Photon Custom Operation

1) Send custom operation with client, for example after connect in MyPhotonGame (see blank client setup):

```
public void PeerStatusCallback(int returnCode)
{
    // handle peer status callback
   switch ((ReturnCode)returnCode)
   {
      case ReturnCode.Connect:
         // send hello world when connected
         this.Peer.OpCustom(1, new Hashtable { { 100, "Hello World" } }, true);
         break;
```

2) Handle operation on server, file MyPeer.cs (see blank server setup) – simple version:

```
public void OnOperationRequest(OperationRequest request)
{
    // handle operation here (check request.OperationCode)
     switch (request.OperationCode)
     {
        case 1:
        {
           var message = (string)request.Params[100];
           if (message == "Hello World")
           {
              // received hello world, send an answer!
              var response = new OperationResponse(request, 0, "OK",
                   new Dictionary<short, object> { { 100, "Hello yourself!" } } );
              this.PhotonPeer.SendOperationResponse(response);
           }
           else
           {
              // received something else, send an error
              var response = new OperationResponse(request, 1,
                                        "Don't understand, what are you saying?");
              this.PhotonPeer.SendOperationResponse(response);
           }

           break;
        }
     }
}
```

3) Handle operation on server – advanced version:

a) Create operation class

```
namespace MyPhotonServer
{
    using Photon.SocketServer;
    using Photon.SocketServer.Rpc;

    public class MyCustomOperation : Operation
    {
        public MyCustomOperation(OperationRequest request)
            : base(request)
        {
        }

        [RequestParameter(Code = 100, IsOptional = false)]
        [ResponseParameter(Code = 100, IsOptional = false)]
        public string Message { get; set; }
    }
}
```

b) Handle operation, file MyPeer.cs (see blank server setup)

```
public void OnOperationRequest(OperationRequest request)
{
    // handle operation here (check request.OperationCode)
    switch (request.OperationCode)
    {
     case 1:
     {
       var operation = new MyCustomOperation(request);

       // invalid params?
       if (!operation.IsValid)
       {
         // received garbage, send an error
         var response = new OperationResponse(request, 1, "That's garbage!");
         this.PhotonPeer.SendOperationResponse(response);
        }

        if (operation.Message == "Hello World")
        {
         // received hello world, send an answer!
         operation.Message = "Hello yourself!";
         OperationResponse response = operation.GetOperationResponse(0, "OK");
         this.PhotonPeer.SendOperationResponse(response);
        }
        else
        {
         // received something else, send an error
         var response = new OperationResponse(request, 1,
                                    "Don't understand, what are you saying?");
         this.PhotonPeer.SendOperationResponse(response);
        }

        break;
       }
     }
  }
```

4) Handle operation response on client

```
public void OperationResult(byte opCode, int returnCode,
                            Hashtable returnValues, short invocID)
{
    // handle operation response
    switch (opCode)
    {
            // hello world response
            case 1:
            {
                   // OK
                   if (returnCode == 0)
                   {
                          // show the response message
                          Console.WriteLine(returnValues[(byte)100]);
                   }
                   else
                   {
                          // show the error message (always param code 1)
                          Console.WriteLine(returnValues[(byte)1]);
                    }

                    break;
            }

    }
}
```