

Essbase error 1042006 when API tries to do too many connections from one machine in quick succession

Note: this document was originally obtained from Hyperion tech support sources and has been edited to remove identifying information for the customer, etc and to place context into the document. We also included notes from our internal testing with the olapunderground Outline Extractor tool. If you have other comments about this document which others may find helpful, please let us know and we will try to incorporate them into this document.

– Tim Tow (timtow@appliedolap.com)

Occasionally, we have heard of the Essbase API failing when it tries to do too many connections in quick succession on the Windows operating system. One of the more frequent places we have seen this occurring is in the olapunderground Outline Extractor which does literally thousands to hundreds of thousands of calls (or more) to the API to get outline information. The typical scenario seen in that product is that parent member information may be missing for members in the extract file.

The problem relates to the port numbers used on the client. Those ports are ephemeral (“briefly used”) port numbers. The Windows default for the `TcpTimedWaitDelay` is 240 seconds (valid values are 30-300) and for the `MaxUserPort` is 5000 (valid values are 5,000-65,534). The default values essentially limit the number of ephemeral ports available and the API runs out of ports to use for the connection. Adjustment of the `MaxUserPort` and `TcpTimedWaitDelay` settings in the Windows Registry may fix the error. Other alternatives to solving this issue include modifying the API code to avoid a massive number of calls to the application in a short period of time. For member manipulation calls, for example, you may try to get the outline to the client machine and then lookup the attributes of the members using a local copy of the outline.

The values of these 2 settings determine how many connections can open on the client side and how long those connections last. You can examine how many client ports are in a `TIME_WAIT` state by using the `Netstat` tool on the client computer. Run the `Netstat` tool with the `-n` flag and count the number of client sockets to your Server IP address that are in a `TIME_WAIT` state. Note that the `MaxUserPort` and `TcpTimedWaitDelay` settings are applicable only for a client computer that is rapidly opening and closing connections to a remote computer.

When you use the TCP/IP protocol to open a connection to a computer that is running Essbase, the underlying network library opens a TCP/IP socket to the that computer. When it opens this socket, the network library does not enable the `SO_REUSEADDR` TCP/IP socket option. Note that the Essbase network library specifically does not enable the `SO_REUSEADDR` TCP/IP socket option for security reasons. When `SO_REUSEADDR` is

enabled, a malicious user can hijack a client port to Essbase and use the credentials that the client supplies to gain access to the computer that is running Essbase. By default, because the network library does not enable the SO_REUSEADDR socket option, every time you open and close a socket through the network library on the client side, the socket enters a TIME_WAIT state for four minutes (240 seconds using the default TcpTimedWaitDelay). If you are rapidly opening and closing connections over TCP/IP, you are rapidly opening and closing TCP/IP sockets. In other words, each connection has one TCP/IP socket. If you rapidly open and close approximately 4000 sockets in less than 240 seconds, you will reach the default maximum setting for client anonymous ports, and new socket connection attempts fail until the existing set of TIME_WAIT sockets times out.

In our testing with the olapunderground Outline Extractor, we noted the following items while debugging a reported issue and testing these registry adjustments:

- We replicated the problem on Windows XP but, despite some limited efforts, did not replicate the issue on Windows 2000; we did not try to replicate the issue on Windows 2003. Our tests were limited to a single outline submitted by an Outline Extractor user.
- The registry key 'MaxUserPort' did not appear to exist by default in the Windows XP registry. We had to create it and, in our test case, a value of 12000 solved the issue. It seems logical, however, that the processor speed of the client machine, combined with the code path of the actual API code, could have a tremendous effect on whether the number of ports becomes an issue.
- After changing this registry key, we needed to reboot XP for the new setting to take effect.

Note: the below was apparently provided by a Hyperion engineer; the name has been removed to maintain keep the identity anonymous. This section is quite technical and goes into the background of how the ports work.

Registered Ports, ports between 1024 and 49151, are listed by the IANA and on most systems can be used by applications or programs executed by users. Table C.2 specifies the port used by the server process as its contact port. The IANA registers uses of these ports as a convenience to the Internet community. To the extent possible, these same port assignments are used with UDP. The Registered Ports are in the numerical range of 1024-49151. The Registered Ports between 1024 and 5000 are also referred to as the Ephemeral Ports. At least on Windows, the TCP stack (OS) re-uses these ports internally on every socket connection cycling from 1024...5000 and wrapping around to 1024 again. This could lead to some interesting problems if sockets are opened and close very quickly as there is usually a time delay before that port is made available again...

Second, the number of user-accessible ephemeral ports that can be used to source outbound connections is configurable with the MaxUserPort registry entry

(HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters). By default, when an application requests any socket from the system to use for an outbound call, a port between the values of 1024 and 5000 is supplied. You can use the MaxUserPort registry entry to set the value of the highest port number to be used for outbound connections. For example, setting this value to 10000 would make approximately 9000 user ports available for outbound connections. For more details, see RFC 793. See also the MaxFreeTcbs and MaxHashTableSize registry settings (HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters).

Below are also excerpts from Microsoft website and the links for your references:

- TcpTimedWaitDelay

Determines the time that must elapse before TCP can release a closed connection and reuse its resources. This interval between closure and release is known as the TIME_WAIT state or 2MSL state. During this time, the connection can be reopened at much less cost to the client and server than establishing a new connection.

Registry key= HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Data type=REG_DWORD

Default value= 0xF0 (240 seconds = 4 minutes)

Valid values=0x1E to 0x12C (30 to 300 seconds)

- MaxUserPort

Determines the highest port number TCP can assign when an application requests an available user port from the system. Typically, ephemeral ports (those used briefly) are allocated to port numbers 1024 through 5000.

Registry key=HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Data type= REG_DWORD

Default value=5000

Valid values= 5,000 to 65,534 (port numbers)

For more information, see the following link:

<http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/regentry/58811.asp>