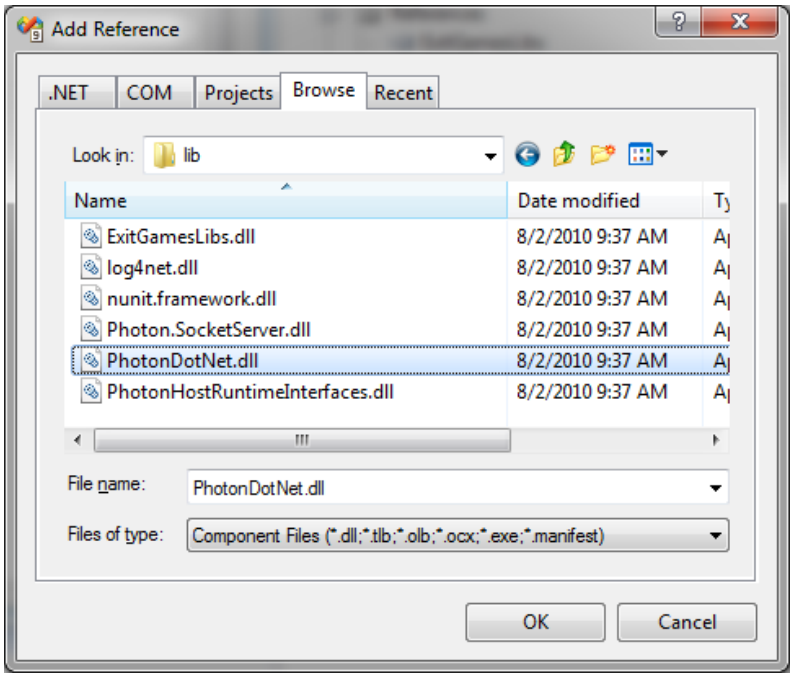
Blank Photon Client Setup

1) Add photon client lib reference to your game project



2) Create IPhotonPeerListener implementation

```
namespace MyPhotonClient
{
    using System;
    using System.Collections;
    using ExitGames.Client.Photon;

    public class MyPhotonGame : IPhotonPeerListener
    {
        private readonly PhotonPeer peer;

        public MyPhotonGame()
        {
            this.peer = new PhotonPeer(this, false);
        }

        public PhotonPeer Peer
        {
            get { return this.peer; }
        }

        public void DebugReturn(string debug)
        {
            Console.WriteLine(debug);
        }

        public void OperationResult(byte opCode, int returnCode,
                                    Hashtable returnValues, short invocID)
        {
            // handle operation response
        }

        public void PeerStatusCallback(int returnCode)
        {
            // handle peer status callback
            switch ((ReturnCode)returnCode)
            {
                case ReturnCode.Connect:
                    {
                        // do something when connected
                        break;
                    }

                case ReturnCode.Disconnect:
                case ReturnCode.DisconnectByServer:
                case ReturnCode.DisconnectByServerLogic:
                case ReturnCode.DisconnectByServerUserLimit:
                    {
                        // do something when disconnected
                        break;
                    }
            }
        }

        public void EventAction(byte eventCode, Hashtable photonEvent)
        {
            // handle events
        }
    }
}
```

3) Add it to the game loop
(code demonstrates the general flow)

```
namespace MyPhotonClient
{
    class Program
    {
        static void Main(string[] args)
        {
            var game = new MyPhotonGame();
            game.Peer.Connect("127.0.0.1:5055", "MyPhotonApp");

            // game loop
            while (true)
            {
                game.Peer.Service();

                // paint frame, etc
            }
        }
    }
}
```