

Copy object values in Visual Studio debug mode

Asked 9 years, 3 months ago Active 2 months ago Viewed 54k times



In Visual Studio debug mode it's possible to hover over variables to show their value and then right-click to "Copy", "Copy Expression" or "Copy Value".

113



In case the variable is an object and not just a basic type, there's a + sign to expand and explore the object. Is there a way to copy all that into the clipboard?



visual-studio

visual-studio-2010

debugging

29

asked May 28 '10 at 9:46



Farinha

8,288

20

60

76

11 Answers



In the immediate window, type

164

?name_of_variable



This will print out *everything*, and you can manually copy that anywhere you want, or use the immediate window's logging features to automatically write it to a file.



UPDATE: I assume you were asking how to copy/paste the nested structure of the values so that you could either search it textually, or so that you can save it on the side and then later compare the object's state to it. If I'm right, you might want to check out the commercial extension to Visual Studio that I created, called [OzCode](#), which lets you do these things much more easily through the "Search" and "Compare" features.

UPDATE 2 To answer @ppumkin's question, or [new EAP](#) has a new Export feature that allows users to Export the variable values to Json, XML, Excel, or C# code.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

edited Oct 16 '18 at 18:15



OmegaMan

17.4k 5 51 69

answered May 31 '10 at 9:19



Omer Raviv

8,151 4 37 79

-
- 4 Would have been even better if it could recursively expand to show everything. It could be lots of text but I would then be possibly to search it for a value deeper in the tree. – [Kobus Smit](#) Mar 17 '11 at 11:42
-
- 3 PS I am so loving the features of bugaid. downloading trail now and when it expires and i start gauging my eyeballs out will make my boss buy it for me.. :) – [Piotr Kula](#) Oct 25 '12 at 9:35
-
- 12 Anyone know how to output more than 100 items? I get this after the first 100 have been displayed in the Immediate window: < More... (The first 100 of 335 items were displayed.) > – [lachs](#) Jan 19 '16 at 5:50
-
- 3 What exactly is meant by "the immediate window"? – [alex](#) Feb 12 '16 at 18:14
-
- 3 @alex: in Visual Studio, go to Debug > Windows > Immediate - It allows you to enter expressions to be evaluated or executed during debugging – [JTech](#) Jan 9 '17 at 3:45
-

You can run below code in immediate window and it will export to an xml file the serialized XML representation of an object:

38

```
(new System.Xml.Serialization.XmlSerializer(obj.GetType())).Serialize(new
System.IO.StreamWriter(@"c:\temp\text.xml"), obj)
```

Source: [Visual Studio how to serialize object from debugger](#)

edited May 23 '17 at 12:02



Community ♦

1 1

answered Dec 16 '16 at 2:31



Bat_Programmer

4,553 7 44 65

-
- 5 This should be upvoted so people can find it more easily. I waded through a lot of similar SO questions before I found this great solution, which allowed me to dump a large List<List<byte>> to an xml text file after the crash of a long-running program with a rare crash problem. Just one addition: when I first ran it I got an error: "Evaluation requires a thread to run temporarily. Use the Watch window to perform the evaluation." After refreshing the watch window, a second attempt to run the command was successful. – [Craig.Feied](#) Jan 16 '17 at 23:22
-

Bat_Programmer: you might want to change the first line of your answer, because it suggests that what will be output is a list of objects, whereas what is really being output is a serialized XML representation of a single specified object (variable). – [Craig.Feied](#) Jan 16 '17 at 23:27 ✎

@Craig.Feied Thanks I have edited my answer. – [Bat_Programmer](#) Jan 17 '17 at 0:40

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Thanks, This was actually the most useful answer for me too. – [Chris Rogers](#) Jun 27 at 3:19



25



There is a recent extension Object Exporter that does this conveniently.

<http://www.omarelabd.net/exporting-objects-from-the-visual-studio-debugger/>

Extension: <https://visualstudiogallery.msdn.microsoft.com/c6a21c68-f815-4895-999f-cd0885d8774f>

edited Apr 16 '15 at 2:58



[Omar Elabd](#)

1,046 13 21

answered Feb 26 '15 at 11:06



[Animesh](#)

2,945 9 58 98

1 Seems to be giving problems for larger list of objects – [Bat_Programmer](#) Dec 16 '16 at 1:02

@Bat_Programmer Not to in any way discourage from using Omar's wonderful tool, please note that the OzCode Export feature has been fine-tuned to support huge lists of objects. Please try it and let me know if it works well for you. – [Omer Raviv](#) Jan 9 '17 at 18:28



18



Not exactly what youre asking for but you can add a watch for that object, and in the watch window, expand and select everything you want to copy and then copy it.

answered May 28 '10 at 9:52



[PMN](#)

1,407 10 9

1 That does the job. It's not a pretty output, but it works. – [Farinha](#) May 28 '10 at 10:11

1 Thanks PMN that helped me as well. By manually expanding the tree (would be nice if there were a right click, expand all) and then copying it to the clipboard, I can then paste it and search for a value I'm looking deep down in the tree. – [Kobus Smit](#) Mar 17 '11 at 11:44



By using [attributes to decorate your classes and methods](#) you can have a specific value from your object display during debugging with the [DebuggerDisplay attribute](#) e.g.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
{  
    public string Name { get; set; }  
    public int Age { get; set; }  
}
```

answered May 28 '10 at 10:06

[Dave Anderson](#)**9,332** 2 45 71

The thing is, I need to be able to copy all the object contents, and not just some of them. Yes, I could build a complex enough DebuggerDisplay attribute with the whole object, but any changes to the class would need a change in the attribute as well. – [Farinha](#) May 28 '10 at 10:11

@Farinha What do you want to do with the object? Perhaps could have a property that is the serialized xml version of it. – [Dave Anderson](#) May 28 '10 at 11:18

Most popular answer from <https://stackoverflow.com/a/23362097/2680660>:

4

With any luck you have Json.Net in you appdomain already. In which case pop this into your Immediate window:

```
Newtonsoft.Json.JsonConvert.SerializeObject(someVariable)
```

edited Jun 21 at 18:11

answered Jun 3 at 18:06

[Efreto](#)**768** 8 16

Google led me to this 8-year-old question and I ended up using [ObjectDumper](#) to achieve something very similar to copy-pasting debugger data. It was a breeze.

1

I know the question asked specifically about information from the *debugger*, but ObjectDumper gives information that is basically the same. I'm assuming those who google this question are like me and just need the data for debugging purposes and don't care whether it technically comes from the debugger or not.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



I always use:

1 `string myJsonString = JsonConvert.SerializeObject(<some object>);`

Then I copy the string value which unfortunately also copies the back slashes.

To remove the backslashes go here: https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_replace

Then within the `<p id="demo">Visit Microsoft!</p>` element replace the text with the text you copied. then replace the `var res = str.replace("Microsoft", "W3Schools");` line with

```
var res = str.replace(/\\/g, '')
```

Run these new changes but don't forget to click the "try it" button on the right.

Now you should have all the text of the object in json format that you can drop in a json formatter like <http://jsonformatter.org> or to create a POCO you can now use <http://json2csharp.com/>

answered Aug 6 '18 at 16:38



[Helzgate](#)

5,944 2 29 37

I know I'm a bit late to the party, but I wrote a JSON implementation for serializing an object, if you prefer to have JSON output. Uses Newtonsoft.Json reference.

1

```
private static void WriteDebugJSON (dynamic obj, string filePath)
{
    using (StreamWriter d = new StreamWriter(filePath))
    {
        d.Write(JsonConvert.SerializeObject(obj));
    }
}
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

answered Oct 11 '18 at 16:30

[Marcus Parsons](#)

719 3 13



if you have a list and you want to find a specific variable: In the immediate window, type

0

```
myList.Any(s => s.ID == 5062);
```



if this returns true

```
var myDebugVar = myList.FirstOrDefault(s => s.ID == 5062);  
?myDebugVar
```

answered Dec 4 '18 at 7:11

[emert117](#)

413 2 6 17



I've just right clicked on the variable and selected *Add Watch*, that's bring up watch window that consists of all the values. I selected all and paste it in a text a text editor, that's all.

0



answered Apr 24 at 6:13

[agileDev](#)

69 9

the problem is to copy the object that way you need to expand node by node and I do have 145 nodes – [derloopkat](#) Aug 8 at 9:35

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).