**Project-4**

**Rationale for Single Threaded Apartment (STA) Model Servers using Comm Channels:**

* + Using remote objects to completely service requests on their own threads means that we either allow an arbitrary number of threads, one for each client, or we throttle the ServiceHost.
	+ It's not a good idea to allow a lot of threads to run concurrently because that will result in a lot of context switching and cause poor performance.
	+ TestHarness
		- Each remote object will be doing a lot of work, so with throttling we will get a lot of failures to connect.
		- With the STA model the service objects are just dropping messages into a shared queue and so are very short-lived. So we don't have to throttle and clients can always connect.
	+ Repository
		- With the STA model we also have the added benefit that in the Repository there will be very little contention for files because only one thread is servicing messages.
		- The only time when that may not be the case is for file upload.
		- One way to handle that is to upload to a receiving directory then the upload service object, when done, sends a message to the Repository’s queue to copy the file(s) into RepoStorage.
		- For sending files (opened for shared reading) we can use child threads to handle each download request. I would probably only do that if the Repository became a performance bottleneck.

**Messages:**

* + **Message Structure:**
		- ToUrl
		- FromUrl
		- Author
		- DateTime
		- MessageType
		- Body - XML
	+ **Message Types:**
		- TestRequest : Client to TestHarness
		- TestResults : TestHarness to Client
		- TestResultsQuery : Client to Repository
		- TestResultsReply : Repository to Client
		- LogsQuery : Client to Repository
		- LogsReply : Repository to Client
		- FilesQuery : Client to Repository
		- FilesReply : Repository to Client
		- FilesRequest : TestHarness to Repository (push model only)