

Crear una base de datos sin especificar SIZE

Este ejemplo crea una base de datos llamada **products2**. El archivo **prods2_dat** se convierte en el archivo principal, con un tamaño igual al tamaño del archivo principal de la base de datos **model**. El archivo de registro de transacciones se crea automáticamente y es un 25 por ciento del tamaño del archivo principal, o 512 KB, el que sea mayor. Como no se ha especificado MAXSIZE, los archivos pueden crecer hasta llenar todo el espacio disponible en el disco.

```
USE master
GO
CREATE DATABASE Products2
ON
( NAME = prods2_dat,
  FILENAME = 'c:\program files\microsoft sql server\mssql\data\prods2.mdf' )
GO
```

CREATE DATABASE

Crea una nueva base de datos y los archivos que se utilizan para almacenar la base de datos o adjunta una base de datos desde los archivos de una base de datos creada anteriormente.

Nota Para obtener más información acerca de la compatibilidad con el antiguo comando DISK INIT, consulte [Dispositivos \(Nivel 3\)](#) en Detalles de compatibilidad con versiones anteriores de Microsoft® SQL Server™.

Sintaxis

```
CREATE DATABASE database_name
[ ON
  [ < filespec > [ ,...n ] ]
  [ , < filegroup > [ ,...n ] ]
]
[ LOG ON { < filespec > [ ,...n ] } ]
[ COLLATE collation_name ]
[ FOR LOAD | FOR ATTACH ]
```

< filespec > ::=

```
[ PRIMARY ]
( [ NAME = logical_file_name , ]
  FILENAME = 'os_file_name'
  [ , SIZE = size ]
  [ , MAXSIZE = { max_size | UNLIMITED } ]
  [ , FILEGROWTH = growth_increment ] ) [ ,...n ]
```

< filegroup > ::=

FILEGROUP *filegroup_name* < filespec > [,...*n*]

Argumentos

database_name

Es el nombre de la nueva base de datos. Los nombres de base de datos deben ser únicos en un servidor y deben seguir las reglas de los identificadores. *database_name* puede tener un máximo de 128 caracteres, a menos que no se especifique ningún nombre lógico para el registro. Si no se especifica ningún nombre lógico de archivo de registro, Microsoft® SQL Server™ genera un nombre lógico al anexar un sufijo a *database_name*. Esto limita *database_name* a 123 caracteres, por lo que el nombre lógico generado del archivo de registro es menor de 128 caracteres.

ON

Especifica que los archivos de disco utilizados para almacenar la parte de datos de la base de datos (archivos de datos) se han definido explícitamente. La palabra clave va seguida de una lista separada por comas de elementos <filespec> que definen los archivos de datos del grupo de archivos principal. A continuación de la lista de archivos del grupo de archivos principal se puede colocar una lista opcional, separada por comas, de elementos <filegroup> que definen los grupos de archivos de usuario y sus archivos.

n

Es un marcador de posición que indica que se pueden especificar múltiples archivos para la nueva base de datos.

LOG ON

Especifica que los archivos del disco utilizados para almacenar el registro de la base de datos (archivos de registro) se han definido explícitamente. La palabra clave va seguida de una lista separada por comas de elementos <filespec> que definen los archivos de registro. Si no se especifica LOG ON, se crea automáticamente un único archivo de registro con un nombre generado por el sistema y un tamaño que es el 25 por ciento de la suma de los tamaños de todos los archivos de datos de la base de datos.

FOR LOAD

Esta cláusula se mantiene por compatibilidad con las versiones anteriores de Microsoft SQL Server. La base de datos se crea con la opción de base de datos **dbo use only** activada y el estado es "cargando". Esto no es necesario en SQL Server versión 7.0 porque la instrucción RESTORE puede volver a crear una base de datos como parte de la operación de restauración.

FOR ATTACH

Especifica que se ha adjuntado una base de datos desde un conjunto existente de archivos del sistema operativo. Debe haber una entrada <filespec> que especifique el primer archivo principal. De las restantes entradas <filespec>, las únicas necesarias son las de cualquier otro archivo que tenga una ruta de acceso distinta de cuando se creó la base de datos por primera vez o se adjuntó por última vez. Debe especificarse una entrada <filespec> para estos archivos. La base de datos adjunta se debe haber creado utilizando la misma página de códigos y ordenación que SQL Server. Utilice el procedimiento almacenado del sistema **sp_attach_db** en lugar de emplear CREATE DATABASE FOR ATTACH directamente. Utilice

CREATE DATABASE FOR ATTACH sólo cuando deba especificar más de 16 elementos <filespec>.

Si adjunta una base de datos a un servidor distinto de aquél del que se separó y la base de datos separada estaba habilitada para duplicación, deberá ejecutar **sp_removedbreplication** para quitar la duplicación de la base de datos.

collation_name

Especifica la intercalación predeterminada de la base de datos. El nombre de intercalación puede ser un nombre de intercalación de Windows o un nombre de intercalación de SQL. Si no se especifica, se asigna a la base de datos la intercalación predeterminada de la instancia de SQL Server.

Para obtener más información acerca de los nombres de intercalación de Windows y SQL, consulte COLLATE.

PRIMARY

Especifica que la lista <filespec> asociada define el archivo principal. El grupo de archivos principal contiene todas las tablas del sistema de base de datos. También contiene todos los objetos no asignados a los grupos de archivos de usuario. La primera entrada <filespec> del grupo de archivos principal pasa a ser el archivo principal, que es el archivo que contiene el inicio lógico de la base de datos y las tablas del sistema. Una base de datos sólo puede tener un archivo principal. Si no se especifica PRIMARY, el primer archivo enumerado en la instrucción CREATE DATABASE se convierte en el archivo principal.

NAME

Especifica el nombre lógico del archivo definido por <filespec>. El parámetro NAME no se requiere cuando se especifica FOR ATTACH.

logical_file_name

Es el nombre utilizado para hacer referencia al archivo en las instrucciones Transact-SQL que se ejecuten después de que se haya creado la base de datos. *logical_file_name* debe ser único en la base de datos y debe seguir las reglas de los identificadores. El nombre puede ser una constante de caracteres o Unicode, o un identificador regular o delimitado.

FILENAME

Especifica el nombre de archivo del sistema operativo del archivo definido por <filespec>.

'os_file_name'

Es la ruta de acceso y nombre de archivo que el sistema operativo utiliza cuando crea el archivo físico definido por <filespec>. La ruta de acceso de *os_file_name* debe especificar un directorio en una instancia de SQL Server. *os_file_name* no puede especificar un directorio en un sistema de archivo comprimido.

Si el archivo se crea en una partición sin formato, *os_file_name* sólo debe indicar la letra de la unidad de una partición sin formato existente. En cada partición sin formato sólo se puede crear un archivo. Los archivos de las particiones sin formato no aumentan automáticamente; por tanto, los parámetros MAXSIZE y FILEGROWTH no son necesarios cuando *os_file_name* especifica una partición sin formato.

SIZE

Especifica el tamaño del archivo definido en <filespec>. Cuando en <filespec> no se especifica un parámetro SIZE para el archivo principal, SQL Server utiliza el tamaño del archivo principal de la base de datos **model**. Cuando en <filespec> no se especifica un parámetro SIZE para un archivo secundario o de registro, SQL Server hace el archivo de 1 MB.

size

Es el tamaño inicial del archivo definido en <filespec>. Se pueden utilizar los sufijos kilobyte (KB), megabyte (MB), gigabyte (GB) o terabyte (TB). El valor predeterminado es MB. Especifique un número entero; no incluya decimales. El valor mínimo de *size* es 512 KB. Si no se especifica *size*, el valor predeterminado es 1 MB. El tamaño especificado para el archivo principal debe tener al menos el tamaño del archivo principal de la base de datos **model**.

MAXSIZE

Especifica el tamaño máximo al que puede aumentar el archivo definido en <filespec>.

max_size

Es el tamaño máximo al que puede aumentar el archivo definido en <filespec>. Se pueden utilizar los sufijos kilobyte (KB), megabyte (MB), gigabyte (GB) o terabyte (TB). El valor predeterminado es MB. Especifique un número entero; no incluya decimales. Si *max_size* no se especifica, el archivo aumenta hasta que el disco esté lleno.

Nota El registro del sistema S/B de Microsoft Windows NT® avisa al administrador del sistema SQL Server cuando un disco está a punto de llenarse.

UNLIMITED

Especifica que el archivo definido en <filespec> aumenta hasta que el disco esté lleno.

FILEGROWTH

Especifica el incremento de crecimiento del archivo definido en <filespec>. El valor FILEGROWTH de un archivo no puede exceder del valor MAXSIZE.

growth_increment

Es la cantidad de espacio que se agrega al archivo cada vez que se necesita más espacio. Especifique un número entero; no incluya decimales. Un valor 0 indica que no hay crecimiento. El valor se puede especificar en MB, KB, GB, TB o bien como un porcentaje (%). Si se especifica un número sin los sufijos MB, KB o %, el valor predeterminado es MB. Cuando se especifica %, el tamaño de incremento de crecimiento es el porcentaje especificado del tamaño del archivo en el momento en que tiene lugar el incremento. Si no se especifica FILEGROWTH, el valor predeterminado es 10 por ciento, y el valor mínimo es 64 KB. El tamaño especificado se redondea al múltiplo de 64 KB más cercano.

Observaciones

Puede utilizar una instrucción CREATE DATABASE para crear una base de datos y los archivos que almacenan la base de datos. SQL Server implementa la instrucción CREATE DATABASE en dos pasos:

1. SQL Server utiliza una copia de la base de datos **model** para inicializar la base de datos y sus metadatos.
2. SQL Server rellena el resto de la base de datos con páginas vacías, excepto las páginas que tengan datos internos que registren cómo se emplea el espacio en la base de datos.

Cualquier objeto definido por el usuario de la base de datos **model** se copiará a todas las bases de datos recién creadas. Puede agregar a la base de datos **model** cualquier objeto, por ejemplo, tablas, vistas, procedimientos almacenados, tipos de datos, etc., que desee incluir en todas las bases de datos.

Cada base de datos nueva hereda los valores opcionales de la base de datos **model** (a menos que se especifique FOR ATTACH). Por ejemplo, la opción

de base de datos **select into/bulkcopy** es OFF en **model** y en cualquier base de datos nueva que se cree. Si utiliza ALTER DATABASE para cambiar las opciones de la base de datos **model**, los valores de estas opciones entrarán en efecto para cualquier base de datos nueva que se cree. Si se especifica FOR ATTACH en la instrucción CREATE DATABASE, la nueva base de datos hereda los valores opcionales de la base de datos original.

En un servidor se puede especificar un máximo de 32.767 bases de datos.

Para almacenar una base de datos se emplean tres tipos de archivos:

- El archivo principal contiene la información de inicio de la base de datos. El archivo principal se utiliza también para almacenar datos. Cada base de datos tiene un único archivo principal.
- Los archivos secundarios almacenan todos los datos que no caben en el archivo principal de datos. Las bases de datos no necesitan archivos de datos secundarios si el archivo principal es suficientemente grande para contener todos los datos de la base de datos. Otras bases de datos pueden ser suficientemente grandes para necesitar múltiples archivos de datos secundarios o pueden utilizar archivos secundarios en discos separados para repartir los datos entre múltiples discos.
- Los archivos del registro de transacciones contienen la información del registro que se utiliza para recuperar la base de datos. Debe haber al menos un archivo de registro de transacciones para cada base de datos, aunque puede haber más de uno. El tamaño mínimo de un archivo de registro de transacciones es de 512 KB.

Cada base de datos tiene al menos dos archivos, un archivo principal y un archivo de registro de transacciones.

Aunque '*os_file_name*' puede ser cualquier nombre de archivo válido del sistema operativo, el nombre reflejará más claramente el propósito del archivo si utiliza las siguientes extensiones recomendadas.

Tipo de archivo	Extensión del nombre de archivo
Archivo de datos principal	.MDF
Archivo de datos secundario	.NDF
Archivo de registro de transacciones	.ldf

Nota Cada vez que se crea una base de datos de usuario, se debe hacer una copia de seguridad de la base de datos **master**.

No es posible especificar fracciones en los parámetros SIZE, MAXSIZE y FILEGROWTH. Para especificar una fracción de un megabyte en los parámetros SIZE, conviértalos a kilobytes; para ello, multiplique el número por 1.024. Por ejemplo, especifique 1.536 KB en lugar de 1,5 MB (1,5 por 1.024 es igual a 1.536).

Cuando se especifica una instrucción individual CREATE DATABASE *database_name* sin parámetros adicionales, la base de datos se crea con el mismo tamaño que la base de datos **model**.

Todas las bases de datos tienen al menos un grupo de archivos principal. Todas las tablas del sistema se encuentran en el grupo de archivos principal. Una base de datos puede tener también grupos de archivos definidos por el usuario. Si se crea un objeto con una cláusula ON *filegroup* que especifica un grupo de archivos definido por el usuario, todas las páginas del objeto se asignan desde el grupo de archivos especificado. Las páginas de todos los objetos de usuario creados sin una cláusula ON *filegroup*, o con una cláusula ON DEFAULT, se asignan desde el grupo de archivos predeterminado. Cuando se crea una base de datos por primera vez, el grupo de archivos principal es el grupo de archivos predeterminado. Con ALTER DATABASE, se puede especificar que un grupo de archivos definido por el usuario sea el grupo de archivos predeterminado:

```
ALTER DATABASE database_name MODIFY FILEGROUP filegroup_name
DEFAULT
```

Cada base de datos tiene un propietario con capacidad para realizar actividades especiales en la base de datos. El propietario es el usuario que crea la base de datos. El propietario de la base de datos se puede cambiar mediante **sp_changedbowner**.

Para mostrar un informe de una base de datos o de todas las bases de datos de una instancia de SQL Server, ejecute **sp_helpdb**. Para obtener un informe acerca del espacio utilizado en una base de datos, emplee **sp_spaceused**. Para obtener un informe de los grupos de archivos de una base de datos, utilice **sp_helpfilegroup**, y utilice **sp_helpfile** para obtener el informe de los archivos de la base de datos.

Las versiones anteriores de SQL Server utilizaban las instrucciones DISK INIT para crear los archivos de la base de datos antes de que se ejecutara la instrucción CREATE DATABASE. Por motivos de compatibilidad con versiones anteriores de SQL Server, la instrucción CREATE DATABASE puede crear también una base de datos nueva con los archivos o dispositivos creados con la instrucción DISK INIT. Para obtener más información, consulte Detalles de compatibilidad con versiones anteriores de SQL Server.

Permisos

De forma predeterminada, los permisos de CREATE DATABASE son los de los miembros de las funciones fijas de servidor **sysadmin** y **dbcreator**. Los miembros de las funciones fijas de servidor **sysadmin** y **securityadmin** pueden conceder permisos CREATE DATABASE a otros inicios de sesión. Los miembros de las funciones fijas de servidor **sysadmin** y **dbcreator** pueden agregar otros inicios de sesión a la función **dbcreator**. El permiso CREATE DATABASE debe concederse explícitamente; no se concede mediante la instrucción GRANT ALL.

El permiso CREATE DATABASE se limita normalmente a unos cuantos inicios de sesión para mantener el control de la utilización de los discos de una instancia de SQL Server.

Ejemplos

A. Crear una base de datos que especifica los archivos de datos y de registro de transacciones

Este ejemplo crea una base de datos llamada **Sales**. Debido a que no se utiliza la palabra clave PRIMARY, el primer archivo (**Sales_dat**) se convierte en el archivo principal. Como no se especifican MB ni KB en el parámetro SIZE del archivo **Sales_dat**, de forma predeterminada utiliza MB y el tamaño se asigna en megabytes. El tamaño del archivo **Sales_log** se asigna en megabytes porque se ha indicado explícitamente el sufijo MB en el parámetro SIZE.

```
USE master
GO
CREATE DATABASE Sales
ON
( NAME = Sales_dat,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\saledat.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = 'Sales_log',
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\salelog.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
GO
```

B. Crear una base de datos mediante la especificación de múltiples archivos de datos y de registro de transacciones

Este ejemplo crea una base de datos llamada **Archive** con tres archivos de datos de 100 MB y dos archivos de registro de transacciones de 100 MB. El archivo principal es el primer archivo de la lista y se especifica explícitamente con la palabra clave PRIMARY. Los archivos de registro de transacciones se especifican a continuación de las palabras clave LOG ON. Observe las extensiones que se emplean para los archivos de la opción FILENAME: .MDF se utiliza para los archivos de datos principales, .NDF para los archivos de datos secundarios y .LDF para los archivos de registro de transacciones.

```
USE master
GO
CREATE DATABASE Archive
ON
PRIMARY ( NAME = Arch1,
          FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archdat1.mdf',
          SIZE = 100MB,
          MAXSIZE = 200,
          FILEGROWTH = 20),
( NAME = Arch2,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archdat2.ndf',
  SIZE = 100MB,
  MAXSIZE = 200,
  FILEGROWTH = 20),
( NAME = Arch3,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archdat3.ndf',
  SIZE = 100MB,
  MAXSIZE = 200,
  FILEGROWTH = 20)
LOG ON
( NAME = Archlog1,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archlog1.ldf',
  SIZE = 100MB,
  MAXSIZE = 200,
  FILEGROWTH = 20),
( NAME = Archlog2,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archlog2.ldf',
  SIZE = 100MB,
  MAXSIZE = 200,
  FILEGROWTH = 20)
GO
```

C. Crear una base de datos individual

En este ejemplo se crea una base de datos llamada **Products** y se especifica un único archivo. El archivo especificado se convierte en el archivo principal y se crea automáticamente un archivo de registro de transacciones de 1 MB. Como no se especifican MB ni KB en el parámetro **SIZE** del archivo principal, se asigna en megabytes. Ya que no existe <filespec> para el archivo de registro de transacciones, éste no tiene MAXSIZE y puede crecer hasta llenar todo el espacio disponible en el disco.

```
USE master
GO
CREATE DATABASE Products
ON
( NAME = prods_dat,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\prods.mdf',
  SIZE = 4,
  MAXSIZE = 10,
  FILEGROWTH = 1 )
GO
```

D. Crear una base de datos sin especificar los archivos

Este ejemplo crea una base de datos llamada **mytest** y crea los archivos principal y de registro de transacciones correspondientes. Debido a que la instrucción no tiene elementos <filespec>, el archivo principal de la base de datos tiene el tamaño del archivo principal de la base de datos **model**. El registro de transacciones tiene el tamaño del archivo del registro de transacciones de la base de datos **model**. Como no se ha especificado MAXSIZE, los archivos pueden crecer hasta llenar todo el espacio disponible en el disco.

```
CREATE DATABASE mytest
```

E. Crear una base de datos sin especificar SIZE

Este ejemplo crea una base de datos llamada **products2**. El archivo **prods2_dat** se convierte en el archivo principal, con un tamaño igual al tamaño del archivo principal de la base de datos **model**. El archivo de registro de transacciones se crea automáticamente y es un 25 por ciento del tamaño del archivo principal, o 512 KB, el que sea mayor. Como no se ha especificado MAXSIZE, los archivos pueden crecer hasta llenar todo el espacio disponible en el disco.

```
USE master
GO
CREATE DATABASE Products2
ON
( NAME = prods2_dat,
```

```
FILENAME = 'c:\program files\microsoft sql  
server\mssql\data\prods2.mdf' )  
GO
```

F. Crear una base de datos con grupos de archivos

Este ejemplo crea una base de datos llamada **sales** con tres grupos de archivos:

- El grupo de archivos principal con los archivos **SPri1_dat** y **SPri2_dat**. El incremento de FILEGROWTH de estos archivos se especifica como el 15%.
- Un grupo de archivos llamado **SalesGroup1** con los archivos **SGrp1Fi1** y **SGrp1Fi2**.
- Un grupo de archivos llamado **SalesGroup2** con los archivos **SGrp2Fi1** y **SGrp2Fi2**.

```
CREATE DATABASE Sales  
ON PRIMARY  
( NAME = SPri1_dat,  
  FILENAME = 'c:\program files\microsoft sql  
server\mssql\data\SPri1dat.mdf',  
  SIZE = 10,  
  MAXSIZE = 50,  
  FILEGROWTH = 15% ),  
( NAME = SPri2_dat,  
  FILENAME = 'c:\program files\microsoft sql  
server\mssql\data\SPri2dt.ndf',  
  SIZE = 10,  
  MAXSIZE = 50,  
  FILEGROWTH = 15% ),  
FILEGROUP SalesGroup1  
( NAME = SGrp1Fi1_dat,  
  FILENAME = 'c:\program files\microsoft sql  
server\mssql\data\SG1Fi1dt.ndf',  
  SIZE = 10,  
  MAXSIZE = 50,  
  FILEGROWTH = 5 ),  
( NAME = SGrp1Fi2_dat,  
  FILENAME = 'c:\program files\microsoft sql  
server\mssql\data\SG1Fi2dt.ndf',  
  SIZE = 10,  
  MAXSIZE = 50,  
  FILEGROWTH = 5 ),  
FILEGROUP SalesGroup2  
( NAME = SGrp2Fi1_dat,
```

```
FILENAME = 'c:\program files\microsoft sql
server\mssql\data\SG2Fi1dt.ndf',
    SIZE = 10,
    MAXSIZE = 50,
    FILEGROWTH = 5 ),
( NAME = SGrp2Fi2_dat,
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\SG2Fi2dt.ndf',
    SIZE = 10,
    MAXSIZE = 50,
    FILEGROWTH = 5 )
LOG ON
( NAME = 'Sales_log',
  FILENAME = 'c:\program files\microsoft sql
server\mssql\data\salelog.ldf',
    SIZE = 5MB,
    MAXSIZE = 25MB,
    FILEGROWTH = 5MB )
GO
```

G. Adjuntar una base de datos

El ejemplo B crea una base de datos llamada **Archive** con los siguientes archivos físicos:

```
c:\program files\microsoft sql server\mssql\data\archdat1.mdf
c:\program files\microsoft sql server\mssql\data\archdat2.ndf
c:\program files\microsoft sql server\mssql\data\archdat3.ndf
c:\program files\microsoft sql server\mssql\data\archlog1.ldf
c:\program files\microsoft sql server\mssql\data\archlog2.ldf
```

La base de datos puede separarse con el procedimiento almacenado **sp_detach_db** y, después, puede volverse a adjuntar mediante **CREATE DATABASE** con la cláusula **FOR ATTACH**:

```
sp_detach_db Archive
GO
CREATE DATABASE Archive
ON PRIMARY (FILENAME = 'c:\program files\microsoft sql
server\mssql\data\archdat1.mdf')
FOR ATTACH
GO
```

H. Utilizar particiones sin formato

Este ejemplo crea una base de datos llamada **Employees** con particiones sin formato. Las particiones sin formato deben existir cuando se ejecuta la

instrucción y en cada partición sin formato sólo puede almacenarse un archivo.

```
USE master
GO
CREATE DATABASE Employees
ON
( NAME = Empl_dat,
  FILENAME = 'f:',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = 'Sales_log',
  FILENAME = 'g:',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
GO
```

I. Utilizar unidades montadas

Este ejemplo crea una base de datos denominada **Employees** utilizando unidades montadas que apuntan a particiones sin formato. Esta característica sólo está disponible en Microsoft® Windows® 2000 Server. Las unidades montadas y las particiones sin formato deben existir cuando se ejecuta la instrucción, y en cada partición sin formato sólo puede almacenarse un archivo. Cuando se crea un archivo de base de datos en una unidad montada, la ruta de acceso a la unidad debe terminar con una barra diagonal inversa (\).

```
USE master
GO
CREATE DATABASE Employees
ON
( NAME = Empl_dat,
  FILENAME = 'd:\sample data dir\',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = 'Sales_log',
  FILENAME = 'd:\sample log dir\',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
GO
```

©1988-2000 Microsoft Corporation. Reservados todos los derechos.