

## Debugger. Data Views

Use this page to manage the way the data is displayed in the Java debugger.

| Item                              | Description  |
|-----------------------------------|--|
| Sort alphabetically               | Show the nodes in the alphabetical order.  |
| Autoscroll to new local variables | Automatically scroll to new variables that appear in the scope when stepping.  |
| Auto tooltips for values          | <p>Select or clear the check box to enable or disable showing tooltips for values.</p> <p>A tooltip in this context means a pop-up that provides an alternative, sometimes more convenient presentation of values in the <b>Variables</b> pane.</p> <p>To illustrate, let's assume that there is a statement like this in your code:</p> <pre>String s = "Hello, World!            \n            Hello, World!";</pre> <p>When this statement is executed in the debugger, you'll see a line looking similar to this in the <b>Variables</b> pane:</p> <pre>s = {java.lang.String@62}"Hello, World!    \n    Hello, World!"</pre> <p>with the line break shown as <code>\n</code>.</p> <p>If the <b>Auto tooltips for values</b> option is on and you click this line and then hold the mouse pointer on it, you'll see a yellow area (the "tooltip") in which the value of <code>s</code> is shown as</p> <pre>Hello, World! Hello, World!</pre> <p>with a real line break in place of <code>\n</code>.</p> <p>If the option is off, to show or hide such a tooltip, you should press the <b>Alt</b> key.</p> |
| Value tooltips delay              | Specify the delay in milliseconds between the moment when the mouse pointer hovers over an object in the <b>Variables</b> pane, and the moment when a tooltip with the object's value is displayed.  |
| Show declared type                | Enable showing declared type.  |
| Show synthetic fields             | Enable showing synthetic fields (automatically generated by the Java compiler).  |
| Show object id                    | Enable showing object IDs.   |
| Show static fields                | Enable showing static fields.  |

| Item  | Description   |
|---|---|
| Show static final fields                        | Enable showing static final fields.   |
| Array start/end index                           | Define the range of array elements to be shown.   |
| Show maximum <number> array elements            | <p>Define the maximum number of array elements to be shown.</p> <p>This and the previous setting depend on one another. For example, if the range is 0-100 and the maximum number of elements is 50, IntelliJ IDEA will show elements 0 to 25 and 75 to 99 omitting those in between.</p>   |
| Hide null array elements                        | If selected, the null array elements are omitted.   |
| Enable auto expressions in Variables view       | <p>The auto expressions mentioned in this option work like this:</p> <p>The debugger analyzes the context near the breakpoint (the current statement, and one statement before and one after). It does so to find various expressions in the source code (if available) such as, for example, <code>myvar.myfield</code>.</p> <p>If such expressions don't contain explicit method invocations, the debugger evaluates the expressions and adds the corresponding values to the debugger tree view (shown in the <b>Variables</b> pane) assuming that these values may be of interest.</p> <p>In languages such as Groovy, just by looking at an expression, it's not possible to conclude whether any methods are invoked when evaluating the expression. And such method invocations often cause unwanted side effects. So you may want to turn this option off when debugging Groovy code.</p> |
| Enable alternative view for Collection classes  | Present Collections and Maps in a more convenient form.   |
| Enable toString() object view                   | In this group you can select classes if you need them and their descendants to be presented as a result of the <code>toString()</code> method call while debugging.   |
| For all classes that override toString() method | Show all classes as <code>toString()</code> .   |
| For classes from the list                       | Define the list of classes to be shown as <code>toString()</code> , using the Add Class, Add Pattern and the Remove buttons. Use the check boxes next to the class names to temporarily enable or disable particular filters.   |
| Add class                                       | Add a class to the list using the <b>Choose Class</b> dialog.   |
| Add pattern                                     | Add a custom class filter using the <b>New Filter</b> dialog. To define a filter, enter a string pattern e.g., <code>*.Test, javax.swing.*</code> , etc.  |

| Item   | Description                            |
|--------|--|
| Remove | Remove a file or filter from the list. |

## See Also

### Concepts:

- [Running, Debugging and Testing](#)

### Procedures:

- [Debugging](#)

### Reference:

- [Debug Tool Window](#)

### Web Resources:

- [Developer Community](#) 