Debugger. Data Type Renderers

IntelliJ IDEA allows you to specify how different objects are displayed in the debugger on a class-by-class basis. You can assign expressions to display rather than rely on the object's String representation.

For example, if an object represents a user, you may want to see users represented by their login name; or, for a cache entry object, its age and contents may be appropriate. IntelliJ IDEA refers to these as type renderers.

> If no rendering scheme is defined, the dialog does not show any controls. To start working with renderers click **Add**.

Item	Description
+	Use this icon to add a new rendering scheme to the list.
−	Use this icon to remove the selected scheme from the list.
📋	Use this icon to create a copy of the selected scheme.
↑ and ↓	Use these icons to move the selected item one line up or down in the list. The order of items in the list determines which renderer is used in the case of ambiguity born of class inheritance.
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.
When rendering a node	This option determines how the object is displayed in the debugger: ▪ **Use default renderer**: Select this option to display the node in a default IntelliJ IDEA way. ▪ **Use following expression**: 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. When defining the expressions, you can use codecompletion (`Ctrl+Space`). All method calls and member variable access are relative to the object you're rendering. Use `this` to refer to the instance, to which the renderer is applied. ▪ Using *heavy* expressions in renderers may slow down data rendering in views. ▪ Method calls should be used with caution because of possible side-effects.

Item	Description
When expanding a node	Define how the children information is presented. 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. ■ **Use default renderer**: information for the selected node children is presented in a default way. ■ **Use expression**: enter a Java expression to calculate information to be presented for the node. **Test if the node can be expanded (optional)**: enter a Boolean expression. If it is `true` the renderer displays the expandable nodes for the defined objects. Otherwise, no nodes are displayed. ■ **Use the list of expressions**: create a list of separate expressions to be calculated and presented as node children. Use: + (`Alt+Insert`) to create a new expression. — (`Alt+Delete`) to remove the selected expression from the list. ⬆ (`Alt+Up`) to move the selected expression one line up in the list. ⬇ (`Alt+Down`) to move the selected expression one line down in the list. When defining the expressions, you can use code completion (`Ctrl+Space`). All method calls and member variable access are relative to the object you're rendering.

See Also

Concepts:

- Running, Debugging and Testing

Procedures:

- Debugging

Reference:

- Debug Tool Window

Web Resources:

- Developer Community