REASSIGN 将所有作业再分配给 new\_owner\_login\_name。

current\_owner\_login\_name

current\_owner\_login\_name 是一个系统名称，是当前拥有者的注册名称。

new\_owner\_login\_name

new\_owner\_login\_name 是一个系统名称，是新拥有者的注册名称。

## sp\_mergedummyupdate

sp\_mergedummyupdate 系统存储过程用以合并复制,以便将用 UPDATETEXT 或 WRITETEXT 对文本列所作的更新送至下一合并中。

### 语法

```
sp_mergedummyupdate [ @source_object = ] source_object ,  
[ @rowguid = ] rowguid
```

### 变量

[ @source\_object = ] source\_object source\_object 是源对象的  
nvarchar(386)型名称。

[ @rowguid = ] rowguid rowguid 是行标识符(唯一标识符)。

## sp\_mergesubscription\_cleanup

删除订户的合并预订后, sp\_mergesubscription\_cleanup 系统存储过程可清理 sysmergesubscription 和 sysmergearticle 系统表中的触发器和行。

### 语法

```
sp_mergesubscription_cleanup [ @publisher = ] publisher ,  
[ @publisher_db = ] publisher_db [ , [ @publication = ] publication  
]
```

## 变量

[ @publisher = ] publisher	publisher 是出版商的系统名称。
[ @publisher_db = ] publisher_db	publisher_db 是出版商数据库的系统名称。
[ @publication = ] publication	publication 是出版物的系统名称。

## sp\_montior

sp\_montior 系统存储过程可返回 SQL Server 如何使用资源的统计信息。

### 语法

```
sp_montior
```

## sp\_msx\_defect

sp\_msx\_defect 系统存储过程可编辑注册并从多重服务器操作中取出本地服务器。

### 语法

```
sp_msx_defect [ forced_defection ]
```

式中变量 forced\_defection 是位数据类型，其中 1=是，0=否。该变量可用于 Master Server Agent 丢失或遭破坏的情形。

如果使用 1，系统管理员(sa)必须执行下述命令：

```
EXECUTE msdb.dbo.sp_delete_targetserver @server_name = xxxxxxxx ,  
@post_defection = 0
```

## sp\_msx\_enlist

sp\_msx\_enlist 系统存储过程可编辑注册，并将本地服务器添加至多重服务器操作中。

### 语法

```
sp_msx_enlist { msx_server } [ , location ]
```

### 变量

msx\_server

msx\_server 是 varchar(30)型数据，是要添加的多重服务器管理的服务器名。

location

location 是 varchar(100)型数据，是用户提供的多重服务器管理服务器的位置。(例如 South Computer Room E45)。

## sp\_OACreate

sp\_OACreate 系统存储过程可在 Microsoft SQL Server 中建立 OLE 对象，用于 SQL-DMO。使用 progid 变量 (SQL-DMO.SQL Server) 或 clsid 变量 { 00026BA1-0000-0000-C000-000000000046 }，并将对象标志为整型局部变量 (DECLARE @object1 INT)，就可以建立 OLE 对象。

### 语法

```
sp_OACreate { progid | clsid } { , objecttoken OUTPUT } [ , context ]
```

### 变量

progid

progid 是以 OLEComponent.Object 格式建立的 OLE 对象的有效类的字符串，其中 OLEComponent 是 OLE Automation 服务器的名称。Object 是支持 IDispatch 界面的 OLE 对象名。(例如：SQLDMO.SQLServer)。

clsid

clsid 是以 { nnnnnnnnn-nnnnnnnn-  
nnnn-nnnn-nnnnnnnn } 格式建立的 OLE 对象的类标识符的字符串：例如，SQL-DMO SQL Server 对象的分类标识符 { 00026BA-0000-0000-C000-

objecttoken OUTPUT

Context

000000000046 } )。

objecttoken OUTPUT 是一个整型局部变量，是返回的对象标记。当调用其他 OLE Automation 存储过程时，该变量标识新建立的 OLE 对象。

context 是一个整数，表示已建立的 OLE 对象的运行环境。其中 1=在处理过程中，允许使用 a.dll 和 OLE 服务器，4=本地，允许使用 .exe 和 OLE 服务器，5=允许使用在处理过程中的本地 OLE 服务器(缺省值)，如果允许使用处理过程中的 OLE 服务器，就可以访问 SQL Server 拥有的资源，而本地 OLE 服务器不能访问任何 SQL Server 资源。

sp\_OADestroy

sp\_OADestroy 系统存储过程可删除用户建立的 OLE 对象。

语法

```
sp_OADestroy { objecttoken }
```

变量 objecttoken 是用户建立的 OLE 对象的对象标记。

## sp\_OAGetErrorInfo

sp\_OAGetErrorInfo 系统存储过程可返回 OLE Automation 错误信息。每次调用 OLE Automation 存储过程时，错误信息都会复位。

错误有：

- bad variable type(0x80020008)=作为方式参数传送的 SQL Server 数据类型与 Microsoft Visual Basic 数据类型不匹配，或传送了 NULL
- unknown name(0x80020006)=没有找到指定对象的指定属性或方式名称
- invalid class string(0x800401f3)=ProgID 或 CLSID 没有在 SQL Server 中注册为 OLE 对象
- server execution failed(0x80080005)=OLE 对象注册为本地 OLE 服务器和 .exe 文件注册，但找不到或不能启动它
- the module could not be found(0x8007007e)=OLE 对象注册为处理过程中的 OLE 服务器和 .DLL 文件，但找不到它
- type mismatch(0x80020005)=用于属性值或方式的本地变量的数据类型与 Visual 数据类型不匹配
- parameter not found(0x80020004)=变量在位置变量前定义
- the parameter is incorrect(0x80070057)=指定为 sp\_OACreate 的参数环境的数值无效

语法

```
sp_OAGetErrorInfo[ objecttoken [ , source OUTPUT [ , description
```



```
OUTPUT  
[ , helpfile OUTPUT [ , helpid OUTPUT ]]]]
```

### 变量

objecttoken	objecttoken 是用户建立的 OLE 对象的目标标记。
source OUTPUT	source OUTPUT 是一个局部字符或 varchar 变量，是错误信息源。
description OUTPUT	description OUTPUT 是一个局部字符或 varchar 变量，是对错误的说明。
helpfile OUTPUT	helpfile OUTPUT 是一个局部字符或 varchar 变量，是 OLE 对象的 Help 文件。
helpid OUTPUT	helpid OUTPUT 是一个局部整型变量，是 Help 文件的 context ID。

## sp\_OAGetProperty

sp\_OAGetProperty 系统存储过程可返回 OLE 对象的属性值。

### 语法

```
sp_OAGetProperty { objecttoken,propertyname } [ , propertyvalue  
OUTPUT ]  
[ , index ... ]
```

## 变量

objecttoken	objecttoken 是用户建立的 OLE 对象的目标标记。
Propertyname	propertyname 是属性的名称。
propertyvalue OUTPUT	propertyvalue OUTPUT 是存储返回属性值的正确数据类型的局部变量。
Index	index 是正确数据类型的值，索引参数象某些参数一样有一个或多个参数。

## sp\_OAMethod

sp\_OAMethod 系统存储过程可调用 OLE 对象的方法，也可用于获取属性值。对 OUTPUT 局部变量而言，应确保使用返回值的正确数据类型。

### 语法

```
sp_OAMethod { objecttoken, methodname } [ , returnvalue OUTPUT  
[ , [ @parametername = ] parameter [ OUTPUT ] [ ...n ] ] ]
```

### 变量

Objecttoken	objecttoken 是用户建立的 OLE 对象的目标标记。
Methodname	Methodname 是要调用的 OLE 对象的方法名称。
returnvalue OUTPUT	returnvalue OUTPUT 是一本地变量，它返回 OLE 对象的方法值。
[ @parametername = ]	parameter 是方法的参数。该参数必须是 Microsoft

parameter	Visual Basic 指定参数的名称。
[ OUTPUT ]	
N	n 表示前面的项可以重复。

## sp\_OASetProperty

sp\_OASetProperty 系统存储过程可将 OLE 对象的属性设置改为另一个值。

### 语法

```
sp_OASetProperty { objecttoken, propertyname, newvalue } [ , index... ]
```

### 变量

objecttoken	objecttoken 是用户建立的 OLE 对象的目标标记。
Propertyname	propertyname 是属性的名称。
Newvalue	newvalue 是属性改变后的值。
Index	index 是正确数据类型的值，索引参数象某些属性一样有一个或多个参数。

## sp\_OAStop

sp\_OAStop 系统存储过程可终止服务器的 OLE Automation 存储过程执行的环境。此存储过程不必显式调用，因为当 SQL Server 循环时，环境将自动终止。当用户话路调用 sp\_OAStop 时，环境就开启。如果有一个用户话路发布

sp\_OAStop ,那么所有其他的 SQL-DMO 程序都将失败 ,直到再次调用 sp\_OAStop 为止。

语法

sp\_OAStop

sp\_password

sp\_password 系统存储过程可更新 Microsoft SQL Server 注册的口令。

语法

```
sp_password[[ @old = ]      old_password      , ]{[ @new = ]      new_password  
      }
```

```
[ , [ @loginame = ]      login      ]
```

变量

[ @old = ] old\_password

old\_password 是旧口令的系统名称。

[ @new = ] new\_password

new\_password 是口令新值的系统名称。

[ @loginame = ] login

login 是 SQL Server 注册的系统名称。

sp\_pkeys

sp\_pkey 系统存储过程可显示当前数据库中表格的主键信息。

## 语法

```
sp_pkeys { name } [ , owner ] [ , qualifier ]
```

## 变量

`name` `name` 是表格的系统名称。

`owner` `owner` 是表的拥有者的系统名称。

`qualifier` `qualifier` 是表格限定符 (或 Microsoft SQL Server 数据库限定符) 的系统名称

## sp\_primarykeys

`sp_primarykeys` 系统存储过程可显示表格的主键列。

## 语法

```
sp_primarykeys { table_server } [ , table_name ] [ , table_schema ]
```

```
[ , table_catalog ]
```

## 变量

`table_server` `table_server` 是服务器的系统名称。

`table_name` `table_name` 的表格的系统名称，其中“%”用来表示全部表格。

`table_schema` `table_schema` 是表格模式的系统名称。用“%”表示所有数据库或全部模式。

table\_catalog

table\_catalog 是数据库的系统名称。

## sp\_post\_msx\_operation

sp\_post\_msx\_operation 系统存储过程用于提供 sysdownloadlist 系统表。此程序用来将目标服务器的作业操作传送至下载。

### 语法

```
sp_post_msx_operation{ operation }[, object ][, job_id  
]  
[, target_server ][, value ]
```

### 变量

operation

operation 是 varchar(64)型数据，是传送操作的类型。执行系统存储过程 sp\_help\_downloadlist 可以查看操作的可用值。

object

object 是操作的 varchar(64)型对象。缺省值为 JOB。当使用 SERVER 时，不需要使用 job\_id 变量。

job\_id

job\_id 是作业的唯一标识符，其中 0x00=全体作业。

target\_server

target\_server 是操作的目标服务器的 varchar(30)型名称。

Value

value 是一个整数，表示轮询的时间间隔，以秒为单位。

## sp\_processmail

sp\_processmail 系统存储过程可处理收件箱中进来的邮件信息，并将查询的结果集返回信息发送器，进来的邮件要求包括一个查询和一个已知的 Subject 变量。可以将其作为 SQL Server Agent 任务来设定，也可以将文件类型与服务器和接收客户机上的应用程序联接起来，以得到结果集中的正确图标。

### 语法

```
sp_processmail [[ @subject = ] subject ][ , [ @filetype = ] filetype
][ ,
[ @separator = ] separator ][ , [ @user = ] user ][ , [ @dbuse
= ] dbname ]
```

### 变量

subject	subject 是 SQL Server 的 varchar(255) 型主题行查询。
filetype	filetype 是 varchar(3) 型文件扩展名，用于发送回结果集。缺省值为 txt。
separator	separator 是结果集的 varchar(3) 型列分隔符，缺省值为 TAB。
user	user 是 有权执行查询的安全帐户的系统名称。缺省值为 客户。

dbname

dbname 是执行查询的数据库的系统名称，缺省值为主人。

## sp\_procoption

sp\_procoption 系统存储过程可定义程序选项。使用不带变量的该存储过程可以查看可用的选项。

语法

```
sp_procoption [ 变量 procedure ] [ , option ] [ , value ]
```

变量

procedure

procedure 是受影响的存储过程的 nvarchar(776) 型名称。

option

option 是要设置的 varchar(35) 型选项，其中 Startup= 将存储过程定义为启动存储过程，当 SQL Server 启动时，启动存储过程自动执行，Allow updates= 允许修改存储程序。

value

value 是 varchar(12) 型数据，其中 TRUE= 开，FALSE= 关。

## sp\_publication\_validation

sp\_publication\_validation 系统存储过程可用于快照和合并复制，它可确认文



章数据的有效性。可使用文章数据的行计数确认或全检查和确认。通过 SQL Server Agent 可以安排该存储程序定期运行。

### 语法

```
sp_publication_validation [ @publication = ] publication  
[ , [ @rowcount_only = ] rowcount_only ]  
[ , [ @full_or_fast = ] full_or_fast ]  
[ , [ @shutdown_agent = ] shutdown_agent ]
```

### 变量

```
[ @publication = ] publication  
[ @rowcount_only = ]  
rowcount_only
```

```
[ @full_or_fast = ] full_or_fast
```

```
[ @shutdown_agent = ]  
shutdown_agent
```

publication 是出版物的系统名称。

rowcount\_only 是一个位，其中 1=只返回表格的行计数，0=确认检查和的有效性。

full\_or\_fast 是 tinyint 型行计数计算类型，其中 0=全部计数(\*)，1=在 sysindexes.rows 中快速计数(可能产生不精确的行计数)，2=如果快速计数不合理，回复全计数。

shutdown\_agent 是一个位，其中 0=确认后关闭 Distribution 代理，1=不关闭 Distribution 代理。

## sp\_purgehistory

sp\_purgehistory 系统存储过程只向后兼容，它可从任务的历史表中删除信息。

### 语法

```
sp_purgehistory [ task_name ] [ , task_id ]
```

### 变量

task\_name task\_name 是任务的 varchar(100)型名称。

task\_id task\_id 是一个整数，是任务标识符。

## sp\_purge\_jobhistory

sp\_purge\_jobhistory 系统存储过程可删除作业历史记录行。如果不提供作业变量，将删除全部历史记录。只能提供一个变量，不能提供两个。

### 语法

```
sp_purge_jobhistory { [ @job_name = ] job_name | [ @job_id = ] job_id }
```

### 变量

job\_name job\_name 是要删除记录的作业的 varchar(100)型名称。

job\_id

job\_id 是作业的唯一标识符。

## sp\_reassigntask

sp\_reassigntask 系统存储过程只向后兼容，更详细的资料参阅版本 6.5 中的文本。sp\_recompile 在系统存储过程和触发器下一次执行时，sp\_recompile 系统存储过程为重新编译的表格标记存储过程和触发器(从 syscomments 系统表中读取，将程序放在高速缓存内)。

### 语法

```
sp_recompile { table }
```

变量 table 是 nvarchar(776) 型表名。

## sp\_refreshsubscriptions

sp\_refreshsubscriptions 系统存储过程可用于推出复制(快照、事务处理、合并)，为所有的订户将订阅添加到文章中。

### 语法

```
sp_refreshsubscriptions [ @publication = ] publication
```

变量 publication 是出版物的系统名称。

## sp\_refreshview

sp\_refreshview 系统存储过程可由于基本从属对象下的变化刷新视图。

### 语法

```
sp_refreshview [ @viewname = ] viewname
```

变量 viewname 是视图的 nvarchar 型名称。

## sp\_reinitmergepullsubscription

sp\_reinitmergepullsubscription 系统存储过程在 Merge Agent 运行时为重新初始化标记合并推出预订。

### 语法

```
sp_reinitmergepullsubscription [ @publisher = ] publisher  
[ @publisher_db = ] publisher_db , [ @publication = ] publication
```

### 变量

[ @publisher = ] publisher

publisher 是出版商的系统名称。

[ @publisher\_db = ] publisher\_db

publisher\_db 是出版商数据库的系统名称。

publisher\_db

[ @publication = ] publication

publication 是出版物的系统名称。

## sp\_reinitmergesubscription

sp\_reinitmergesubscription 系统存储过程可在 Merge Agent 运行时为重新初始化标记合并预订。它可以从 Publisher 中调用，也可重新执行 Snapshot 代理。

### 语法

```
sp_reinitmergesubscription [ @publisher = ] publisher ,  
[ @publisher_db = ] publisher_db , [ @publication = ] publication
```

### 变量

[ @publisher = ] publisher	publisher 是出版商的系统名称。
[ @publisher_db = ] publisher_db	publisher_db 是出版商数据库的系统名称。
[ @publication = ] publication	publication 是出版物的系统名称。

## sp\_reinitpullsubscription

sp\_reinitpullsubscription 系统存储过程可在 Merge Agent 运行时为重新初始化标记合并预订。它可以从订户端调用（除了 non\_immediate\_sync 出版物）。

### 语法

```
sp_reinitpullsubscription [ @publisher = ] publisher ,
```

```
[ @publisher_db = ] publisher_db , [ @publication = ] publication
```

### 变量

```
[ @publisher = ] publisher publisher 是出版商的系统名称。  
[ @publisher_db = ] publisher_db publisher_db 是出版商数据库的系统  
publisher_db 名称。  
[ @publication = ] publication publication 是出版物的系统名称。
```

## sp\_reinitsubscription

sp\_reinitsubscription 系统存储过程使快照、事务复制和合并复制同步。如果包含 nonimmediate\_sync 出版物的 auto\_sync 预订，也可执行 Snapshot Agent。如果包含 no\_sync 预订，就负责保证订户的目标表格同步，保证没有未送达的复制指令。使用订户变量的 all 或 NULL 使匿名订户同步。

### 语法

```
sp_reinitsubscription [ @publication = ] publication , [ @article = ]  
article  
[ @subscriber = ] subscriber , [ @destination_db = ]  
destination_db
```

### 变量

[ @publication = ] publication

[ @article = ] article

publication 是出版物的系统名称。

article 是文章的系统名称。对 immediate\_sync 出版物，可使用缺省值 All。

[ @subscriber = ] subscriber

subscriber 是订户的系统名称。缺省值为 ALL，NULL 用于匿名预订。

[ @destination\_db = ]

destination\_db

destination\_db 是目标数据库的系统名称，缺省值为 all。

## sp\_remotoption

sp\_remotoption 系统存储过程可为在运行 Microsoft SQL Server 的本地服务器上定义的远程注册显示或改变选项。

### 语法

```
sp_remotoption [ , [ @remoteserver = ]      remoteserver      ]  
[ , [ @loginame = ]      loginame      ][ , [ @remotename = ]      remotename  
  ]  
[ , [ @optname = ]      optname      ][ , [ @optvalue = ]      optvalue      ]
```

### 变量

[ @remoteserver = ] remoteserver

remoteserver 是远程服务器的系统名称

[ @loginame = ] loginame

loginame 是用户在本地 SQL Server 上注册的系统名称。

[ @remotename = ] remotename

remotename 是用户在 remoteserver 上注册的系统名称。

[ @optname = ] optname

optname 是要设置的选项的 varchar(35) 型名称。

[ @optvalue = ] optvalue

optvalue 是 optname 的 varchar(10) 型值，缺省值为 NULL，其中 TRUE=委托(集成注册)，FALSE=非委托(标准注册)。

## sp\_removedbreplication

当恢复复制的数据库且不需要用该数据库恢复复制的对象时使用 sp\_removedbreplication 系统存储过程。该存储过程可从数据库中删除所有的复制对象。分配器数据库中没有更新数据。

### 语法

sp\_removedbreplication [ @dbname = ] dbname

变量 dbname 是数据库的系统名称。

## sp\_remove\_job\_from\_targets

sp\_remove\_job\_from\_targets 系统存储过程可从目标服务器或组中删去作



业。需提供 job\_id 变量或 job\_name 变量，但不需两者都提供。

#### 语法

```
sp_remove_job_from_targets {      job_id |      job_name      } [ ,  
target_server_groups  ]  
[ ,      target_servers      ]
```

#### 变量

job\_id

job\_id 是作业的唯一标识符。

job\_name

job\_name 是作业的系统名称。

target\_server\_groups

target\_server\_groups 是 nvarchar(1024) 型数据，它包含了目标服务器组的列表。多个组用逗号分开。

target\_servers

target\_servers 是 nvarchar(1024) 型数据，它包含了目标服务器的列表。多个服务器用逗号分开。

## sp\_rename

sp\_rename 系统存储过程给用户建立的现有对象起一个新名。可在当前数据库中 使用此程序。

#### 语法

```
sp_rename {      object_name      } [ ,      new_name      ] [ ,      object_type      ]
```

## 变量

<code>object_name</code>	<code>object_name</code> 是对象的 <code>nvarchar(776)</code> 型旧名称。对于列或索引，可使用 <code>table.column</code> 或 <code>table.index</code> 。
<code>new_name</code>	<code>new_name</code> 是赋予对象的新的系统名称。如果是列或索引，此处不要使用限定的表名。
<code>object_type</code>	<code>object_type</code> 是 <code>varchar(13)</code> 型数据，它决定对象的类型，其中 <code>COLUMN=列</code> ， <code>DATABASE=用户建立的数据库</code> ， <code>INDEX=用户建立的索引</code> ， <code>OBJECT=sysobjects</code> 系统表中的对象类型项， <code>USERDATATYPE=用户定义的数据类型</code> 。

## `sp_renamedb`

`sp_renamedb` 系统存储过程为用户建立的现有数据库起一个新名(数据库必须在单用户模式下才能重命名)。

## 语法

```
sp_renamedb {      old_name      ,      new_name      }
```

## 变量

<code>old_name</code>	<code>old_name</code> 是数据库旧的系统名称。
<code>new_name</code>	<code>new_name</code> 是赋予数据库的新系统名称。

## sp\_replcmds

sp\_replcmds 系统存储过程用于事务复制，此事务复制指定了数据库中第一个执行 sp\_replcmds 存储过程的客户机将成为日志的阅读器。此存储过程由 Logreader Agent 使用。它返回事务的指令，以及事务日志中在发送到分配器时还没有指定的复制事务信息。它也从其运行之处返回出版数据库的信息。每次，只能有一个用户在数据库中执行此存储过程(错误 18752)。只在需要时执行此程序以进行故障检修的复制。

如果不能复制文本命令，将出现错误 18759，因为同一事务中不能检索文本指针。

### 语法

```
sp_replcmds [ @maxtrans = ] maxtrans
```

变量 maxtrans 是一个整数，是要检验的事务数。缺省值为一个事务。

## sp\_replcounters

sp\_replcounters 系统存储过程用于事务复制，以显示以下信息：未传输事务的等待时间，吞吐量及事务的计数，以及 SQL Server 上出版数据库的生成列表每秒钟处理的事务。

### 语法

```
sp_replcounters
```

## sp\_repldone

sp\_repldone 系统存储过程由 Logreader Agent 用以事物处理复制,以便对标识 SQL Server 最后一个分布式事物处理的记录作标记。除非知道如何操作,否则不要手动执行此程序,但可以使用 sp\_repldone 系统存储过程中断复制过程。而且,使用变量 xactid 和 xact\_seqno 可以标记事务已发送至分配器,或将事务指定为下一个要复制的事务。运行 sp\_repltrans 或 sp\_replcmd 系统存储过程可以获取变量。所有小于或等于该事务的事务都指定为分布式事务。要将事务日志中的所有复制的事务都指定为分布式事务,须指定 sp\_repldone NULL, NULL, 0, 0, 1。要查看事务日志中的复制事务,就指定 sp\_repldone NULL, NULL, 0, 0, 0。

### 语法

```
sp_repldone [ @xactid = ] xactid , [ @xact_seqno = ] xact_seqno  
[ , [ @numtrans = ] numtrans ] [ , [ @time = ] time [ , [ @reset  
= ] reset ]
```

### 变量

[ @xactid= ] xactid

xactid 是最后一个分布式事务的第一个记录的记录序号。

[ @xact\_seqno= ] xact\_seqno

xact\_seqno 是最后一个分布式事务的最后一个记录的序号。

[ @numtrans= ] numtrans

numtrans 是分布式事务的数目。

[ @time= ] time

[ @reset= ] reset

time 是分配最后一批事务所需要的毫秒数。

reset 是一个位型重置状态，其中 1= 标记日志中已分配的所有复制事务，0= 将事务日志复位至第一个复制事务，不将复制事务指定为分布式事务。此变量要将 xactid 和 xact\_seqno 设为 NULL。

## sp\_replflush

sp\_replflush 系统存储过程用于快照、事务和合并复制，以刷新文章高速缓存。当修改了文章的定义时，将由其它的复制存储过程使用该程序，但通常不手动执行它。

语法

sp\_replflush

## sp\_replicationdboption

sp\_replicationdboption 系统存储过程用于当前数据库中的快照、事务及合并复制，以设置复制数据库选项。该系统存储过程设置 sysdatabases 系统表中的

类别位，并建立复制系统表。

语法

```
sp_replicationdboption[ @dbname= ]    dbname    ,[ @optname= ]    optname  
    , [ @value= ]  
    value    [ , [ @ignore_distributor= ]    ignore_distributor    ]
```

变量

[ @dbname = ] dbname

dbname 是要删除的数据库的系统名称。

[ @optname = ] optname

optname 是要建立或要删除的选项的系统名称，其中 merge publish=数据库供合并出版，publish=数据库供其它类型的出版。

[ @value = ] value

value 是存储 true 或 false 的系统名称。(false 也删除合并预订)。

[ @ignore\_distributor = ]  
ignore\_distributor ]

ignore\_distributor 是一个位，其中 1=true，0=false(缺省值)。

## sp\_replicationoption

sp\_replicationoption 系统存储过程可添加或撤消复制存储过程，用于快照、事务与合并复制，以启动复制。

## 语法

```
sp_replicationoption [ @optname = ] optname ,  
[ @value = ] value [ , [ @security_mode = ] security_mode ] [ ,  
[ @login = ] login  
[ , [ @password = ] password ] [ , [ @reserved = ] reserved ]
```

### 变量

[ @optname = ] optname

optname 是复制选项的系统名称，其中 transactional=事务复制，merge=合并复制。

[ @value = ] value

value 是 nvarchar(5)型数据，其中 true=建立选项，false=删除选项。

[ @security\_mode = ] security\_mode

security\_mode 是一个整数，是分配器的安全模式，其中 0=SQL Server 验证(缺省值)，1=Windows NT 验证。

[ @login = ] login

login 是分配器注册的系统名称，缺省值为系统管理员(sa)。

[ @password = ] password

password 是分配器注册的口令的系统名称。

[ @reserved = ] reserved

reserved 用于 Microsoft SQL Server，缺省值为 NULL。

## sp\_replication\_agent\_checkup

sp\_replication\_agent\_checkup 系统存储过程用于快照、事务以及合并复制，以检查正在执行复制代理的分布数据库，但在 heartbeat\_interval 中还没有写入历史表。该存储过程将在适当时引发错误 14151。

### 语法

```
sp_replication_agent_checkup [ [ @heartbeat_interval = ]  
heartbeat_interval ]
```

变量 heartbeat\_interval 是代理在不需写入历史表，且不引发错误的情况下能正常运行的分钟数，缺省值为 10 分钟。

## sp\_replshowcmds

sp\_replshowcmds 系统存储过程可显示标志为复制的事务指令，且只有当所有用户联接都没有访问复制记录事务时，才能使用该存储过程。

### 语法

```
sp_replshowcmds [ @maxtrans = ] maxtrans
```

变量 [ @maxtrans = ] maxtrans 是一个整数，表示要查看其信息的事务数，缺省值为 1。



## sp\_repltrans

sp\_repltrans 系统存储过程用于事务的复制，以显示出版数据库事务日志中还没有指定为分布式的复制事务列表。在出版数据库中执行此存储过程。它和 sp\_replcmds 类似，但不返回事务指令。

### 语法

```
sp_repltrans
```

## sp\_resync\_targetserver

sp\_resync\_targetserver 系统存储过程可使多重服务器域内的服务器同步。

### 语法

```
sp_resync_targetserver { server }
```

变量 server 是服务器的 nvarchar(30) 型名称，其中，ALL 使所有的服务器在多重服务器域中重新同步。

## sp\_revoke\_publication\_access

sp\_revoke\_publication\_access 系统存储过程用于快照、事务和合并复制，以从出版访问列表中删除注册。

## 语法

```
sp_revoke_publication_access [ @publication = ]      publication      ,  
[ @login = ]      login
```

## 变量

```
[ @publication = ]      publication      publication 是出版物的系统名称，  
                        如果没有提供此变量，将从服务器全局访问列表中删除注册。  
[ @login = ]      login      login 是注册的系统名称。
```

## sp\_revokedbaccess

sp\_revokedbaccess 系统存储过程可从当前数据库中删除任一安全帐户、许可和别名。

## 语法

```
sp_revokedbaccess{ [ @name_in_db = ]      name      }
```

变量 name 是已删除的数据库用户的系统名称。

## sp\_revokelogin

sp\_revokelogin 系统存储过程可删除 Microsoft Windows NT 用户或组的注册项，这些用户或组是在执行注册的 sp\_grantlogin 与 sp\_denylogin 存储过程时添

加的。

语法

```
sp_revokelogin{ [ @loginame = ] login }
```

变量 login 是 Windows NT 用户或组的系统名称，其格式为 Domain \ user。

### sp\_runwebtask

sp\_runwebtask 系统存储过程可执行现有的 Web 任务并生成 HTML 文档。

语法

```
sp_runwebtask [[ @procname = ] procname ] [ ,@outputfile = ]  
outputfile ]
```

变量

procname            procname 是属于 Web 任务的存储过程的 nvarchar(128)型名称。

outputfile          outputfile 是 Web 任务输出文件的 nvarchar(255)型名称。

### sp\_script\_synctrans\_commands

sp\_script\_synctrans\_commands 系统存储过程用于快照和事务复制，为每个订户上的应用程序的文章用 sp\_addsyntrigger 调用建立脚本。

## 语法

```
sp_script_synctrans_commands [ @publication = ]      publication  
[ , [ @article = ]      article      ]
```

## 变量

[ @publication = ] publication publication 是出版物的系统名称。

[ @article = ] article article 是文章的系统名称，缺省值为 ALL。

## sp\_scriptdelproc

sp\_scriptdelproc 系统存储过程用于快照和事务复制，为包含删除语句的定制存储过程自动建立 CREATE PROCEDURE 语句。

## 语法

```
sp_scriptdelproc [ @artid = ]      artid
```

变量 artid 是标识文章的整数。

## sp\_scriptinsproc

sp\_scriptinsproc 系统存储过程用于快照与事务复制，为包含插入语句的定制存储过程自动建立 CREATE PROCEDURE 语句。

## 语法

```
sp_scriptinsproc [ @artid = ]      artid
```

变量 artid 是标识文章的整数。

## sp\_scriptupdproc

sp\_scriptupdproc 系统存储过程用于快照与事务复制，为包含更新语句的定制存储过程自动建立 CREATE PROCEDURE 语句。

## 语法

```
sp_scriptupdproc [ @artid = ]      artid
```

变量 artid 是标识文章的整数。

## sp\_server\_info

sp\_server\_info 系统存储过程可显示有关 Microsoft SQL Server、数据库网关或者其他数据源的信息。

## 语法

```
sp_server_info [ [ @attribute_id = ]      attribute_id      ]
```

变量 attribute\_id 是属性的整型标识符。

## sp\_serveroption

sp\_serveroption 系统存储过程定义远程服务器与链接服务器的服务器选项，并且定义分布式查询的选项。

### 语法

```
sp_serveroption [[ @server = ] server ,  
[ @optname = ] option_name ,  
[ @optvalue = ] option_value ]
```

### 变量

server

server 是服务器的系统名称，如果没有提供此变量，将显示服务器选项列表。

option\_name

option\_name 是选项的 varchar(35) 型名称。如果没有提供此变量，将显示服务器选项列表，其中：collation compatible=true 表示当对链接服务器执行该存储过程时，指定分布式查询执行为兼容式查询，false 和 SQL Server 在本地比较，data access=启用

option\_value

或禁用分布式查询访问的链接服务器；dist=分配器，dpub=此分配器的远程出版商，pub=出版商，rpc=true表示自服务器的 Remote Procedure Calls，rpc out= true 表示到服务器的 Remote Procedure Calls；sub=订户，system=Microsoft SQL Server 使用。option\_value 是 varchar(10)型数据，它决定选项是否表示 True=开启或 False=关闭。如果没有提供此变量，将显示选项值。

sp\_setapprole

sp\_setapprole 系统存储过程可使与用户有关的其他许可被忽略，而数据库中应用程序的角色有效，如果设置了应用程序角色，将只允许使用应用程序角色许可。

语法

```
sp_setapprole { [ @rolename = ] name , [ @password = ] password  
              }  
[ , [ @encrypt = ] encrypt_style ]
```

变量

[ @rolename = ]	name	name 是数据库中应用程序角色的系统名称。
[ @password = ]	password	password 是系统名称，是应用程序角色的口令。
[ @encrypt = ]	encrypt_style	encrypt_style 是 varchar(10)型数据，它决定加密类型，其中 none=不加密口令(缺省值)，ODBC=用 ODBC 规范加密函数为口令加密。

## sp\_setnetname

sp\_setnetname 系统存储过程用于将远程存储过程执行过程中有非法字符的远程服务器名，万一远程服务器的名称与 SQL Server 允许在表格中放置的名称不兼容，就将名称放在 sysservers 列或 srvnetname 中。

### 语法

```
sp_setnetname { server } [ , network_name ]
```

### 变量

server	server 是用于远程存储过程调用的远程服务器的系统名称，服务器必须提前添加至 sysservers 系统表中。
network_name	network_name 是远程计算机的网络名称。



## sp\_spaceused

sp\_spaceused 系统存储过程可返回表格或数据库使用的磁盘空间信息。updateusage 变量将对 sysindexes 表进行更新，以防 sysindexes 表中的信息陈旧时，空格信息为最新的，注意在大表格中使用此变量，它可能需要一定的时间。

### 语法

```
sp_spaceused [      objname      ][      updateusage      ]
```

### 变量

objname                    objname 是表格或数据库的 nvarchar(776)型名称。缺省值 NULL 表示在数据库中执行此存储过程。

updateusage                updateusage 是 varchar(5)型数据，它可以为 true 或 false(缺省值)，决定 DBCC UPDATEUSAGE 是否在表格或数据库中运行。

## sp\_special\_columns

sp\_special\_columns 系统存储过程可显示表中的专用标识列和自动更新的列。

### 语法

```
sp_special_columns {      name      }[ ,      owner      ][ ,      qualifier  
][ ,      col_type      ]
```

[ , scope ]  
 [ , nullable ][ , ODBCVer ]  
 变量

name	name 是表格的系统名称。
owner	owner 是表格拥有者的系统名称。
qualifier	qualifier 是表格限定符(SQL Server 的数据库名称)的系统名称。
col_type	col_type 是字符(1)型数据，是列的类型，其中 R=返回专用标识列，V=返回表格中自动更新的列。
scope	scope 是字符(1)型数据，表示 ROWID 的作用域。其中 C=定位在行中时，ROWID 有效，T=事务处理期间 ROWID 有效(缺省值)。
nullable	nullable 是字符(1)型数据，其中 U=专用列允许空值(缺省值)，0=专用列不允许空值。
ODBCVer	ODBCVer 是表示当前使用的 ODBC 版本的整数，缺省值为 2(ODBC 版本 2.0)。

## sp\_sproc\_columns

sp\_sproc\_columns 系统存储过程可显示当前数据库中存储过程的变量和结

果集信息。

语法

```
sp_sproc_columns [ name ] [ , owner ] [ , qualifier ]  
 [ , column_name ]  
 [ , ODBCVer ]
```

变量

name	name 是存储过程的 nvarchar(390)型名称，缺省值 % 表示当前数据库中的所有存储程序。
owner	owner 是存储过程拥有者的 nvarchar(384)型名称。
qualifier	qualifier 是程序限定符 (SQL Server 中的数据库名称) 的系统名称。
column_name	column_name 是列的 nvarchar(384)型名称。如果没有提供该变量，将返回所有的列。该程序支持 “%” 和 _ 通配符。
ODBCVer	ODBCVer 是表示当前使用的 ODBC 版本的整数，缺省值为 2 (ODBC 版本 2.0)。

## sp\_srvrolepermission

sp\_srvrolepermission 系统存储过程可显示固定服务器角色的许可。

语法

```
sp_srvrolepermission [ [ @srvrolename = ]          role          ]
```

变量 `role` 是固定服务器角色的名称，可能的值是：`sysadmin`、`securityadmin`、`serveradmin`、`setupadmin`、`processadmin`、`diskadmin` 和 `dbcreator`。

## sp\_start\_job

`sp_start_job` 系统存储过程用于启动作业的运行，可使用 `job_name` 变量或 `job_id` 变量，但不能两者都使用。

### 语法

```
sp_start_job { [ @job_name = ]          job_name          | [ @job_id = ]          job_id          ]
```

```
[ , [ @error_flag = ]          error_flag          ]
```

```
[ , [ @server_name = ]          server_name          ]
```

```
[ , [ @step_name = ]          step_name          ]
```

```
[ , [ @output_flag = ]          output_flag          ]
```

### 变量

`job_name` `job_name` 是作业的系统名称。

`job_id` `job_id` 是作业的唯一标识符。缺省值为 `NULL`。

`error_flag` `error_flag` 专供 SQL Server 使用。

`server_name` `server_name` 是运行作业的服务器的 `nvarchar(30)` 型名称。缺省值为 `NULL`。

`step_name` `step_name` 是执行开始处步骤的系统名称，缺省值为

NULL。

output\_flag

output\_flag 专供 SQL Server 使用。

## sp\_statistics

sp\_statistics 系统存储过程可显示表格中所有的索引。

### 语法

```
sp_statistics { name } [ , owner ] [ , qualifier ] [ ,  
index_pattern ]  
[ , is_unique ]  
[ , accuracy ]
```

### 变量

name

name 是表格的系统名称。

owner

owner 是表格拥有者的系统名称。

qualifier

qualifier 是表格限定符 (SQL Server 的数据库名称) 的系统名称。

index\_pattern

index\_pattern 是索引的系统名称，缺省值 % 表示全部索引。

is\_unique

is\_unique 是字符 (1) 型数据，其中 Y=返回专用，N=返回非专用。

accuracy

accuracy 是字符 (1) 型数据，是统计信息的页精确度，其

中 E=确保更新统计，Q=不确保更新统计（缺省值）。

## sp\_stop\_job

sp\_stop\_job 系统存储过程可用于取消作业的执行。如果作业类型为 CmdExec，使用此存储过程时要注意，因为可能产生预料不到的行为——包括遗漏当前文件。可使用 job\_name 或者 job\_id 变量，不能两个变量都使用。

### 语法

```
sp_stop_job [ @job_name = ]          job_name          | [ @job_id = ]      job_id  
[ @originating_server = ]            server              | [ @server_name = ] server_name
```

### 变量

job_name	job_name 是作业的系统名称。
job_id	job_id 是作业的唯一标识符，其缺省值为 NULL。
originating_server	originating_server 是启动作业的服务器的 nvarchar(30)型名称。
server_name	server_name 是执行作业的服务器的 nvarchar(30)型名称，缺省值为 NULL。

## sp\_stored\_procedures

sp\_stored\_procedures 系统存储过程可显示当前数据库中的存储过程。

### 语法

```
sp_stored_procedures [ name ] [ , owner ] [ , qualifier ]  
变量
```

**name** name 是存储过程的 nvarchar(390)型名称。支持 % 和通配符。  
**owner** owner 是存储过程拥有者的 nvarchar(384)型名称。支持 % 和通配符。  
**qualifier** qualifier 是表格限定符 (SQL Server 的数据库名称) 的系统名称。

## sp\_subscription\_cleanup

sp\_subscription\_cleanup 系统存储程序用于各种类型的复制，清理 MSreplication\_subscriptions 复制表中的项目，也用于同步事务预订。此存储过程也可清理同步事务触发器。

### 语法

```
sp_subscription_cleanup [ @publisher = ] publisher ,  
[ @publisher_db = ] publisher_db [ , [ @publication = ]
```

publication ]

变量

[ @publisher = ]	publisher	publisher 是出版商的系统名称。
[ @publisher_db ]	publisher_db	publisher_db 是出版商数据库的系统名称。
[ @publication = ]	publication	publication 是出版物的系统名称，缺省值为 NULL。

sp\_table\_privileges

sp\_table\_privileges 系统存储过程可显示当前数据库的表格及其许可。

语法

```
sp_table_privileges { table_name_pattern } [ , table_owner_pattern ]
```

```
[ , table_qualifier ]
```

变量

table_name_pattern	table_name_pattern 是表格的 nvarchar(384)型名称。它没有缺省值，支持 % 和通配符。
table_owner_pattern	table_owner_pattern 是表格拥有者的 nvarchar(384)型名称。支持 % 和通配符。
table_qualifier	table_qualifier 是表格限定符 (SQL Server 的数据库



名称)的系统名称。

## sp\_table\_privileges\_ex

sp\_table\_privileges\_ex 系统存储过程可在链接服务器上显示数据库中的表格及其许可。

### 语法

```
sp_table_privileges_ex { table_server } [ , table_name ] [ ,  
table_schema ]  
[ , table_catalog ]
```

### 变量

table_server	table_server 是链接服务器的系统名称。
table_name	table_name 是表格的系统名称。
table_schema	table_schema 是表格模式的系统名称。
table_catalog	table_catalog 是表格数据库的系统名称。

## sp\_table\_validation

sp\_table\_validation 系统存储过程用于复制，以返回表格的行计数信息或检查和信息。也可通过显示预期行数与实际行数间的差别，来确定表中所提供的行计数信息与检查和信息的差别。

## 语法

```
sp_table_validation [ @table = ]      table
[ , [ @expected_rowcount = ]          expected_rowcount    ]
[ , [ @expected_checksum = ]          expected_checksum    ]
[ , [ @rowcount_only = ]              rowcount_only        ][ , [ @owner = ]      owner
]
[ , [ @full_or_fast = ]               full_or_fast        ]
[ , [ @shutdown_agent = ]             shutdown_agent      ]
```

### 变量

```
[ @table = ]      table
[      @expected_rowcount      ]
expected_rowcount

[      @expected_checksum      =    ]
expected_checksum

[      @rowcount_only          =    ]
rowcount_only

[ @owner = ]      owner
[ @full_or_fast = ]      full_or_fast
```

table 是表格的系统名称。

expected\_rowcount 是一个整数，其中 NULL=返回实际的行计数，value=对照实际的行计数检查的值。

expected\_checksum 是一个数值，其中 NULL=返回实际检查和和数值，value=对照实际检查和检查的值。

rowcount\_only 是一个位，其中 1=只返回表格的行计数，0=返回行计数与检查和（缺省值）。

owner 是表格拥有者的系统名称。

full\_or\_fast 是 tinyint 型数据，表示如

何确定行计数，其中 0=用 COUNT(\*) 进行全体计数，1=从 sysindexes.rows 中快速记数，但只精确到最后一次更新 sysindexes，2=条件快速计数——如删除了分歧，就使用全体计数。

shutdown\_agent 是一个位，其中 0=不关闭复制代理，1=关闭复制代理。

```
[ @shutdown_agent ]  
shutdown_agent
```

## sp\_tableoption

sp\_tableoption 系统存储过程用于为用户建立的表格设置选项。

语法

```
sp_tableoption{ table } [ , option_name ] [ , value ]
```

变量

table table 是用户建立的数据库表格的 nvarchar(776)型名称。

option\_name option\_name 是 varchar(35)型数据，是选项的名称，其中 NULL=列出所有选项及其缺省值。

value value 是 varchar(12)型数据，它决定选项是开启还是关闭的，其中 true=开启，false=关闭。

sp\_tables 系统存储过程可显示当前数据库中表格对象的列表。

语法

## sp\_tables

sp\_tables [ name ] [ , owner ] [ , qualifier ] [ , type ]  
变量

name	name 是表格的 nvarchar(384)型名称，缺省值为 NULL，支持 % 和通配符。
owner	owner 是 nvarchar(384)型的表格拥有者。支持 % 和通配符。
qualifier	qualifier 是表格限定符的系统名称，缺省值为 NULL。
type	type 是 varchar(100)型数据，是用逗号分隔开的表格类型列表 (TABLE, SYSTEM TABLE 或 VIEW)。

## sp\_tables\_ex

sp\_tables\_ex 系统存储过程可从指定的链接服务器中显示可查询的对象列表。

### 语法

sp\_tables\_ex { table\_server } [ , table\_name ] [ , table\_schema ]  
[ , table\_catalog ] [ , table\_type ]

### 变量

table\_server table\_server 是链接服务器的系统名称。

<code>table_name</code>	<code>table_name</code> 是表格的系统名称。
<code>table_schema</code>	<code>table_schema</code> 是表格模式的系统名称。
<code>table_catalog</code>	<code>table_catalog</code> 是表格数据库的系统名称。
<code>table_type</code>	<code>table_type</code> 是一个系统名称，是用逗号分开的表格类型列表：ALIAS，GLOBAL，TEMPORARY，LOCAL TEMPORARY，SYNONYM，SYSTEM TABLE，VIEW 和 TABLE。

## `sp_unbindefault`

`sp_unbindefault` 系统存储过程可拆分列上或用户定义的数据类型上的缺省值。

### 语法

```
sp_unbindefault { object_name } [ , futureonly_flag ]
```

### 变量

<code>object_name</code>	<code>object_name</code> 是 <code>nvarchar(776)</code> 型数据，是表格和列，或用户定义的数据类型。
<code>futureonly_flag</code>	<code>futureonly_flag</code> 是 <code>varchar(15)</code> 型数据，它只用于通过阻止用户定义的数据类型中现有的列删除缺省值，来从用户定义的数据类型中拆分缺省值。

## sp\_unbindrule

sp\_unbindrule 系统存储过程拆分列上或用户定义的数据类型上的角色。

### 语法

```
sp_unbindrule{ object_name } [ , futureonly_flag ]
```

### 变量

object_name	object_name 是 nvarchar(776)型数据，是表格和列，或用户定义的数据类型。
futureonly_flag	futureonly_flag 是 varchar(15)型数据，它只用于通过阻止用户定义的数据类型中现有的列删除角色，来从用户定义的数据类型中拆分角色。

## sp\_update\_alert

如果要更新警告，sp\_update\_alert 系统存储过程必须在 msdb 数据库中运行。

### 语法

```
sp_update_alert { name } [ , new_name ] [ , enabled ] [ , message_id ]  
[ , severity ]  
[ , delay_between_responses ] [ , notification_message ]  
[ , include_event_description_in ] [ , database_name ]  
[ , event_description_keyword ] [ , job_id ] [ , job_name ]
```

```
[ ,occurrence_count ][ ,count_reset_date ][ ,count_reset_time ]
[ ,last_occurrence_date ] [ ,last_occurrence_time ]
[ ,last_response_date ]
[ ,last_response_time ][ ,raise_snmp_trap ][ , performance_condition
]
[ , category ]
    变量
```

[ @name = ] name	name 是系统名称，是已有警告的名称。
[ , [ @new_name = ] new=name ]	new=name 是系统名称，是已有警告的新名称。
[ , [ @enabled = ] enabled ]	enabled 是 tinyint 型数据，其中 1=启用(缺省值)，0=不启用。
[ , [ @message_id = ] message_id ]	message_id 是 sysmessages 表中与警告有关的的整型错误号，(可以用 sp_addmessage 添加自己的信息标识符，其中变量 WITH_LOG 将错误信息写入 NT Event 日志。)如果提供了严重级别变量，该变量必须为 NULL 或 0，还必须写入 Microsoft Windows NT 应用程序事件日志中以发出警

[ , [ @delay\_between\_responses= ]  
delay\_between\_response ]

[ , [ @notification\_message = ]  
notification\_message ]

[ , [ @include\_event\_description\_in = ]  
include\_event\_description \_in ]

[ , [ @database\_name = ]

告。[ , [ @severity = ] severity ] severity 是一整数，表示与警告有关的错误的严重等级。此严重等级的任意信号写入 NT Event 日志中，都将发出警告，如果使用 message\_id，该变量必须为 NULL 或 0。

delay\_between\_response 是一个整数，表示发出下一个警告前等待的秒数。该变量可以防止为同一警告重复发送电子邮件或重复寻呼，缺省值为 0。

notification\_message 是 nvarchar(512) 型数据，缺省值为 NULL，它可用以向警告接受者发送附加信息。可以在该参数中包括故障诊断信息。

include\_event\_description\_in 是 tinyint 型数据，缺省值为 0，它决定 NT Event Log 中 SQL Server 错误的说明是否是发送给接受者的部分通知信息，其中 0=None, 1=Email, 2=Pager, 4=NetSend, 7=All。

database\_name 是当前数。缺省值



database\_name ]

[ , [ @event\_description\_keyword= ]  
event\_description\_keyword ]

[ , [ @job\_id = ]job\_id | [ @job\_name  
= ] job\_name ]

[ , [ @occurrence\_count = ]  
occurrence\_count ]

[ , [ @count\_reset\_date = ]  
count\_reset\_date ]

[ , [ @count\_reset\_time= ]  
count\_reset\_time ]

[ , [ @last\_occurrence\_date = ]  
last\_occurrence\_date ]

NULL 表示服务器的任一数据库中的错误都将发出警告。

event\_description\_keyword 是 nvarchar(100) 型数据，缺省值为 NULL，它描述了 Windows NT 事件日志中的 SQLServer 错误，对象名称可以使用 LIKE 表达式通配符。

job\_id 是作业的唯一标识符，缺省值为 NULL。需提供 job\_id 或 job\_name 作为变量。发出警告时，job\_name 是要执行作业名的 CmdExec 或 Transact-SQL 系统名称，缺省值为 NULL。

occurrence\_count 是表示已发出警告的次数的整数值，它可以复位为 0。

count\_reset\_date 是一个整数，表示出现计数复位的日期。

count\_reset\_time 是一个整数，表示出现数复位的时间，它可以复位为 0。

last\_occurrence\_date 是一个整数，是出现警告的终结日期。其只能复位为 0。

[ , [ @last\_occurrence\_time = ]  
last\_occurrence\_time ]

[ , [ @last\_response\_date = ]  
last\_response\_date ]

[ , [ @last\_response\_time = ]  
last\_response\_time ]

[ , [ @raise\_snmp\_trap = ]  
raise\_snmp\_trap ]

[ , [ @performance\_condition = ]  
performance\_condition ]

[ , [ @category\_name = ]  
category\_name ]

last\_occurrence\_time 是一个整数，是出现警告的终结时间。其只能复位为 0。

last\_response\_date 是一个整数，表示 SQL Server Agent 响应警告的终结日期，它只能复位为 0。

last\_response\_time 是一个整数，表示 SQL ServerAgent 响应警告的终结时间，它只能复位为 0。

raise\_snmp\_trap 专供 Microsoft SQL Server 之用。

performance\_condition 专供 Microsoft SQL Server 使用。

category\_name 专供 Microsoft SQL Server 使用。

## sp\_updatecategory

sp\_updatecategory 系统存储过程可修改目录名称。

### 语法

sp\_update\_category { [ @class = ] class [ , [ @name = ] old\_name

```
e      ]
[ , [ @new_name = ]      new_name      ]}
      变量
[ @class = ]      class      class 是目录的 varchar(8)型分类，
                                其中 JOB=更新作业（类型必须为
                                MULTISERVER），ALERT=更新警告，
                                OPERATOR=更新操作员。
[ , [ @name = ]      old_name      ] old_name 是系统名称，表示目录的旧
                                名称。
[ , [ @new_name = ]      new_name      ] new_name 是系统名称，表示目录的新
                                名称。
```

## sp\_update\_job

sp\_update\_job 系统存储过程可修改作业。需提供 job\_id 或 job\_name 变量，但不必同时提供两个变量。

### 语法

```
sp_update_job { [ @job_id ]      job_id      | [ @job_name = ]      job_name      }
[ , [ @new_name = ]      new_name      = ] [ , [ @enabled = ]      enabled      ]
[ , [ @description = ]      description      ] [ , [ @start_step_id = ]      step_id
]
[ , [ @category_name = ]      category      ] [ , [ @owner_login_name = ]
```

```

login      ]
[ , [ @notify_level_eventlog = ]      eventlog_level      ]
[ , [ @notify_level_email = ]        email_level        ]
[ , [ @notify_level_netsend = ]      netsend_level      ]
[ , [ @notify_level_page = ]         page_level         ]
[ , [ @notify_email_operator_name = ]      email_name        ]
[ , [ @notify_netsend_operator_name = ]    netsend_operator   ]
[ , [ @notify_page_operator_name = ]      page_operator     ]
[ , [ @delete_level = ]               delete_level       ] [ , @automatic_post = ]
automatic_post ]

```

## 变量

[ @job_id= ] job_id   [ @job_name = ] job_name	job_id 是作业的唯一标识符 ,job_name 是作业的系统名称。
[ , [ @new_name = ] new_name ]	new_name 是作业的新系统名称。
[ , [ @enabled = ] enabled ]	enabled 决定启用作业还是不启用 , 其中 1=启用 , 0=禁用。缺省值为 1。
[ , [ @description = ] description ]	description 是作业的 nvarchar(512)型 说明。缺省值为 “ No description available ”。
[ , [ @start_step_id = ] step_id ]	step_id 是一个整数 , 表示作业第一步

```
[ , [ @category_name = ]      category
]
```

```
[ , [ @owner_login_name = ]    login
]
```

```
[ , [ @nofity_level_eventlog = ]
eventlog_level ]
```

```
[ , [ @nofity_level_email = ]
email_level ]
```

```
[ , [ @notify_level_netsend = ]
netsend_level ]
```

```
[ , [ @notify_level_netsend = ]
netsend_level ]
```

```
[ , [ @notify_level_eventlog = ]
```

的序号，其缺省值为 1。

category 是作业目录的系统名称。缺省值为 NULL。

login 是注册的系统名称，该注册的许可用于执行作业。缺省值为 NULL。

eventlog\_level 是一个整数，它决定何时出现作业的 NT 事件记录项，其中 0=Never, 1=On Success, 2=On Failure, 3=Always，缺省值为 2。

email\_level 是一个整数，它确定发送电子邮件的时间，其中 0=Never, 1=On Success, 2=On Failure, 3=Always，缺省值为 0。

netsend\_level 是一个整数，它确定发送网络信息的时间，其中 0=Never, 1=On Success, 2=On Failure, 3=Always，缺省值为 0。

page\_level 是一个整数，它确定将寻呼信号发送至寻呼机上的时间，其中 0=Never, 1=On Success, 2=On Failure, 3=Always，缺省值为 0。

E-mail\_name 是在 E-mail\_level 上

<pre>eventlog_level ]</pre>	<p>发送电子邮件的人的 nvarchar(100) 型名称。缺省值为 NULL。</p>
<pre>[ , [ @notify_netsend_operator_name = ] netsend_operator ]</pre>	<p>netsend_operator 是在 netsend_level 上发送网络信息的操作员的 nvarchar(100) 名称。缺省值为 NULL。</p>
<pre>[ , [ @notify_page_operator_name = ] page_operator ]</pre>	<p>page_operator 是在 page_level 上要寻呼的人的 nvarchar(100) 型名称。缺省值为 NULL。</p>
<pre>[ , [ @delete_level = ] delete_level ]</pre>	<p>delete_level 是一个整数，它确定删除作业的时间，其中 0=Never, 1=On Success, 2=On Failure, 3=Always，缺省值为 0。</p>
<pre>[ , [ @automatic_post = ] automatic_post ]</pre>	<p>automatic_post 专供 Microsoft SQL Server 使用。</p>

## sp\_update\_jobschedule

sp\_update\_jobschedule 系统存储过程可修改作业进度表，并更新其版本号。

### 语法

```
sp_update_jobschedule{ [ @job_id = ] job_id | [ @job_name = ] job_name
```

```

    }
[ , [ @name = ]      name      ][ , [      newname      = ]      new_name      ]
[ , [ @enabled = ]   enabled    ][ , [ @freq_type = ]      freq_type    ]
[ , [ @freq_interval = ]   freq_interval    ]
[ , [ @freq_subday_type = ]   freq_subday_type    ]
[ , [ @freq_subday_interval = ]   freq_subday_interval    ]
[ , [ @freq_relative_interval = ]   freq_relative_interval    ]
[ , [ @freq_recurrence_factor = ]   freq_recurrence_factor    ]
[ , [ @active_start_date = ]   active_start_date    ]
[ , [ @active_end_date = ]     active_end_date    ]
[ , [ @active_start_time = ]   active_start_time    ]
[ , [ @active_end_time = ]     active_end_time    ]

```

### 变量

<pre>[ @job_id ] job_id   [ @job_name = ] job_name    ]</pre>	<p>job_id 是作业的唯一专用标识符，缺省值为 NULL。需提供 job_id 或 job_name。job_name 是要放在调度表上的作业的系统名称，缺省值为 NULL。</p>
<pre>[ @name = ]      name      ]</pre>	<p>name 是作业调度条目的系统名称。</p>
<pre>[ , [ @newname = ]      new_name      ]</pre>	<p>new_name 是作业调度条目的系统新名称。</p>
<pre>[ , [ @enabled = ] enabled ]</pre>	<p>enabled 确定是启用还是禁用作业按</p>

[ , [ @freq\_type = ] freq\_type ]

[ , [ @freq\_interval = ]  
freq\_interval ]

[ , [ @freq\_subday\_type = ]  
freq\_subday\_type ]

[ , [ @freq\_subday\_interval = ]  
freq\_subday\_interval ]

[ , [ @freq\_relative\_interval = ]  
freq\_relative\_interval ]

照调度表运行，其中 1=启用，0=禁用，缺省值为 1。

freq\_type 是一个整数，它确定执行作业的时间，缺省值为 0，其中 1=1 次，4=每日，8=每星期，16=每月，32=每月（依频率间隔而定），64=SQL Server Agent 启动时执行。

freq\_interval 是一个整数，它确定作业的执行日期，其中 1=周日，2=周一，3=周二，4=周三，5=周四，6=周五，7=周六，8=周日，9=工作日，10=周末。

freq\_subday\_type 是一个整数，表示 freq\_subday\_interval 的单位，缺省值为 0，1=指定的时间，2=秒，3=分，4=小时。

freq\_subday\_interval 是一个整数，缺省值为 0，它确定每个作业执行期间以 freq\_subday\_type 为单位的数目。

freq\_relative\_interval 是一个整数，它确定每月的 freq\_interval(缺



```
[ , [ @freq_recurrence_factor = ]  
freq_recurrence_factor ]
```

```
[ , [ @active_start_date = ]  
active_start_date ]
```

```
[ , [ @active_end_date = ]  
active_end_date ]
```

```
[ , [ @active_start_time = ]  
active_start_time ]
```

```
[ , [ @active_end_time = ]  
active_end_time ]
```

省值为 0)，其中 1=第一个，2=第 2 个，4=第 3 个，8=第 4 个，16=最后一个。

freq\_recurrence\_factor 是一个整数，它表示作业执行期间的月数，缺省值为 0。

active\_start\_date 是一个整数，表示开始执行作业的日期（缺省值为 NULL），其格式为 YYYYMMDD。

active\_end\_date 是一个整数，表示停止执行作业的日期，其缺省值为 99991231(9999 年 12 月 31 日)。

active\_start\_time 是一个整数，表示开始执行作业的时间，缺省值为 000000(12 : 00 : 00AM)，格式为 HHMMSS。

active\_end\_time 是一个整数，表示结束执行作业的时间，缺省值为 235959(11:59:59 P.M.)。

## sp\_update\_jobstep

sp\_update\_jobstep 系统存储过程修改作业的步骤，并更新其版本号。

SP 语法表法如下：

```
sp_update_jobstep{[ @job_id = ]      job_id      | [ @job_name = ]      job_name
}
{ , [ @step_id = ]      step_id      }[ , [ @step_name = ]      step_name      ]
[ , [ @subsystem = ]      subsystem      ][ , [ @command = ]      command      ]
[ , [ @additional_parameters = ]      parameters      ]
[ , [ @cmdexec_success_code = ]      success_code      ]
[ , [ @on_success_action = ]      success_action      ]
[ , [ @on_success_step_id = ]      success_step_id      ]
[ , [ @on_fail_action = ]      fail_action      ][ , [ @on_fail_step_id = ]
fail_step_id      ]
[ , [ @server = ]      server      ][ , [ @database_name = ]      database
]
[ , [ @database_user_name = ]      user      ][ , [ @retry_attempts = ]
retry_attempts      ]
[ , [ @retry_interval = ]      retry_interval      ][ , [ @os_run_priority = ]
run_priority      ]
[ , [ @output_file_name = ]      file_name      ][ , [ @flags = ]      flags      ]
```

## 变量

[ @job\_id = ] job\_id | [ @job\_name  
= ] job\_name

{ , [ @step\_id = ] step\_id } [ ,  
[ @step\_name = ] step\_name ]

[ , [ @subsystem = ] subsystem ]

[ , [ @command = ] command ]

job\_id 是作业的唯一标识符，缺省值为 NULL，job\_name 是作业的系统名称，缺省值为 NULL，需提供 job\_id 或 job\_name。

step\_id 是表示作业步骤的整数，缺省值 NULL，step\_name 是步骤的系统名称。

subsystem 子是 SQL Server Enterprise Manager 用来执行命令的 nvarchar(40) 型子系统，缺省值为 TSQL。

command 是发送至 SQL Server Agent 上执行的 nvarchar(3200) 型指令，缺省值为 NULL。指令可以标记，其中 [ A-DBN ] = 数据库名称，[ A-SVR ] = 服务器名称，[ A-ERR ] = 错误号，[ A-SEV ] = 错误严重性，[ A-MSG ] = 信息文本，[ DATE ] = 格式为 YYYYMMDD 的当前日期，[ JOBID ] = 作业 ID，[ MACH ] = 计算机名称，[ MSSA ] = Master SQL Server Agent 名称，

```
[ , [ @additional_parameters = ]  
  parameters ]  
[ , [ @cmdexec_success_code = ]  
success_code ]  
[ , [ @on_success_action = ]  
success_action ]  
  
[ , [ @on_success_step_id = ]  
success_step_id ]  
  
[ , [ @on_fail_action = ]  
fail_action ]
```

[ SQLDIR ] =SQL Server 的安装目录 (C: \ MSSQL7), [ STEPCT ] =此步骤执行的次数, [ STEPID ] =步骤 ID, [ TIME ] =格式为 HHMMSS 的当前时间, [ STRTTM ] =作业启动时间, 格式为 HHMMSS, [ STRDRT ] =作业启动日期, 格式为 YYYYMMDD。

parameters 专供 Microsoft SQL Server 之用。

success\_code 是成功完成的整型返回值, 缺省值为 0。

success\_action 是 tinyint 型数据, 它确定成功代码返回时的操作, 其中 1=成功时退出 (缺省值), 2=失败时退出, 3=转到下一步, 4=转到 Step success\_step\_id。

success\_step\_id 是一个整数, 表示 success\_code 和 success\_action 为 4 时, 此作业执行的步骤。缺省值为 0。

fail\_action 是 tinyint 型数据, 表示非代码返回值, 其中 1=成功时退出

```
[ , [ @on_fail_step_id = ]  
fail_step_id ]
```

```
[ , [ @server = ] server ]
```

```
[ , [ @database_name = ] database  
 ]
```

```
[ , [ @database_user_name = ]  
user ]
```

```
[ , [ @retry_attempts = ]  
retry_attempts ]
```

```
[ , [ @retry_interval = ]  
retry_interval ]
```

```
[ , [ @os_run_priority = ]  
run_priority ]
```

```
[ , [ @output_file_name = ]  
file_name ]
```

(缺省值), 2=失败时退出, 3=转到下一步, 4=转到 Step fail\_step\_id。

fail\_step\_id 是一个整数, 表示 success\_action 为 4 时, 此作业执行的步骤。缺省值为 0。

server 是执行步骤的服务器的 nvarchar(30) 型名称。缺省值为 NULL。

database 是执行步骤的数据库的系统名称, 缺省值为 NULL。

user 是用来执行此步骤的用户帐户的系统名称, 缺省值为 NULL。

retry\_attempts 是确定步骤失败时重试次数的整数, 缺省值为 NULL。

retry\_interval 是一个整数, 表示重试期间的秒数, 其缺省值为 0。

run\_priority 是一个整数, 表示用以执行步骤的 OS 优先权, 其中 -15=空闲, -1=在正常之下, 0=正常(缺省值), 1=在正常之上, 15=时间紧迫。

file\_name 是存储此步骤输出的输出文件的 nvarchar(200) 型名称, 它可

[ , [ @flags = ] flags ]

以包括命令标志。

flags 专供 Microsoft SQL Server 使用。

## sp\_update\_notification

sp\_update\_notification 系统存储过程可修改警告通知的方式，该程序必须在 msdb 系统数据库中运行。

### 语法

```
sp_update_notification [ @alert_name = ] alert ,  
[ @operator_name = ] operator ,  
[ @notification_method = ] notification
```

### 变量

[ @alert\_name = ] alert  
[ @operator\_name = ] operator

所选择的警告的系统名称。

operator 是系统名称，表示警告通知的操作员。

[ @notification\_method = ]  
notification

notification\_method 是 tinyint 型数据，表示出现警告时如何通知操作员，其中 1=电子邮件（缺省值，必须配置 SQL Mail），2=寻呼，4=NetSend,7=All。

## sp\_update\_operator

sp\_update\_operator 系统存储过程在 msdb 系统数据库中运行,以修改操作员的信息。

### 语法

```
sp_update_operator [ @name = ]          name
[ , [ @new_name = ]      new_name      ]
[ , [ @enable = ]      enabled      ]
[ , [ @email_address = ]      email_address      ]
[ , [ @pager_address = ]      pager_number      ]
[ , [ @weekday_pager_start_time = ]      weekday_pager_start_time      ]
[ , [ @weekday_pager_end_time = ]      weekday_pager_end_time      ]
[ , [ @saturday_pager_start_time = ]      saturday_pager_start_time      ]
[ , [ @saturday_pager_end_time = ]      saturday_pager_end_time      ]
[ , [ @sunday_pager_start_time = ]      sunday_pager_start_time      ]
[ , [ @sunday_pager_end_time = ]      sunday_pager_end_time      ]
[ , [ @pager_days = ]      pager_days      ]
[ , [ @netsend_address = ]      netsend_address      ]
[ , [ @category_name = ]      category      ]
```

### 变量

```
[ @name = ] name  
[ , [ @new_name = ] new_name  
 ]  
[ , [ @enable = ] enabled ]
```

```
[ , [ @email_address = ]  
email_address ]  
 [ , [ @pager_address = ]  
pager_number ]  
 [ , [ @weekday_pager_start_time = ]  
weekday_pager_start_time ]
```

```
[ , [ @weekday_pager_end_time = ]  
weekday_pager_end_time ]
```

```
[ , [ @saturday_pager_start_time  
= ] saturday_pager_start_time ]
```

name 是现有操作员的系统名称。

new\_name 是现有操作员的系统新名称。

enabled 是表示操作员状态的 tinyint 型数据，其中 1=启动(缺省值)，0=不启用。

email\_address 是操作员的 varchar(100) 型电子邮件地址。

pager\_address 是 varchar(100)型寻呼地址，缺省值为 NULL。

weekday\_pager\_start\_time 是 HHMMSS 格式的整数，缺省值为 24 小时制的 090000，即 9:00A.M，表示开始给操作员发送寻呼通知的时间(周一至周五)。

weekday\_pager\_end\_time 是 HHMMSS 格式的整数，缺省值为 24 小时制的 180000，即 6:00P.M，表示停止给操作员发送寻呼通知的时间(周一至周五)。

saturday\_pager\_start\_time 是 HHMMSS 格式的整数，缺省值为 24 小



[ , [ @saturday\_pager\_end\_time = ]  
saturday\_pager\_end\_time ]

[ , [ @sunday\_pager\_start\_time = ]  
sunday\_pager\_start\_time ]

[ , [ @sunday\_pager\_end\_time = ]  
sunday\_pager\_end\_time ]

[ , [ @pager\_days = ] pager\_days ]

[ , [ @netsend\_address = ]

时制的 090000，即 9:00A.M，表示星期六开始给操作员发送寻呼通知的时间。

saturday\_pager\_end\_time 是 HHMMSS 格式的整数，缺省值为 24 小时制的 180000，即 6:00P.M，表示星期六停止给操作员发送寻呼通知的时间。

sunday\_pager\_start\_time HHMMSS 格式的整数，缺省值为 24 小时制的 090000，即 9:00A.M，表示星期日开始给操作员发送寻呼通知的时间。

sunday\_pager\_end\_time 是 HHMMSS 格式的整数，缺省值为 24 小时制的 180000，即 6:00P.M，表示星期日停止给操作员发送寻呼通知的时间。

pager\_days 是 tinyint 型数据，其值为 1~127，表示可以寻呼操作员的日期，缺省值为 0=Nerver，将每一天的值加在一起。其中 1=周日，2=周一，4=周二，8=周三，16=周四，32=周五，64=周六，从周一到周五是 62。

netsend\_address 是通过网络信息弹

<pre> netsend_address ] </pre>	<p>出通知操作员的 varchar(100)型电子邮件地址。缺省值为 NULL。</p>
<pre> [ , [ @category_name = ] category ] </pre>	<p>category 是目录名称的系统名称。缺省值为 NULL。</p>

## sp\_updatestats

sp\_updatestats 系统存储过程可对当前数据库中用户建立的所有表格执行 UPDATE STATISTICS。

### 语法

```
sp_updatestats
```

```
sp_update_targetservergroup
```

sp\_update\_targetservergroup 系统存储过程可修改目标服务器组的名称。

```
sp_update_targetservergroup [ @name = ] current_name
```

```
[ , [ @new_name = ] new_name ]
```

### 变量

```
[ @name = ] current_name
```

current\_name 是现有目标服务器组的系统名称。

```
[ , [ @new_name = ] new_name ]
```

new\_name 是现有目标服务器组的系统新名称。

## sp\_updatetask

sp\_updatetask 系统存储过程只向后兼容，它用于修改任务。Microsoft SQL Server 7 程序员可用作作业代替任务，用 SQL Server Agent 代替 SQL Executive。

## sp\_validname

sp\_validname 系统存储程序用于检查标识符名称是否是有效的 SQL Server 标识符。

### 语法

```
sp_validname [ @name = ] name  
[ , [ @raise_error = ] raise_error ]
```

### 变量

```
[ @name = ] name
```

name 是要确认的字符串的系统名称。

```
[ , [ @raise_error = ] raise_error ]
```

raise\_error 是一个位，其 1=显示错误信息，0=不显示错误信息。

## sp\_validatelogins

sp\_validatelogins 系统存储过程显示不再存在但在 SQL Server 系统表中有行的 Windows NT 用户和组。

### 语法

```
sp_validatelogins
```

## sp\_who

sp\_who 系统存储过程可显示当前在 SQL Server 上运行的用户和过程。

### 语法

```
sp_who [ login ]
```

变量 login 是 SQL Server 上用户的系统名称注册名称。如果没有提供该注册，将显示所有的用户与过程。

## xp\_cmdshell

xp\_cmdshell 扩展存储过程就象在操作系统提示行上一样执行命令。注意当使用 Windows 95 / 98 时，返回代码总是零(0)。

### 语法

xp\_cmdshell { command\_string } [ ,no\_output ]

变量

command\_string            command\_string 是要执行的 varchar(255) 或 nvarchar(4000)型命令，这些命令就象在操作系统命令提示符上执行的命令一样。

no\_output                 no\_output 是不给用户返回输出的可选参数。

## xp\_deletemail

xp\_deletemail 系统存储过程可从 SQL Server 收件箱中删除信息。

语法

xp\_deletemail { message\_number }

变量 message\_number 是 varchar(255)型数据，表示要删除的邮件信息的数目。

## xp\_enumgroups

xp\_enumgroups 系统存储过程可显示 Windows NT 域中的 Windows NT 组。

语法

xp\_enumgroups [ domain\_name ]

变量 domain\_name 是 Windows NT 域的系统名称，如果没有提供域或 SQL

Server 计算机和域同名，将返回本地组，否则将返回全局组。

## xp\_findnextmsg

xp\_findnextmsg 系统存储过程和 sp\_processmail 一起使用，通过接受 message\_id 和返回下一条 message\_id 来查找下一条信息。

### 语法

```
xp_findnextmsg [ [ @msg_id = ] msg_id [ OUTPUT ]  
[ , [ @type = ] type ]  
[ , [ @unread_only = ] unread_value ]
```

### 变量

[ @msg\_id = ] msg\_id

msg\_id 既用作输入变量，又用作输出变量，它接受输入的信息串，显示输出的下一条信息。

### OUTPUT

[ @type = ] type

OUTPUT 将 msg\_id 作为输出。

type 是输入的信息类型，以 IPM 开头，如果要寻找以 IPC 开头的类型，必须设置此变量，其缺省值为 NULL。

[ @unread\_only = ] unread\_value

unread\_value 可以是 true 或 false，其中 true=只是未读取信息，false=所有信息。

## xp\_grantlogin

xp\_grantlogin 系统存储过程为 Windows NT 组或用户提供 SQL Server 注册，该程序只向后兼容。Microsoft SQL Server 程序员可用 sp\_grantlogin 代替该程序。

### 语法

```
xp_grantlogin { [ @loginame = ] login } [ , [ @logintype = ] logintype ]
```

### 变量

[ @loginame = ] login	login 是格式为 Domain \ User 的 Windows NT 用户或组的系统名称。
[ @logintype = ] logintype	logintype 是 varchar(5) 型安全等级，其中 admin=将注册添加至系统管理员固定的服务器角色上。

## xp\_logevent

xp\_logevent 系统存储过程将用户建立的信息写入 SQL Server 错误记录和 / 或 Windows NT Event Viewer 中，该程序用于不通知用户就发出警告。

### 语法

```
xp_logevent { error_number, message } [ , severity ]
```

### 变量

error_num	ererror_number 是用户建立的大于 50,000 的错误号。
message	message 是信息表示记录到 SQL Server 错误日志文件或 NT Event Viewer 中的字符串，最多可有 8000 个字符。
severity	severity 是字符串 INFORMATIONAL, WARNING 或 ERROR, 缺省值为 INFORMATIONAL。

## xp\_loginconfig

xp\_loginconfig 系统存储过程显示与 Microsoft SQL Server 有关的严重等级配置。

### 语法

```
xp_loginconfig [ config_name ]
```

变量 config\_name 是要显示的配置值，不提供该变量，将查看所有配置。

## xp\_logininfo

xp\_logininfo 系统存储过程显示与 Microsoft SQL Server 安全帐户有关的信息。

### 语法

```
xp_logininfo [ account_name [ , all | members ]
```



[ , [ @privilege = ] variable\_name OUTPUT ]]

变量

account\_name account\_name 是带有 SQL Server 安全措施的 Windows NT 用户或组的名称。如果没有提供 account\_name，将显示所有帐户。

all | members all | members 是一个字符串，其中 all=帐户的全部许可，members=Windows NT 组的成员。缺省值为 members。

[ @privilege = ] privilege 是 nchar(21)型输出变量，返回 Windows NT 帐户用户的特权级：admin 或 repl。

variable\_name OUTPUT 用 variable\_name 的值作为输出变量。

## xp\_msver

xp\_msver 系统存储过程提供 SQL Server 版本信息。

语法

xp\_msver [ optname ]

变量 optname 是选项名，不提供该变量可以查看所有的选项名称。

## xp\_readmail

xp\_readmail 系统存储过程可从电子邮件收件箱中读取邮件信息。

### 语法

```
xp_readmail [ [ @msg_id = ]          message_number          ] [ , [ @type = ]          type
            ] [ OUTPUT ]
[ , [ @peek = ]          peek          ]
[ , [ @suppress_attach = ]      suppress_attach          ]
[ , [ @originator = ]          sender          OUTPUT ]
[ , [ @subject = ]          subject_line          OUTPUT ]
[ , [ @message = ]          body_of_message          OUTPUT ]
[ , [ @recipients = ]          recipient_list          OUTPUT ]
[ , [ @cc_list = ]          cc_list          OUTPUT ]
[ , [ @bcc_list = ]          bcc_list          OUTPUT ]
[ , [ @date_received = ]          date          OUTPUT ]
[ , [ @unread = ]          value          OUTPUT ]
[ , [ @attachments = ]          temp_file_paths          OUTPUT ]
[ , [ @skip_bytes = ]          bytes_to_skip          OUTPUT ]
[ , [ @msg_length = ]          length_in_bytes          OUTPUT ]
[ , [ @originator_address = ]          originator_address          OUTPUT ]
```

## 变量

[ @msg\_id = ] message\_number

type

## OUTPUT

[ @peek = ] peek

[ @suppress\_attach = ]  
suppress\_attach

[ @originator = ] sender

[ @subject = ] subject\_line

要读取的信息号

由 MAPI 邮件定义的 varchar(255)信息类型。如果类型由 IPM 开头，信息将显示在邮件客户机的收件箱中。以 IPC 开头的信息类型不显示在客户机的收件箱中。如果信息类型以 IPC 开头，且设置了类型变量，邮件只显示在收件箱中。

OUTPUT 把类型变量值用作输出变量。

peek 是 varchar(5)型数据，其中 true=不必标记为读取就可以读取，false=邮件标记为读取(缺省值)。

suppress\_attach 表示：true=不建立临时文件，false=继续进行并为附件建立临时文件。

sender 是 varchar(255)型数据，是发送器的电子邮件名称。

subject\_line 是 varchar(255)型数据，

[ @message = ] body\_of\_message

[ @recipients = ] recipient\_list

[ @cc\_list = ] cc\_list

[ @bcc\_list = ] bcc\_list

[ @date\_received= ] date

[ @unread = ] value

[ @attachments = ]  
temp\_file\_paths

[ @skip\_bytes = ] bytes\_to\_skip  
OUTPUT

[ @msg\_length = ] length\_in\_bytes  
OUTPUT

是电子邮件内容的摘要。

message 是 varchar(255)型数据，是信息的正文。

recipients 是接受邮件信息的个人。用分号将名称分隔开。

发送邮件复本的个人，用分号将名称分隔开。

另一个收件人不知道用户正用邮件信息拷贝盲目复本收件人。

date 是接收邮件信息的 varchar(255)型日期。

unread 是 varchar(5)型数据，其中 true=没有读取信息，false=已读取信息。

temp\_file\_paths 是 varchar(255)型数据，是邮件附件的临时路径。可用分号将路径分开。

bytes\_to\_skip 是将下一个字节数(由 length\_in\_bytes 变量确定)读入输出变量前要跳过的字节数。缺省值为 0。

length\_in\_bytes 是信息中的字节数，与 bytes\_to\_skip 一起使用时一次可读取信息 255 字节。缺省值为 255 字节。

[ @originator\_address = ]  
originator\_address

originator\_address 是 varchar(255)型数据，是电子邮件信息的发送器。

## xp\_revokelogin

xp\_revokelogin 系统存储过程可从 Windows NT 组或用户中撤消 SQL Server 注册，它只用于向后兼容。Microsoft SQL Server 7 程序员可使用 sp\_revokelogin (存储过程代替扩展存储程序)。

### 语法

xp\_revokelogin { [ loginame = ] login }

变量 login 是格式为域 \ 用户中 Windows NT 用户或组的系统名称 (例如 CONTINUUM \ WCCOMP)。

## xp\_sendmail

xp\_sendmail 系统存储过程可发送电子邮件信息、查询结果集或附件，但必须运行 SQL Mail。

### 语法

xp\_sendmail { [ @recipients = ] recipients [ , ... n ] }  
[ , [ @message = ] message ]  
[ , [ @query = ] query ]

```

[ , [ @attachments = ] attachments ]
[ , [ @copy_recipients = ] copy_recipients ][ ,... n ]
[ , [ @blind_copy_recipients = ] blind_copy_recipients ][ ,...
n ]
[ , [ @subject = ] subject ]
[ , [ @type = ] type ]
[ , [ @attach_results = ] attach_value ]
[ , [ @no_output = ] output_value ]
[ , [ @no_header = ] header_value ]
[ , [ @width = ] width ]
[ , [ @separator = ] separator ]
[ , [ @echo_error = ] echo_value ]
[ , [ @set_user = ] user ]
[ , [ @dbuse = ] database ]
    变量

```

```
[ @recipients = ] recipients
```

recipients 是接收邮件信息的人，用分号将名称分隔开。

N

n 表示前面的项可以重复。

```
[ @message = ] message
```

给接收人的的信息文本。

```
[ @query = ] query
```

在邮件中发送带有结果的 SQL 查询。

```
[ @attachments = ] attachments
```

加至邮件信息上的文件。

```
[ @copy_recipients = ]
```

向其发送邮件的人，用分号将名称分

```
copy_recipients
[ @blind_copy_recipients = ]
blind_copy_recipient
[ @subject = ] subject

[ @type = ] type

[ @attach_results = ]
attach_values

[ @no_output = ] output_value

[ @no_header = ] header_value
```

隔开。

另一个收件人不知道用户正在用邮件信息复制盲目复制收件人。

如果没有指定邮件信息的主题，SQL Server Message 将显示为主题。

信息类型由 MAPI 邮件确定，如果类型以 IPM 开头，且类型为 NULL，信息将显示在邮件客户机的收件箱中。以 IPC 开头的信息类型不显示在收件箱中。如果信息类型以 IPC 开头，且设置了类型变量，邮件将只显示在收件箱中。

TRUE=将结果发送到附加文件中，文件名是附件中的第一个文件名，如果附件为 NULL，文件名的扩展名就是 .txt。FALSE=在信息上附加结果集。

TRUE=输出不发送到发送邮件的用户话路中，FALSE=缺省值，发送邮件的用户话路接受输出。

TRUE=列标题不和查询结果一起发送。FALSE=缺省值、列标题和查询

[ @width = ] width

[ @separator = ] separator

[ @echo\_error = ] echo\_value

[ @set\_user = ] user

[ @dbuse = ] database

结果一起发送。

查询的结果线宽，缺省值为 80 个字符。

查询结果集的列分隔符。

TRUE=在邮件信息上附加查询生成的错误、信息和行计数。FALSE=不在邮件信息上附加查询生成的错误、信息和行计数。

user 是带有执行查询许可的用户，缺省值为客户。

执行查询的数据库。缺省值为 NULL，数据库成为用户的缺省数据库。

## xp\_snmp\_getstate

xp\_snmp\_getstate 系统存储过程说明 SQL Server Simple Network Management Protocol(SNMP)是否可由 SNMP 客户机（也许来自另一个操作系统，例如主机或 UNIX)用来查询 SQL Server。如果存储过程取决于 SNMP 代理，则可以用此程序查询代理的状态。

### 语法

```
xp_snmp_getstate [ return_status OUTPUT ]
```

### 变量



return\_status return\_status 是 SNMP 代理的整型状态。不使用该变量可查看结果状态。

OUTPUT OUTPUT 使 return\_status 成为输出变量。

## xp\_snmp\_raissetrap

xp\_snmp\_raissetrap 系统存储过程用来与 SNMP 客户机通讯，并允许客户机检查 SQL Server，在 Windows Network 控制面板上，可以控制哪些 SNMP 客户机接受警告。

### 语法

```
xp_snmp_raissetrap { server , database , error_message  
 ,message_number ,  
severity, user , comment ,date_and_time,return_status OUTPUT }
```

### 变量

server server 是 varchar(30)型数据，是发出警告的服务器名。

database database 是 varchar(30) 型数据，是发出警告的数据库名。

error\_message error\_message 是 varchar(255)型数据，是警告的错误信息。

message\_number message\_number 是警告的 sysmessages 系统表信息

错误号。

Severity	severity 是一个整数，表示 message_number 的严重性 ( 1 ~ 25 )。
user	user 是发出警告的用户的 varchar(30)型名称。
comment	comment 是 varchar(30)型数据，表示收到警告以后怎么办的注释。
data_and_time	data_and_time 是日期时间，表示警告出现的日期和时间。
return_status	return_status 是表示语句状态的整数，其中 0=发送出警告，2=内存错误，3=网络错误，4=SQL SNMP 代理超时等待 SNMP 客户，5=SQL SNMP 代理停止。
OUTPUT	OUTPUT 将 return_status 放在输出变量中。

## xp\_sprintf

xp\_sprintf 系统存储过程用于格式化并在格式化后存储输出变量中的字符。

语法

```
xp_sprintf { string OUTPUT, format }  
[ , argument [ , ... n ] ]  
变量
```

string 是 varchar 型数据，是存储输出的变量。

OUTPUT  
Format  
Argument  
N

OUTPUT 将字符串放进输出变量中。  
format 是格式字符串，% 表示变量值。  
argument 是与格式变量对应的串值。  
n 表示前面的项可以重复 50 次。

### 实例

下面是 xp\_sprintf 的一个实例：

```
DECLARE @mystring varchar (255)
```

```
EXEC xp_sprintf @mystring OUTPUT, SELECT * FROM %s = %s ,  
MyTable , MyId , 1
```

在本例中，返回 “SELECT \* FROM MyTable WHERE MyId =1”。

## xp\_sqlinventory

xp\_sqlinventory 系统存储过程可存储所选择的数据库和表格中的 SQL Server 配置信息。

### 语法

```
xp_sqlinventory { database , table , { interval |  
STOP } }
```

### 变量

database database 是 varchar(30)型数据，表示存储配置信息的数据库名称。

table	table 是 varchar(30) 型数据，表示存储配置信息的表格名称。此表格自动建立。
Interval	interval 是一个整数，表示再次检查 SQL Server 库存信息前等待的秒数(有效值为 1~3600(1 小时))。
STOP	STOP 终止正进行的盘点操作。

## xp\_sqlmaint

xp\_sqlmaint 系统存储过程用包含 sqlmaint 开关的信息串调用 sqlmaint 程序，以便对数据库执行一系列维护操作。

### 语法

```
xp_sqlmaint    switch_string
```

变量 switch\_string 是包含 sqlmaint 实用程序开关的信息串，关于 SQLMAINT 实用程序的更详尽资料，可以参阅第八章“实用程序的变化”。

## xp\_sscanf

xp\_sscanf 系统存储过程可从信息串中提取数据放入格式变量提供的变量空位符中。

### 语法

```
xp_sscanf {    string    OUTPUT,    format    }
```

[ , argument [ , ... n ] ]  
变量

string string 是 varchar 型数据，是存储输出的变量。  
OUTPUT OUTPUT 将字符串放进输出变量中。  
Format format 是格式字符串，% 表示变量值。  
Argument argument 是与格式变量对应的串值。  
N n 表示前面的项可以重复 50 次。

### 实例

下面是 xp\_sscanf 的一个实例：

```
DECLARE @table_name varchar (30),@column_name varchar (30)
EXEC xp_sscanf SELECT * FROM MyTable WHERE MyId = 1 ,
SELECT *
FORM %s WHERE %s = 1 / ,@table_name OUTPUT,@column_name
OUTPUT
```

在本例中，返回 MyTable 和 MyId。

### xp\_startmail

xp\_startmail 系统存储过程可启动 Microsoft SQL Server 邮件用户对话。

#### 语法

```
xp_startmail [ [ @user = ] user ] [ , [ @password = ] password ]
```

]

## 变量

[ @user = ] user	如果要在与定义的 SQL Mail 启动邮件信箱不同的邮件信箱中启动邮件，就可以提供用户的系统名称。
[ @password = ] password	password 是用户的邮件信箱口令的系统名称。如果在本地运行邮件客户机，口令可以为 NULL。

## xp\_stopmail

xp\_stopmail 系统存储过程可终止 SQL Mail。

### 语法

xp\_stopmail

## 附录 A 安装前要准备的检验表

### 检验表条目

检验所需的硬件

检验所需的软件

安装类型

Typical

Compact

Custom

域名

SQL Server Name(取自 Windows NT 计算机名称)

启动 SQL Server 服务程序的 Windows NT 用户帐户 (SQL Server 和 SQL Server Agent 可以使用相同或不同的帐号)

启动 SQL Server Agent 服务程序的 Windows NT 用户帐户

用 Administrator 优先权执行 SQL Server 安装程序的 Windows NT 用户帐户字符集排序顺序统一码对照身份验证的模式 (NT Authentication 或 Mixed)

负主要责任的 SQL Server 用户的 User Name 用户的公司名 (可任选)

产品 ID(可任选但需技术支持)

网络协议

安装联机帮助

实用程序文件的位置

数据文件的位置

程序文件的位置

自动启动的 SQL Server

自动启动的 SQL Server 代理

SQL Server 的程序组



