Setting Up a MasterServer- Part 3- Writing a custom state serializer (Doing something over the network)

**Writing a custom state serializer** **While initially a game might simply serialize Transform or Rigidbody for testing, eventually it is often necessary to write a custom serialization function. This is a surprisingly easy task.**  **Tip** **Downloading the example code** **You can download the example code files for all Packt books you have pur- chased from your account at http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub.com/support and register to have the files e-mailed directly to you.**  **Here is a script that sends an object's position over the network:**

**using UnityEngine;**

**using System.Collections;**

**public class ExampleUnityNetworkSerializePosition : MonoBehaviour {**

**public void OnSerializeNetworkView( BitStream stream, NetworkMes- sageInfo info )**

**{**

**// we are currently writing information to the network**

**if( stream.isWriting )**

**{       // send the object's position**

**Vector3 position = transform.position;**

**stream.Serialize( ref position );     }**

**// we are currently reading information from the network**

**else     {**

**// read the first vector3 and store it in 'position'**

**Vector3 position = Vector3.zero;**

**stream.Serialize( ref position );**

**// set the object's position to the value we were sent**

**transform.position = position;     }   } }**

**Most of the work is done with BitStream. This is used to check if NetworkView is currently writing the state, or if it is reading the state from the network. Depending on whether it is reading or writing, stream.Serialize behaves differently. If Net- workView is writing, the value will be sent over the network. However, if Net- workView is reading, the value will be read from the network and saved in the refer- enced variable (thus the ref keyword, which passes Vector3 by reference rather than value).**