

microsoft.public.sqlserver.server: Re: 'PRIMARY' filegroup is full – no it's not!

Re: 'PRIMARY' filegroup is full – no it's not!

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.server/2004-12/0772.html>

From: Tom Moreau (*tom_at_dont.spam.me.cips.ca*)

Date: 12/06/04

Date: Mon, 6 Dec 2004 16:34:13 -0500

Consider using `sp_helpfile` to report the sizes.

--
Tom

Thomas A. Moreau, BSc, PhD, MCSE, MCDBA
SQL Server MVP
Columnist, SQL Server Professional
Toronto, ON Canada
www.pinnaclepublishing.com

"Dan Guzman" <guzmanda@nosspam-online.sbcglobal.net> wrote in message
news:eYJnuJ62EHA.4072@TK2MSFTNGP10.phx.gbl...
`sp_helpdb` should report the increased size immediately after the ALTER. If
it shrinks afterward, perhaps a maintenance plan is setup to 'remove unused
space from data files'.

--
Hope this helps.

Dan Guzman
SQL Server MVP

"Ray Scott" <scott@csgsolutions.com> wrote in message
news:OalrpF62EHA.3504@TK2MSFTNGP12.phx.gbl...

> Thanks guys for the additional responses. It does seem to be
> auto-shrinking, BUT auto shrink is not turned on. Here's how I
> checked, maybe I'm looking in the wrong place? Enterprise manager,
> right-clicked database, Properties, Options tab. In the Settings
> section, the only checked properties are Auto update statistics, Auto
> create statistics, and Torn page detection. The other settings,
> including Auto shrink, are not checked. Strange, eh?

>
> I'm still puzzled as to why SQL Server doesn't seem to recognize the new
> large file size after I expand the file using "ALTER DATABASE DataLoad
> MODIFY FILE(NAME='DataLoad_Data', SIZE=12000MB)". I do not see the
> larger 12 GB file size reflected in either `sp_helpdb` or `sp_spaceused` -
> these SPS still report the old 5 GB size.

>
> Thanks again for your help,
> Ray Scott

>
> =====

> From: Dan Guzman

>
> Tom is right about auto-shrink. I never turn on this option in
> production
> and shrink manually if there is a decrease in estimated space
> requirements.

Re: 'PRIMARY' filegroup is full – no it's not!

microsoft.public.sqlserver.server: Re: 'PRIMARY' filegroup is full – no it's not!

```
> For best performance, it's best to pre-allocate the space needed.
>
> With your single file group, you can calculate the available space in
> your
> data files by subtracting the reserved space reported by sp_spaceused
> from
> the sum of the data file sizes reported by sp_helpdb.
>
> --
> Hope this helps.
>
> Dan Guzman
> SQL Server MVP
>
> "Ray.Net" <scott@csgsolutions.com> wrote in message
> news:10ald257.0412052210.609fb2c9@posting.google.com...
>> Thank you for the quick replies - these commands do help me get a
>> better picture of the true database size. However, things are still
>> not adding up properly. Please bear with me and I'll explain:
>>
>> After running DBCC UPDATEUSAGE (0), the output of sp_spaceused is:
>>
>> database_name database_size unallocated space
>> DataLoad 5081.06 MB 107.53 MB
>>
>> reserved data index_size unused
>> 4928224 KB 2632800 KB 2204624 KB 90800 KB
>>
>> So, the db is approximately 5 GB in size. And when I look at the mdf
>> data file, it is indeed 5 GB.
>>
>> According to sp_spaceused, my largest table is about 4 GB in size. So
>> using Dan's 120% rule, I need just under 5GB of free space, in
>> addition to the 5 GB already used, for a total of about 10 GB. Just
>> to allow more than enough margin for error, I went ahead and allocated
>> 12 GB:
>>
>> ALTER DATABASE DataLoad MODIFY FILE(NAME='DataLoad_Data',
>> SIZE=12000MB)
>>
>> 'DataLoad_Data' is the primary filegroup, and the only filegroup for
>> the database. And I can see that after running the ALTER DATABASE
>> command, the mdf file is now 12,288,000 KB in size, just as expected.
>>
>> Question 1: sp_spaceused still gives exactly the same output.
>> Shouldn't it reflect this new larger 12 GB size somewhere in the
>> output - I expected the "database_size" or "reserved" number to jump
>> up, but they didn't.
>>
>> Question 2: Even with the database at this huge new size, my attempt
>> to rebuild an index fails with the "Could not allocate space" error,
>> and the mdf file has shriveled itself back down to 5 GB (or could
>> something be shrinking the file back down to 5 GB before the index
>> creation has finished, which would account for the error?). And I'm
>> not even rebuilding the index on the largest table -- the table whose
>> index I'm recreating only takes up a measly 500 MB. It's not
>> necessarily a problem with this particular index; I've also hit the
>> error on rebuilding a different index, and on a large INSERT
>> operation. Obviously I am missing something here - any ideas?
>>
>> For the record, here's how I'm attempting to recreate the index:
>> CREATE CLUSTERED INDEX idx_Exceptions_LoadID ON dbo.Exceptions
```

microsoft.public.sqlserver.server: Re: 'PRIMARY' filegroup is full – no it's not!

```
>> (LoadId)
>> WITH FILLFACTOR = 90, DROP_EXISTING ON [PRIMARY]
>>
>> Thanks again for sharing your expertise!
>>
>> Ray Scott
>>
>> "Dan Guzman" <guzmanda@nospam-online.sbcglobal.net> wrote in message
>> news:<#u$VEzt2EHA.1564@TK2MSFTNGP09.phx.gbl>...
>>> It's a good practice to pre-allocate enough space to handle
> anticipated
>>> space requirements. Allow autogrow only as a safety net rather than
>>> allow
>>> files to grow routinely during normal operation. This includes both
>>> normal
>>> growth as well as space needed for maintenance.
>>>
>>> Rebuilding a clustered index requires about 120% of the table size so
> you
>>> should have free space of at least 1.2 times the largest table in
> your
>>> database. As Tom suggested, you can use DBCC UPDATEUSAGE or
> sp_spaceused
>>> @updateusage=true to correct inaccurate space reporting.
>>>
>>> --
>>> Hope this helps.
>>>
>>> Dan Guzman
>>> SQL Server MVP
>>>
>>> "Ray.Net" <scott@csgsolutions.com> wrote in message
>>> news:10ald257.0412050543.11b31a32@posting.google.com...
>>> >I have a SQL Server 2000 database, in which the size of the primary
>>> > filegroup is 5 GB. The disk where the data resides has 14 GB of
> free
>>> > space - that should be more than enough free space for a 5 GB
>>> > database, right?
>>> >
>>> > The problem is that when I perform certain operations (esp.
> recreating
>>> > a clustered index), it still seems to run out of space with this
>>> > error:
>>> > Could not allocate space for object 'ActivityTracker' in database
>>> > 'DataLoad' because the 'PRIMARY' filegroup is full. The statement
> has
>>> > been terminated.
>>> >
>>> > Here's where it gets weird: in the database properties
> "Automatically
>>> > grow file" IS checked, and Maximum file size is unrestricted. So
>>> > there's lots of free space (nearly 3X the size of the db), and the
>>> > growth is not restricted - what makes it think the filegroup is
> full?
>>> >
>>> > I found a few newsgroup postings that suggested sometimes the
>>> > auto-grow cannot grow fast enough, so you should manually increase
> the
>>> > size of the database. I tried this, using "alter database" to raise
>>> > the size of the primary filegroup as high as 12GB - more that twice
>>> > its original 5GB size - but it still fails with the error message
>>> > above. And - here's another weird thing - at the time the error
```

microsoft.public.sqlserver.server: Re: 'PRIMARY' filegroup is full – no it's not!

```
>>> > occurs, the database shrinks itself back down to 5 GB (even though
> I
>>> > do NOT have the "Auto shrink" property checked). What's going on
>>> > here?
>>> >
>>> > I appreciate any ideas you can offer on why this database seems to
> run
>>> > out of space when there's plenty of space to be had.
>>> >
>>> > Ray
>
>
>
>
> *** Sent via Developersdex http://www.developersdex.com ***
> Don't just participate in USENET...get rewarded for it!
```