

# Customize Data Views

## Debug tool window | Variables tab | context menu | Customize Data Views

Use this dialog to configure the debugger data view options. Note that the same settings can be alternatively configured on the [Debugger page](#) in the **Settings** dialog.

- [Data Views tab](#)
- [Data Type Renderers tab](#)

### Data Views tab

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.

Item	Description
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.
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> .

Item	Description
For classes from the list	Define the list of classes to be shown as toString(), 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., *.Test, javax.swing.*, etc.
Remove	Remove a file or filter from the list.

### Data Type Renderers tab

On this tab, you can create and customize rendering schemes for data presentation in the debugger Frame view.

If no rendering scheme is currently defined, start by clicking **+**.

Item	Description
<b>+</b> Add	Add a new rendering scheme to the list.
<b>-</b> Remove	Remove the selected scheme from the list.
 Copy	Clone the selected scheme.
 Move Up  Move Down	Move the selected item up or down in the list. The order of items in the list defines which renderer is used when class inheritance is ambiguous.
Renderer name	Edit or view name of the selected renderer.
Apply renderer to objects of type	Define the type of objects to be affected by the renderer. Type the fully qualified name of the object, or click the ellipsis button and choose the desired type from the list in the Renderer Reference Type dialog.

Item	Description
When rendering a node	<p data-bbox="379 165 1257 197">This option determines how the object is displayed in the debugger:</p> <ul data-bbox="386 226 1390 409" style="list-style-type: none"><li data-bbox="386 226 1390 293">■ <b>Use default renderer:</b> Select this option to display the node in a default IntelliJ IDEA way.</li><li data-bbox="386 315 1390 409">■ <b>Use following expression:</b> Type in the java expression you want to use to identify the object. You can use object's properties, constants, and even string math as part of your renderer.</li></ul> <div data-bbox="430 432 1398 633" style="border: 1px solid #ccc; padding: 10px;"><p data-bbox="456 450 1382 589">When defining expressions, you can use IntelliJ IDEA's code-completion features to help you (Ctrl+Space). All method calls and member variable access is relative to the object you're rendering. Use <code>this</code> to refer to the instance, to which the renderer is applied.</p></div> <div data-bbox="430 656 1398 846" style="border: 1px solid #f96; padding: 10px;"><ul data-bbox="456 674 1382 819" style="list-style-type: none"><li data-bbox="456 674 1382 741">■ Using <i>heavy</i> expressions in renderers may slow down data rendering in views.</li><li data-bbox="456 763 1382 819">■ Method calls should be used with caution because of possible side-effects.</li></ul></div>

Item	Description
When expanding a node	<p>Define how the children information is presented.</p> <p>Normally, expanding a node in the debugger lists the object's member variables (using renderer appropriate for their object types). This option lets you override that behavior and select a single expression or a series of expressions to control the display. You may use this to limit the amount of information displayed or to be more precise in how the information is presented for example.</p> <ul style="list-style-type: none"> <li>■ <b>Use default renderer:</b> information for the selected node children is presented in a default way.</li> <li>■ <b>Use expression:</b> enter a Java expression to calculate information to be presented for the node.</li> </ul> <p><b>Test if the node can be expanded (optional):</b> enter a Boolean expression. If it is <code>true</code> the renderer displays the expandable nodes for the defined objects. Otherwise, no nodes are displayed.</p> <ul style="list-style-type: none"> <li>■ <b>Use the list of expressions:</b> create a list of separate expressions to be calculated and presented as node children.</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>When defining expressions, you can use IntelliJ IDEA's code-completion features to help you (<code>Ctrl+Space</code>). All method calls and member variable access is relative to the object you're rendering.</p> </div>
Add / Remove	Use these buttons to add and remove the expressions.
Move Up / Move Down	Use these buttons to change the order of items in the list.

## See Also

### Procedures:

- [Customizing Views](#)

### Reference:

- [Debugger](#)

### Web Resources:

- [Developer Community](#) 