

Forum:[Hack Forums](#)

Thread: [Python's "struct.pack_into" in C#?](#)

.eXe - August 9, 2023, 4:41 am

Hey. I need help implementing some python code in C#. I'm looking for an equivalent to pack_into (not pack), that I can implement in to a C# project. See example Python code below:

```
Code    while fileSize > 0:
        chunkIndex += 1 # Increment chunk
        struct.pack_into("<H", packet, 18, chunkIndex) # Poke in chunk index
        struct.pack_into("<H", packet, 2, checksum + chunkIndex) # Checksum

        sock.sendto(packet, addr)
        data = sock.recvfrom(0x400)[0]

        if not buffer: # First chunk, extract file size
            fileSize = struct.unpack_from("<I", data, 20)[0]
            buffer = BytesIO()

        # Append chunk data to return buffer
        buffer.write(data[24:])
        fileSize -= 0x300

        # Be kind, rewind
        if buffer: buffer.seek(0)
        return buffer
```

Has anyone got any experience with something like this?

Managed to fix it with the following:

```
Code while (fileSize > 0)
    {
        chunkIndex++;
        BitConverter.GetBytes(chunkIndex).CopyTo(packet2, 18); // Poke in chunk index
        BitConverter.GetBytes((ushort)(checksum + chunkIndex)).CopyTo(packet2, 2); //
```

Checksum

```
//sock.Send(packet, packet.Length, addr);
sock.SendTo(packet2, packet2.Length, SocketFlags.None, addr);
Console.WriteLine("SENDING: " + Encoding.UTF8.GetString(packet2));
//var data = sock.Receive(ref addr);
EndPoint sender = new EndPoint(IPAddress.Parse(IP), 9520);
EndPoint remote = (EndPoint)sender;

int v = sock.ReceiveFrom(data, ref remote);
Console.WriteLine("RECEIVING: ");
if (buffer == null)
{
    fileSize = BitConverter.ToInt32(data, 20);
    buffer = new MemoryStream();
```

```
}

int lastIndex = Array.FindLastIndex(data, b => b != 0);

if (lastIndex >= 24) //Dirty - Removes NUL bytes from the UDP Stream.
{
    buffer.Write(data, 24, lastIndex - 23);
} else {
    buffer.Write(data, 24, data.Length - 24);
}
Array.Clear(data); //We clear the data array on each loop to ensure we don't write the
same data twice, if the data received is smaller than the size of the array.
fileSize -= 0x300;
}
```