

References to elements

References to elements are implicitly inserted into the JavaScript context when calling `eval()`, `execute()`, or `call()`. Examples of implicit variables include, for instance, `$Class`, `$UseCase`, `$sorter`, etc.

For example:

A "functions.js"

```
importPackage(java.util)
// variable $Dependency and $sorter can be accessed directly inside
function getSupplier()
{
    var supplierList = new ArrayList();
    var sortedDependencyList = $sorter.sort($Dependency);
    for (var i=0; i<sortedDependencyList.size(); i++)
    {
        var dependency = sortedDependencyList.get(i);
        supplierList.addAll(dependency.getSupplier());
    }
    return supplierList;
}
```

A template code

```
#import ('js', 'com.nomagic.reportwizard.tools.script.JavaScriptTool')
$js.execute("functions.js")
#set ($supplierList = $js.eval("getSupplier()"))
#foreach ($supplier in $supplierList)
$supplier.name
#end
```

Example:

A "functions.js"

```
var qualifiedName = "";
function packageQualifiedName(element)
{
    qualifiedName = "";
    packageName(element);
    return qualifiedName=="?"?"Default":qualifiedName;
}
function packageName(element)
{
    var parent = element.owner;
    if (parent.classType.simpleName == 'Package') {
        if (qualifiedName == "")
            qualifiedName = parent.name;
        else
            qualifiedName = parent.name + '.' + qualifiedName;
        packageName(parent);
    }
}
```

A template code

```
#import ('js', 'com.nomagic.reportwizard.tools.script.JavaScriptTool')
$js.execute("functions.js")
#foreach ($c in $Class)
    Package: $js.eval('packageQualifiedName($c)')
    Name: $c.name
#end
```

Reference Documents

- Mozilla JavaScript Tool
http://developer.mozilla.org/en/Rhino_documentation
- Sun JavaScript programming guide
http://java.sun.com/javase/6/docs/technotes/guides/scripting/programmer_guide/index.html