

# APLNext Supervisor

---

## Multi-threading and use of hardware processors

The APLNext Supervisor provides multi-threading for application-level parallel processing in APL+Win. The Supervisor application enables parallel processing by requesting processing threads from the Microsoft Windows operating system. When a processing thread is provided by the Windows operating system to an application system, the operating system may allocate an available physical or virtual processor (core) to perform processing in that thread. It is the Windows operating system which associates processing threads with processors. Thus the Supervisor has indirect and not explicit access to the target machine's processors through the processing threads that it has received from the Windows operating system.

## Supervisor thread count

The Windows operating system requires at least one processing thread, so at least one of the available processors is allocated by the Windows operating system for this purpose.

Any application systems or services running on the target machine will also request processing threads from the Windows operating system which will allocate available processors to those threads.

The application-specific 'Controlling Application', which creates an instance of the Supervisor, requires at least one processing thread, so at least one of the available processors is allocated by the Windows operating system for this purpose.

When the 'Controlling Application' successfully submits a processing request to the Supervisor, e.g. using the Supervisor 'BeginCall' method, the Supervisor requests two threads from the Windows operating system for each such request. If the Windows operating system provides these threads, two processors, if available, will be allocated by the Windows operating system for this purpose.

- The first thread request is needed by the Supervisor to create an instance of the APL+Win ActiveX engine which will be used to execute the application-specific APL+Win 'Kernel' function.
- The second thread request is needed by the Supervisor to asynchronously wait for the 'Kernel' function processing to be completed. An asynchronous wait is necessary so that the Supervisor instance can handle more than one processing request submitted to it by the 'Controlling Application'.

In some situations, for example, when all available processors are busy, the Windows operating system will not grant the Supervisor additional processing threads, so that the Supervisor will queue up the processing requests submitted by the 'Controlling Application'. A processing request remains in the

Supervisor processing request queue until either two threads become available from the Windows operating system or the application-specific timeout in the Supervisor instance configuration expires.

## **Specifying the 'maxpool' property of the Supervisor configuration**

Each instance of the Supervisor created by the application-specific 'Controlling Application' function has an associated application-specific configuration. That configuration includes the 'maxpool' property which can be used to limit the number of instances of the APL+Win ActiveX engine that the Supervisor will simultaneously create.

Once the 'Controlling Application' has created an instance of the Supervisor, the 'NProcessors' property can be queried to determine the number available on the target machine.

Assuming virtually no other application systems or services running on the target machine besides the Windows operating system and the application-specific 'Controlling Application', setting the value of the 'maxpool' property to a value greater than (NProcessors value – 2) divided by 2 will generally not improve performance of the application system. The number of parallel processing threads is limited by the number of processors.