

APLNext Supervisor – Application System Considerations

The APLNext Supervisor is a program to efficiently schedule and control multiple instances of the APL+Win ActiveX calculation engine to improve the performance of an APL+Win application system that processes a series of similar inputs.

A ‘controlling application’, e.g. APL+Win or C#, creates an instance of the APLNext Supervisor. The ‘controlling application’ then provides a configuration to the APLNext Supervisor which indicates, among other parameters, the minimum and maximum number of instances of the APL+Win ActiveX engine that will be created by the APLNext Supervisor. The ‘controlling application’ starts the APLNext Supervisor and then submits processing requests to the APLNext Supervisor which are queued up by the APLNext Supervisor. The APLNext Supervisor assigns these processing requests to the pool of APL+Win ActiveX engine instances until the request queue has been satisfied. The assignment of a processing request by the APLNext Supervisor to an APL+Win ActiveX engine instance is done asynchronously so that each processing request is associated with an independent ‘thread’.

Microsoft Windows Memory Limits on Win32 Application Instances:

The APL+Win ActiveX engine is a Win32 program, so the APLNext Supervisor must also be a Win32 program so that it can create instances of the APL+Win ActiveX engine. The Microsoft Windows memory limitation of approximately 2Gb applies to an instance of the APLNext Supervisor as well as each instance of the APL+Win ActiveX engine.

The limitation of each instance of the APL+Win ActiveX engine to 2Gb of memory is familiar to experienced APL+Win application programmers which can result in a ‘WSFULL’ error message.

This Microsoft memory limitation on Win32 applications also affects the number of requests which can simultaneously exist in the APLNext Supervisor processing request queue. It is not possible to state in advance the maximum number of processing requests which can be contained in the APLNext Supervisor processing request queue. An application-specific processing request contains several APL+Win programmer-controlled elements including:

- A processing request identifier
- The name of the APL+Win function which will be used to satisfy the processing request, aka the ‘kernel’ function.
- The left argument, if any, of the ‘kernel’ function
- The right argument of the ‘kernel’ function
- The result of the ‘kernel’ function execution upon the arguments provided in the request. Note that this result is maintained in the APLNext Supervisor processing request queue only until the ‘controlling application’ has responded to the firing of the APLNext Supervisor ‘request completed’ event corresponding to a particular processing request.

All processing requests in the APLNext Supervisor queue consume part of the Windows memory available to the APLNext Supervisor instance. If the 'kernel' function arguments or result objects require large amounts of memory, the number of processing requests which can be contained in the APLNext Supervisor processing request queue will be reduced.

For application systems in which the 'kernel' function performs significant processing, it can be beneficial for the 'kernel' function to receive arguments which indicate the file location of the input rather than the actual memory-intensive input. Analogously if the 'kernel' function output is also memory-intensive, it may be better to save its output to a file and have the 'kernel' function return the location of that output.

APLNext Supervisor MinPool and MaxPool Parameters

When the APLNext Supervisor 'maxpool' configuration parameter is set, this indicates the maximum number of APL+Win ActiveX engine instances that the APLNext Supervisor instance will create. The 'minpool' configuration parameter indicates the number of actual APL+Win ActiveX engine instances that will be created when the APLNext Supervisor is started.

The APLNext Supervisor automatically adjusts the actual number of APL+Win ActiveX engine instances between the minpool and maxpool as it assigns processing requests to instances of the APL+Win ActiveX engine. The APLNext Supervisor may create and subsequently delete an APL+Win ActiveX engine instance as the request queue is increased or diminished. However, the APLNextSupervisor will always keep active a set of APL+Win ActiveX engine instances equal to the 'minpool' value.

When the next processing request in the queue is identified, the APLNext Supervisor executes a 'round robin' to determine if within the existing instances of the APL+Win ActiveX engine there is a free one to satisfy the current processing request. If no free instances of the APL+Win ActiveX engine are available, APLNext Supervisor checks if the maxpool has been reached.

If it has been reached, the pending request remains in the queue and APLNext Supervisor waits 50 milliseconds before trying the 'round robin' again.

If the maxpool has not been reached, a check is made to determine if a new thread can be created from the Windows ThreadPool. If permitted by Windows, the APLNext Supervisor creates a new instance of the APL+Win ActiveX engine. If the Windows ThreadPool limitation applies, the processing request remains in the queue and APLNext Supervisor waits 50 milliseconds before trying again.

To decide if a new instance of APL+Win ActiveX engine needs to be created these parameters are considered:

- The APLNext Supervisor 'maxpool' has not been reached.
- There is at least 4Mb of free memory in the system.
- The number of running threads is less than 200.
- The number of APL+Win ActiveX engines running at the same time has not reached the Windows ThreadPool limit which is controlled by the Windows operating system.

Number of Processors

It is recommended that the APL+Win programmer consider the number of processors when establishing the APLNext Supervisor 'minpool' and 'maxpool' parameter values. For example, it is unlikely that setting the 'maxpool' greater than the number of processors will benefit application performance.

The APLNext Supervisor does not explicitly consider the number of processors on the host workstation. The APLNext Supervisor relies on the Windows operating system to appropriately assign processing 'threads', created by application systems, to the available host workstation processors. The APLNext Supervisor creates independent processing 'threads' that it associates with instances of the APL+Win ActiveX engine used to run the 'kernel' function to satisfy a queued-up APLNext Supervisor processing request. Microsoft Windows assigns these 'threads' to the available processors without intervention by the APLNext Supervisor.

Because Microsoft Windows controls the assignment of processing 'threads' to processors, there is not necessarily a 1:1 relationship with the number of APL+Win ActiveX instances and the number of processors.

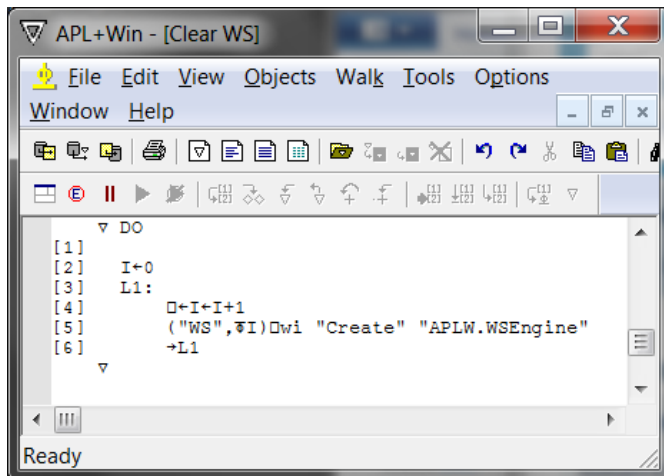
Maximum Number of Instances of the APL+Win ActiveX Engine

There is no explicit limit in the APLNext Supervisor program or in APL+Win program which limits the number of APL+Win ActiveX Engine instances on a workstation. Limits outside of the control of the APLNext Supervisor and APL+Win programs do apply however.

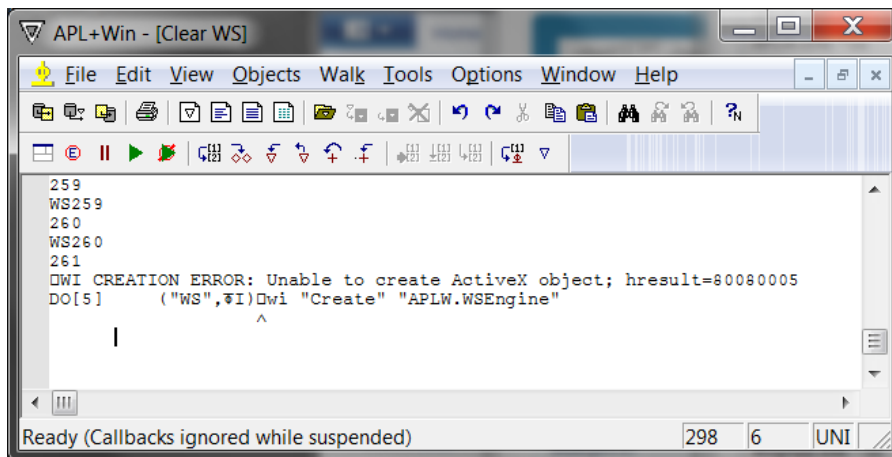
Based on information provided by Microsoft, an apparent limitation on the number of instances of a COM object is 'stress', i.e. poor machine response during heavy processing load, a large number of existing 'threads' or available physical memory reduced to the minimum physical memory required by the Windows operating system:

- <http://social.msdn.microsoft.com/Forums/en-US/architecturegeneral/thread/03527ca5-d743-40e6-9036-c91490b30ec6>
- [http://msdn.microsoft.com/en-us/library/cc704587\(v=PROT.10\).aspx](http://msdn.microsoft.com/en-us/library/cc704587(v=PROT.10).aspx)
- <http://blogs.msdn.com/b/adioltean/archive/2005/06/24/432519.aspx>

A trivial APL+Win function can be used to determine the number of instance of the APL+Win ActiveX calculation engine which can be created on a particular machine. As the number of APL+Win ActiveX engine instances increases, the 'stress' on the workstation is evidently increased. It is instructive to view the Windows Task Manager > Performance panel when the limitation is reached because it illustrates the condition that available physical memory is close to the minimum required for the Windows operating system.



Ultimately the Window HRESULT=80080005 error will apply, for example:



In this case the overall machine memory was 8Gb when the limit was observed. Additional tests using a machine with 16Gb of physical memory indicate that the same 260 limit on COM instances applied.



Available physical memory has been significantly reduced after the 260th instance of APL+Win ActiveX engine has been created and creation of the 261st instance is attempted:

