

```

using System;
using System.IO;
using System.Xml;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace JSOToXML
{
    class MyJsonRoot
    {
        public JArray data;
    }

    class MyJsonTree
    {
        public MyJsonRoot root;
    }

    class Program
    {
        static void Main(string[] args)
        {
            if (args.Length > 1)
            {
                // Read the JSON file.
                string json = File.ReadAllText(args[0]);

                // Create the JSON object structure.
                MyJsonTree tree = new MyJsonTree()
                {
                    root = new MyJsonRoot()
                    {
                        data = JsonConvert.DeserializeObject<JArray>(json)
                    }
                };

                // Adjust the JSON object structure.
                foreach (JObject obj in tree.root.data)
                {
                    foreach (JToken child in obj.Children<JToken>())
                    {
                        foreach (JToken value in child.Values<JToken>())
                        {
                            if (value.Type == JTokenType.Array)
                            {
                                string newValue = string.Empty;
                                foreach (JValue item in value.Values<JValue>())
                                {
                                    if (newValue != string.Empty)
                                    {
                                        newValue += ", ";
                                    }
                                    newValue += item.Value.ToString();
                                }
                                value.Replace(new JValue(newValue));
                            }
                        }
                    }
                }

                // Convert JSON to XML.
                XmlDocument doc = JsonConvert.DeserializeXmlNode(JsonConvert.SerializeObject(tree));
                doc.Save(args[1]);
            }
        }
    }
}

```

}
}
}