

Date: 2019/08/31

FM Network UDP (notes)

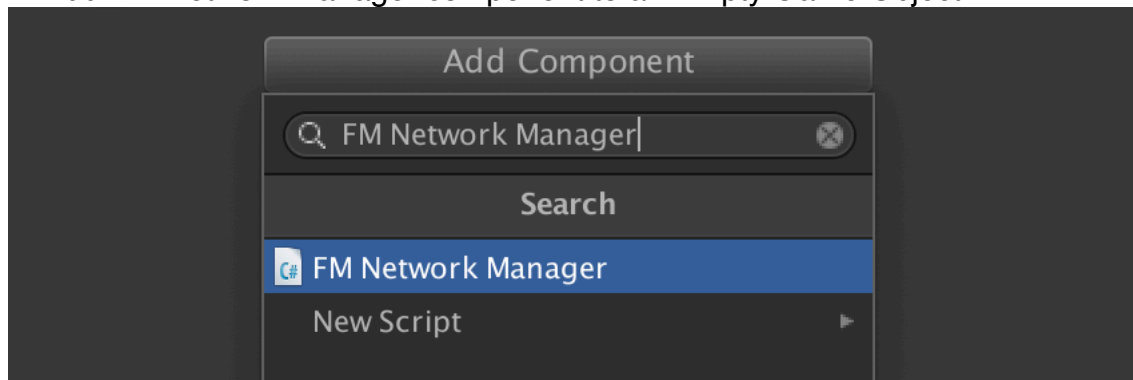
Thanks for using this tool pack! Below is a simple description for core scripts.

Email support: thelghome@gmail.com

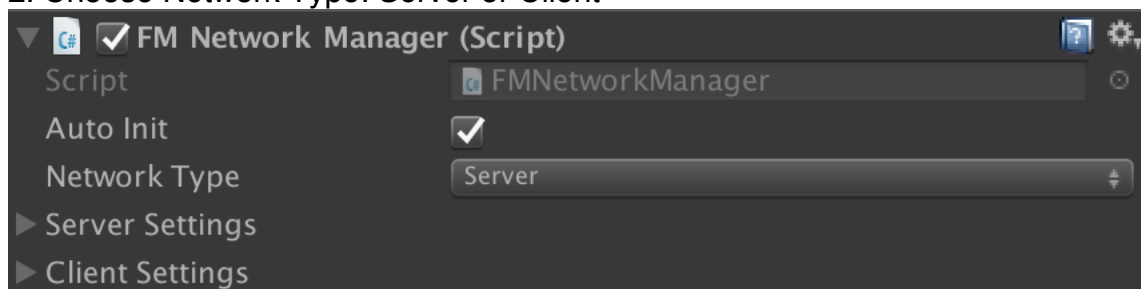
Website: <http://frozenmist.com>

Basic Setup

1. Add FM Network Manager component to an Empty Game Object



2. Choose Network Type: Server or Client

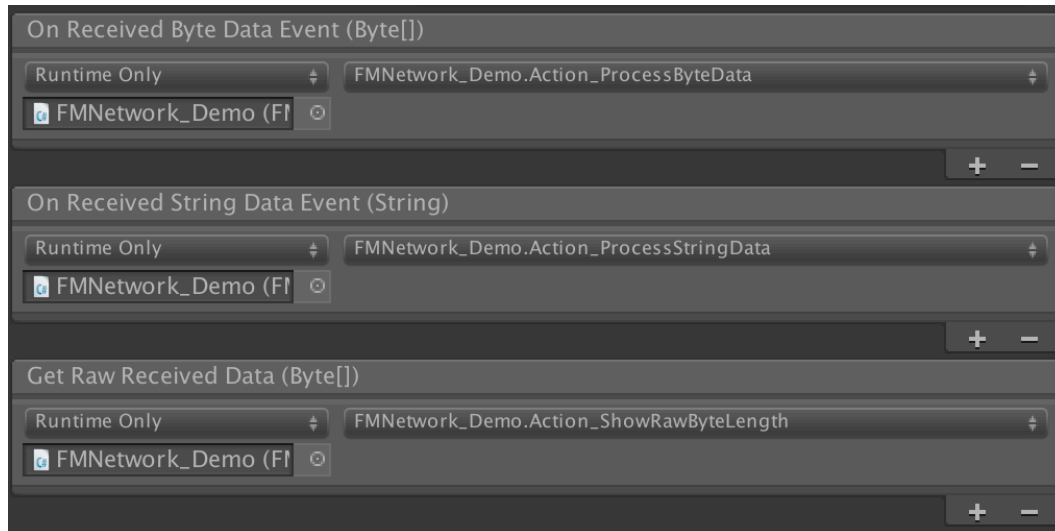


3. Example functions for sending Data: Send to (All / Others / Server)

```
string Message = "Hello World!";
FMNetworkManager.instance.SendToAll(Message);
FMNetworkManager.instance.SendToOthers(Message);
FMNetworkManager.instance.SendToServer(Message);

byte[] ByteData = new byte[1234];
FMNetworkManager.instance.SendToAll(ByteData);
FMNetworkManager.instance.SendToOthers(ByteData);
FMNetworkManager.instance.SendToServer(ByteData);
```

4. Events will be invoked when received data

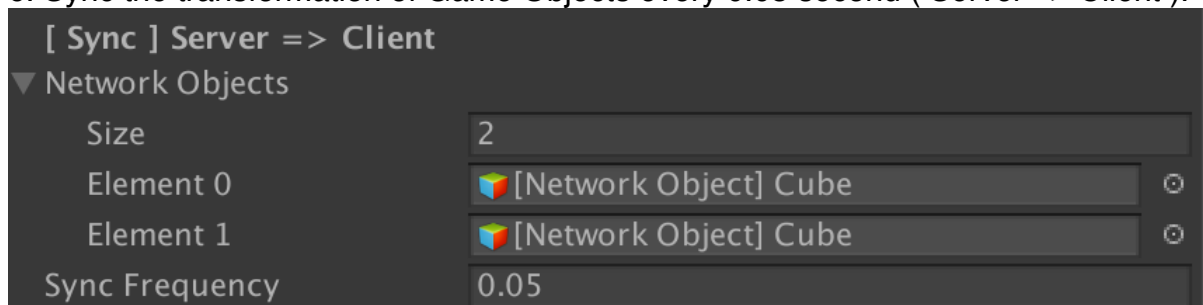


5. Example functions of processing received data

```
public void Action_ProcessStringData(string _string)
{
    if(FMNetworkManager.instance.NetworkType == FMNetworkType.Server)
    {
        if (ServerText != null) ServerText.text = "Server Received : " + _string;
    }
    else
    {
        if (ClientText != null) ClientText.text = "Client Received : " + _string;
    }
}

public void Action_ProcessByteData(byte[] _byte)
{
    if (FMNetworkManager.instance.NetworkType == FMNetworkType.Server)
    {
        if (ServerText != null) ServerText.text = "Server Received byte[] : " + _byte.Length;
    }
    else
    {
        if (ClientText != null) ClientText.text = "Client Received byte[]: " + _byte.Length;
    }
}
```

6. Sync the transformation of Game Objects every 0.05 second (Server => Client).



Information of all settings

Auto Init (bool)

- Init as Server/Client on Start()

Network Type (FMNetworkType)

- Network Status

Server Settings: Server Listen Port (int)

- port for listening any incoming data

Server Settings: Use Async Listener (bool)

- use Async Receive, otherwise will create a thread for listening any incoming data

Server Settings: Use Main Thread Sender (bool)

- use main thread to send data, otherwise will create a thread for sending data

Server Settings: Connection Count (int)

- number of connected clients

Client Settings: Client Listen Port (int)

- port for listening any incoming data

Client Settings: Use Main Thread Sender (bool)

- use main thread to send data, otherwise will create a thread for sending data

Client Settings: Auto Network Discovery (bool)

- will try to find server IP in local network

Client Settings: Server IP (string)

- Need to set the Server IP manually, when "Auto Network Discovery" is disabled

Client Settings: Is Connected (bool)

- Client Connection Status

OnReceivedByteDataEvent (Event: Byte[])

- invoke when received byte[], will return the byte[] without FM Network meta data

OnReceivedStringDataEvent (Event: String)

- invoke when received string, will return the string without FM Network meta data

GetRawReceivedData (Event: byte[])

- invoke when received byte[], will return all byte[]

Network Objects (GameObject[])

- will sync the transformation of game objects