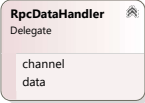
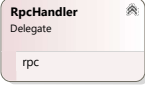


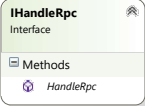
Method to handle an RPC data packet originating from a channel.



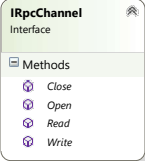
Method to handle an RPC message.



Interface to an object that handles RPC messages.



Interface to an object which reads RPC data packets from a remote caller and writes RPC data packets to a remote caller.



An IRpcChannel object is constructed and given to an RpcConnection. The channel then reads and writes data packet strings for its connection when it is open.

An RPC channel can be any type of IPC channel such as STDIO, Domain Sockets, Named Pipes, TCP, etc.

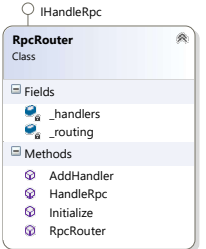
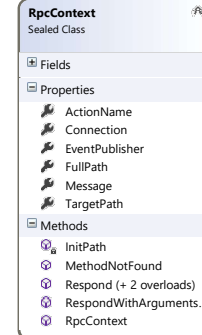
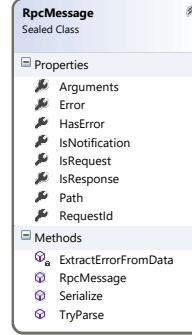
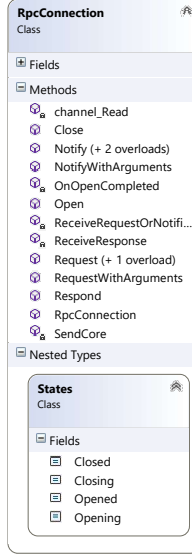
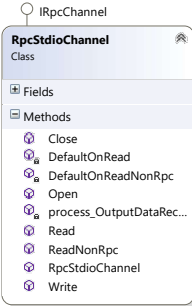
The RpcConnection object creates an RpcMessage object from each JSON data packet received from its channel.

Then it puts that message into a new RpcContext and hands THAT to the primary RpcRouter, which implements IHandleRpc.

RpcMessage has 3 basic main data elements: Path, Arguments and RequestId. All other properties are informational only since they are derived from the main 3 elements and they are not serialized.

RpcContext contains an incoming message, information about the message and any objects that might be needed to fulfill the remote call represented by the message. If the message is a request, the RpcContext can be used to respond to that request.

The primary RpcRouter object is a singleton object shared by all connections. It manages a set of sub-routers which all implement IHandleRpc, similar to RpcRouter itself. When a incoming RPC call arrives, it passes the RpcContext to each one of its handlers in order, until one of them returns true (handled/handling).



#### Diagnostics & Utilities Classes

