Networking custom classes & structs

We are working on adding full auto serialization! For now, you can follow this guide.

Networking custom classes and structs can be extremely useful to send over the network, for example by using basic RPC's. There are currently 3 ways of preparing data to be networked. The managed option is recommended for ease of use.

Managed

This is the current closest thing we have to full auto serialization. All it requires is the "partial" keyword before your class or struct, and to inherit from the interface **IAutoNetworkData.** This can handle any values that also have pointers, for example network identity references, and those will be networked just fine as well.

```
public partial struct ManagedStruct : IAutoNetworkedData
{
    public NetworkIdentity myIdentity;
    public string someValue;
    public List<int> myIntList = new();
}
```

Unmanaged

This is essentially just a struct (and can only be a struct), however, it can't take anything that has a pointer. So for example this can't hold: **lists**, **strings**, **arrays** and **classes**.

```
public struct UnmanagedStruct
{
  public Vector3 position;
  public Quaternion rotation;
}
```

Custom

This allows you to handle your serialization yourself. An example of using this below:

```
public partial struct ManagedStruct : INetworkedData
{
    public bool hasValue;
    public string someValue;

    public void Serialize(NetworkStream packer)
    {
        packer.Serialize(ref hasValue);

        if (hasValue)
            packer.Serialize(ref someValue);
    }
}
```

Previous
Setting up a server

Next
Systems and modules

Last updated 15 days ago