Threading Models and Component Synchronization

1. In-proc components that have no registered threading model or have registered ThreadingModel=Single always run in the client's primary STA, are accessed only by that STA thread, and so require no synchronization.
2. In-proc components that have registered threading model of ThreadingModel=Apartment may run in any STA in the client's process. Consequently, if the client has more than one STA, then multiple instances of the component may reside in different STAs and the component must protect its shared data, e.g., all global data, static member data in C++ classes, or local static data. Those must be protected with synchronization constructs. If the component class factory is implemented as a singleton, then even non-static member data must be synchronized. Since the class factory can be accessed by concurrent threads, it must be fully synchronized.
3. In-proc components that have registered threading model of ThreadingModel=Free or ThreadingModel=Both the object can be placed in an MTA. In this case, the component and class factory must synchronize access to its member data as well as the shared data described in 2. Also, the component must not use thread local storage, since it doesn't know which thread will call.
4. In-proc components that have the threading model of ThreadingModel=Neutral, threads are allowed to leave STA's or MTA's to access components in an NTA. Synchronization is the same as in 3, that is, the component must be completely synchronized; global data, static and non-static member data, and local static data all have to be synchronized.
5. Out-of-process components that create an STA are serviced only by their own main thread. Requests all place messages in a windows message queue and are serviced sequentially, so the component needs no synchronization.
6. Out-of-process components that create an MTA or NTA are serviced by possibly concurrent RPC threads and must be completely synchronized.

Rules:[1]

1. Any thread that calls COM API or interacts in any way with COM objects must call CoInitialize(Ex).
2. STA threads need message loops unless you are sure that they will run in the client's primary STA, created by its main thread.
3. Never pass unmarshaled interface pointers between apartments.
4. Protect shared data in ThreadingModel=Apartment objects.
5. Objects marked ThreadingModel=Free, ThreadingModel=Both, or ThreadingModel=Neutral must be fully synchronized.
6. Avoid using thread local storage in objects with ThreadingModel settings as in 5.

---

[1] Understanding COM Apartments, Parts I and II, from the column Into the IUnknown, Jeff Prosise, Codeguru.earthweb.com