

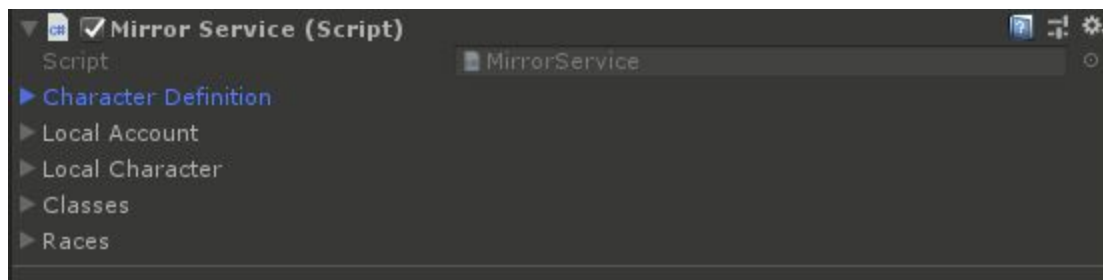
## Multiplayer Service

There is no Multiplayer Service shipped with base CAT, but yet CAT does support Multiplayer games. It does this through a common interface that can be used to integrate Multiplayer systems into CAT. One such integration is Mirror. Mirror (<https://github.com/vis2k/Mirror>) is a descendant of the previously built in Unity Multiplayer system. The CAT Mirror Multiplayer integration and Mirror can be downloaded from here:

<https://github.com/trickyfast/CATAddons/tree/master/CATMirrorMultiplayer>

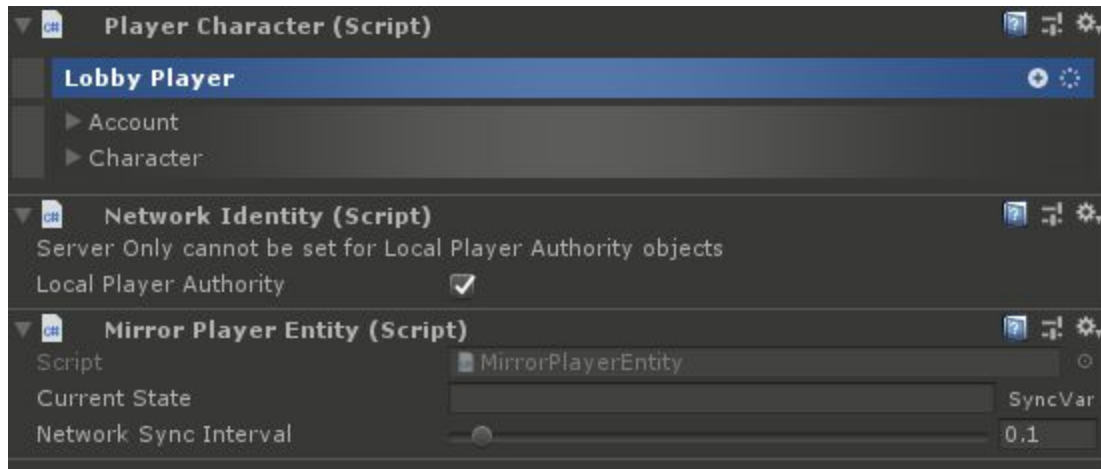
Note that Mirror requires Unity 2018.3 or greater.

Each Multiplayer implementation in CAT has its own service. For Mirror, this is the Mirror Service. It also implements a Player Service, so make sure to remove any existing Player Service before adding the Mirror Service.

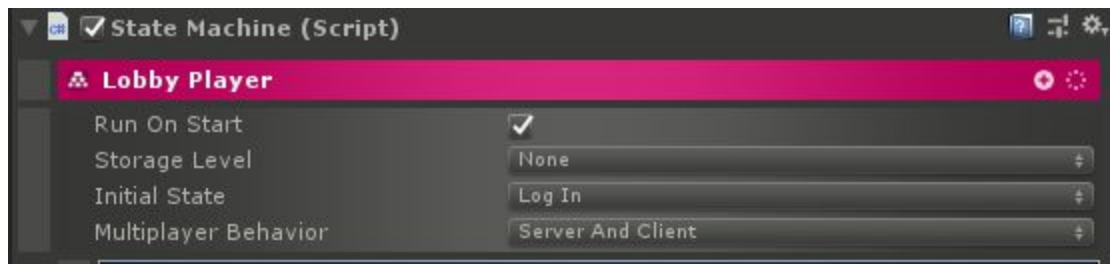


The parameters listed there are the same as for a Player Service. However, other components are added automatically to the Service's Game Object that are specific to Mirror including a CAT Network Manager, Telepathy Transport, Network Identity, and Mirror Multiplayer Entity. These have other parameters that can be set. For more information on those, please see the official Mirror Documentation: <https://mirror-networking.com/docs/>

Once the Service is added and configured, a Player Character should be created. This will be each Players' representation in the game. It needs a Player Character component and a Mirror Player Entity component.

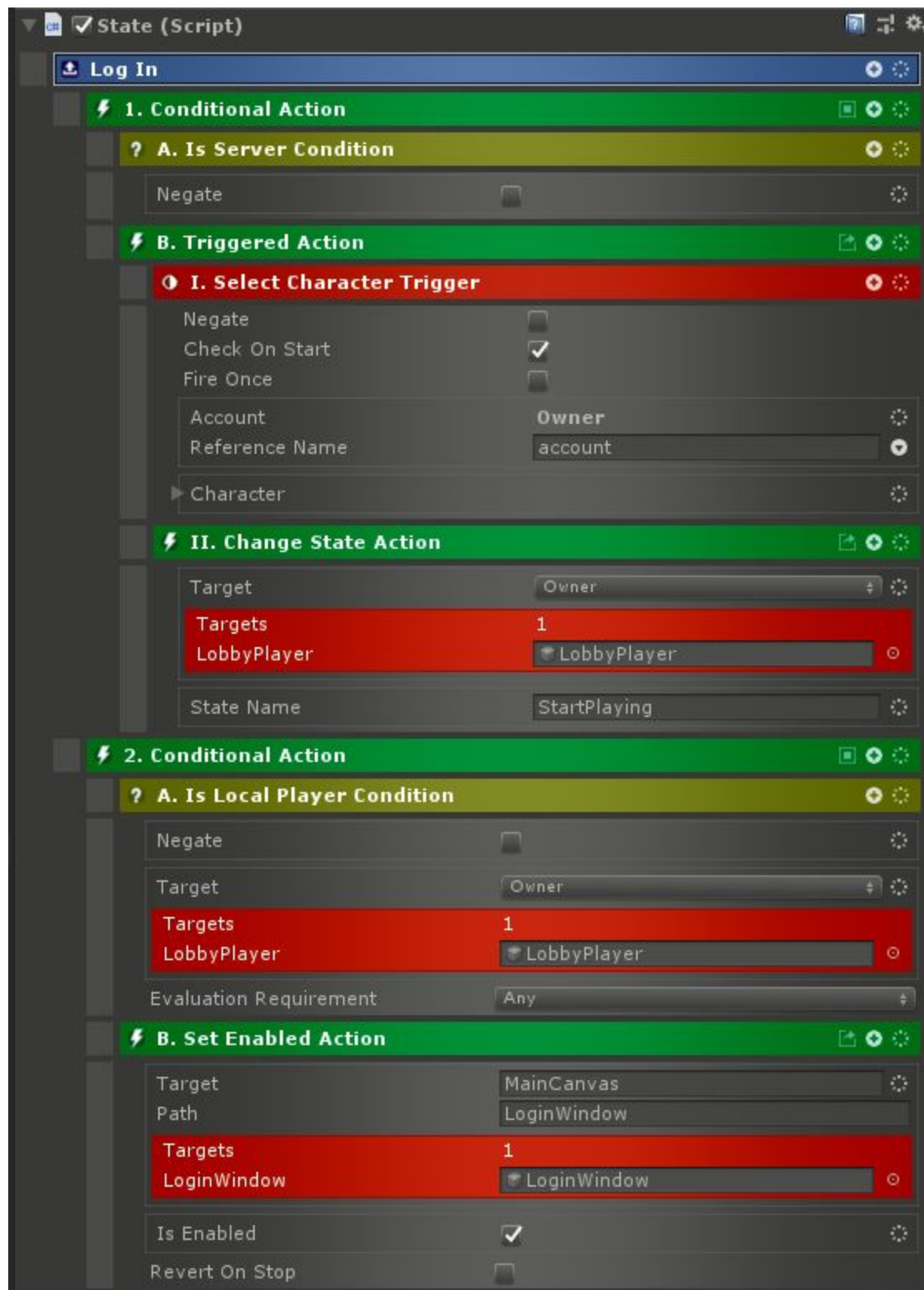


Note that the Network Identity script will be added automatically. It is required to select “Local Player Authority” so that each Player can control their own Character. Typically the Player Character will also have a State Machine component on it and will have whatever States are needed and Actions to allow the Player to control it. On all State Machines there is now an additional parameter called Multiplayer Behavior:



This parameter can be set to Server And Client (as above) which means that the State Machine will run normally on both the Server and Client. Server Only-- it will only run on the Server. Client Only-- it will only run on the client. Local Player Only-- it will only run on the Local Player’s game, or Server Determines State, which means that the State Machine will run on the Server, but the Client will also run. In this case, though, the Server determines what State the Client is in at any time.

Especially for Player Characters, it is useful to execute some Actions only on the Server and others only on the Client. This can be done with the Is Server Condition or Is Local Player Condition. Is Server Condition will cause Actions to only run on the Server and Is Local Player Condition will cause them to only run on the Local Player’s Client.

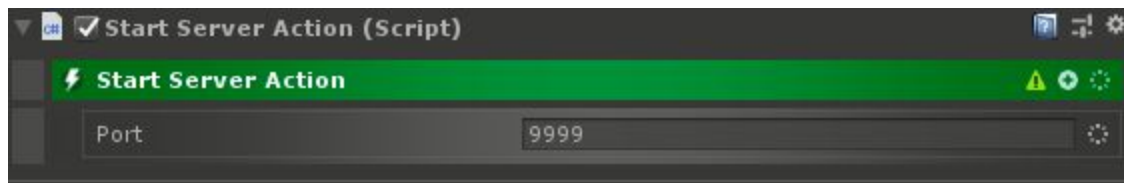


For example, the above CATs will on the Server wait for the Player to select a Character and then change to the Start Playing State. On the Player's Local Client, it will enable a Login Window in the UI so they can log in.

To note, logging in is not required for the Multiplayer Service. It is entirely optional. See the Mirror Multiplayer FPS example for how to set up login and Character selection.

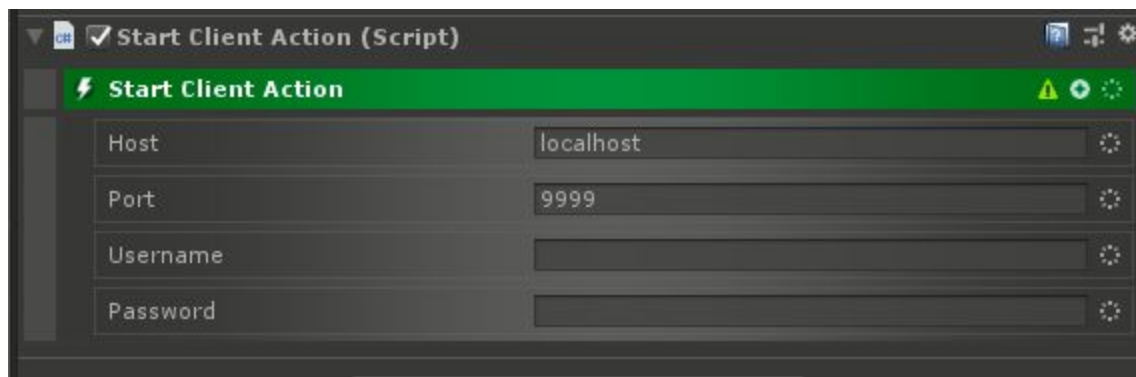
Once the Player is set up, it should be set in the CAT Network Manager on the Mirror Service as the Player Prefab. For any NPCs or other objects that should have their state synced across the network, use the Mirror Multiplayer Entity component. Put this on any State Machine that should be synchronized across players. For Mirror, the Network Transform component should also be used if the NPC or other object should have its position and rotation synchronized as well.

To initiate a Server, use the Start Server Action:



It only takes one parameter which is the port to run the server on. This is however, not used in Mirror.

To connect to a Server, use the Start Client Action.



Note that in Mirror, only the Host parameter is used. This should be set to the ip or DNS name of the server to connect to.

When using Multiplayer, some systems must be set up differently. For the Virtual Currency System, it is required to use a Virtual Currency Manager component on the Player and to specify the Player on all Virtual Currency CATs. Similarly with the Quest System, it is required to use a Quest Manager component on the Player and to specify the Player on all Quest CATs that allow it such as Grant Quest Action. This will ensure each Player has their own unique state in these systems.