RE: Insufficient memory Available

Source:
http://www.tech−archive.net/Archive/SQL−Server/microsoft.public.sqlserver.server/2004−03/1961.html

---

From: Baisong Wei[MSFT] (*v−baiwei_at_online.microsoft.com*)
Date: 03/13/04

Date: Sat, 13 Mar 2004 07:19:19 GMT

Hi Rathna,

Thank you for using the newsgroup.

>From the information you provided, you have transactional update done by
COM+ objects using ADO and MSDTC
You got error from ADO
−2147467259 : Unspecified error.
and error from SQL Server:
Error: 17803, Severity: 20, State: 12
Insufficient memory available.

You have found another post regarding the similar problem. But you problem seems somewhat different. Actually, these problem is very complicated and various factors might cause this problem and the factors will not just within the scope of SQL Server. I would explain it later. From my experience, this problem is very complicated and many advanced tools may need to be invoved which are not supported in the newsgroup. Therefore, we probably will not be able to resolve the issue through the newsgroups. I'd recommend opening a Support incident with Microsoft Support Services, so that a dedicated Support Professional can assist with your advisory case. Please be advised that contacting phone support will be a charged call.

To obtain the phone numbers for specific technology request please take a look at the web site listed below.
http://support.microsoft.com/default.aspx?scid=fh;EN−US;PHONENUMBERS

If you are outside the US please see http://support.microsoft.com for regional support phone numbers.

I would still make some explanation on this error message, and try to make an explanation of the SQL Server memory, I hope it might be helpful for you :

There are two main areas of memory within SQL Server's address space, the buffer pool (BPool) and a smaller memory pool sometimes called the "MemToLeave" area. Consumers of the BPool include cached data; workspace

memory used during execution of queries, for example for in−memory sorts or hashes; most cached

procedure and query plans; memory for locks and other internal structures; and most other miscellaneous memory needs of the SQL Server.

MemToLeave is the other significant memory area, and it is primarily used by non−SQL Server code that happens to be executing within the SQL Server process.

The MemToLeave area is memory that is left unallocated and unreserved for non−BPool memory consumers. Some examples of the components that may use this

memory include extended stored procedures, OLE Automation/COM objects running inside SQL Server's address space, linked server OLEDB providers and ODBC

drivers, MAPI components used by SQLMail, and thread stacks (1/2 MB per thread). This does not just include the .EXE and .DLL binary images for these components;

non−SQL Server code makes its memory allocation requests directly from the OS, not from the SQL Server buffer pool. Because the entire BPool is reserved at

server startup, these memory requests must be satisfied from the MemToLeave area, which is the only source of unreserved memory in the SQL Server address

space. SQL Server itself also uses the MemToLeave memory area for certain allocations that will not fit on a single 8KB buffer in the BPool.

The size of the MemToLeave memory area is fixed when SQL Server starts up and is calculated as follows:

[max worker threads] * 0.5MB + [−g Memory]

"−g" is an optional SQL Server startup parameter that can be used to increase the amount of memory that is left unreserved as the MemToLeave area. The default
−g memory size is 128MB in SQL Server 7.0 and 256MB in SQL Server 2000, and the default sp_configure value for max worker threads is 255, so the amount of

non−Bpool memory set aside as MemToLeave on a SQL Serve 7.0 server that has had neither of these options changed will be 255 * 0.5 + 128 = 256MB (384MB on SQL

Server 2000 because the default −g value is changed to 256MB). This amount is sufficient for the large majority of server installations. Increasing

the MemToLeave size without justification will deprive the BPool of memory that could be used to cache objects and data and could negatively impact performance.

The sp_configure options 'max server memory' and 'min server memory' determine how much memory is committed for the SQL Server BPool (these options do not

apply to the memory allocated within MemToLeave because SQL doesn't have direct control over the majority of the allocations in this area). SQL Server dynamically resizes BPool according the rules described in the Books Onlinearticles "Memory Architecture" and "Server Memory Options". The maximum size of BPool can be changed at runtime by adjusting the sp_configure value 'max server memory' and the change will take effect immediately. This is possible because the maximum possible BPool size is reserved when SQL Server starts up. Because of this, it is not possible to increase the size of the MemToLeave area by decreasing 'max server memory'. In most cases on machines dedicated to a single SQL Server instance, decreasing 'max server memory' simply wastes RAM and will not prevent insufficient memory errors related to the MemToLeave area.

Since your application is COM+ objects using ADO and MSDTC, it is possible caused by shortage of membery in MemToLeave.

Causes of MemToLeave pressure:

1. Large numbers of simultaneous rollbacks. Each rollback requires a 64KB llocation from MemToLeave.

2. Large numbers of large stored procedures. Look at OS memory pages from DBCC MEMORYSTATUS. If OS memory pages explains MemToLeave pressure, DBCC FREEPROCCACHE; if OS pages are reduced, procedure cache is to blame for the memory pressure.

3. Heavy concurrent linked server activity. SQLOLEDB and the SQL Server ODBC driver can reserve up to 7MB per connection.

4. Memory leaks or large memory allocations or reservations within COM objects or extended stored procedures running within the SQL Server address space. COM

objects can be run out of process via an optional sp_OACreate parameter; extended stored procedures can be run outside of the production SQL Server's

address space.

5. Using 'network packet size' 8192: Q238615

If the 17803 and associated errors point to a shortage of memory in MemToLeave:

1. Increase the amount of memory set aside for MemoToLeave by adding −g256
(SQL 7.0) or −g384 (SQL 2000) as a startup parameter. This will give you
additional

room for thread stacks, linked server activity, etc.

But it seems it will not work since you have assign more memory to SQL
Server.

2. If the server makes linked server connections to other SQL Servers using
SQLOLEDB or MSDASQL, upgrade to MDAC 2.5 or MDAC 2.6 on the server. The MDAC
2.5 and 2.6 versions of SQLOLEDB are more conservative with their initial
memory allocations. For more information see the following KB article:

Q258242 PRB: How the SQLOLEDB provider allocates memory for the IRowset

Note that it is very important that you do not install MDAC 2.6 on a
SQLServer .0 MSCS cluster. If SQL 7.0 is not clustered or if it is SQL
Server 2000 then you should be able to safely install MDAC 2.6.

These are what I could explain to you and I hope you could have some
overall point of view of it and you would better to opening a Support
incident with Microsoft Support Services.

Hope this helps. Thanks for understanding.

Sincerely Yours

Baisong Wei
Microsoft Online Support
−−
Get Secure! − www.microsoft.com/security
This posting is provided "as is" with no warranties and confers no rights.
Please reply to newsgroups only. Thanks.