

# JavaScript migration from Nashorn to Rhino engine

The JavaScript *Nashorn* has been removed from the modeling tool ([learn more about deprecated JavaScript Nashorn](#)). Please use JavaScript *Rhino* instead.



To find the usages of the JavaScript *Nashorn* in the project, run the **Deprecated JavaScript** validation suite (**Analyze > Validation > Validate**).

The key differences are listed below. For more information, please check [Nashorn/Rhino Migration Guide](#).

## Class object and .class property

If a java API accepts a `java.lang.Class` object, in *Nashorn* ".class" property (similar to Java) is used. In *Rhino* you pass script representation of class "as is".

```
// Nashorn
var types = [com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Class.class]
// Rhino
var types = [com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Class]

var project = com.nomagic.magicdraw.core.Application.getInstance().getProject()
var Finder = com.nomagic.magicdraw.uml.Finder
Finder.byTypeRecursively().find(project, types, false)
```

## Accessing Java packages and classes from script

*Nashorn's* recommended way to access Java classes (by using `Java.type`) is not supported in *Rhino*.

```
// Nashorn
var Vector = Java.type("java.util.Vector")
var JFrame = Java.type("javax.swing.JFrame")

// Rhino
var Vector = java.util.Vector
var JFrame = Packages.javax.swing.JFrame
```

## Creating Java arrays from script

In *Nashorn*, you can resolve to a Java array class using the same `Java.type` API. And array creation is done using `new` operator. In *Rhino*, you create a Java array using Java reflection from script.

```
// Nashorn
var IntArray = Java.type("int[]")
var array = new IntArray(8)

// Rhino
var Array = java.lang.reflect.Array
var intClass = java.lang.Integer.TYPE
var array = Array.newInstance(intClass, 8)
```

## Java exceptions

In *Nashorn* Java exception objects are thrown "as is". *Rhino* wraps Java exceptions as a script object. If you want underlying Java exception, use "javaException" property to access it.

```
try
{
    <...>
}
catch (e)
{
    // Nashorn
    e.printStackTrace()
    // Rhino
    e.javaException.printStackTrace()
}
```