

JavaScript Examples for VMware vCenter Orchestrator 5.1



Orchestrator implements the Mozilla Rhino 1.6 JavaScripting engine. The Orchestrator workflow engine allows you to use basic JavaScript language features, such as if, loops, arrays, and strings. You can cut, paste, and adapt the Orchestrator JavaScript examples to help you write JavaScripts for common orchestration tasks.

Configuring the vCO Server for Scripting

Before running scripts in vCO, you need to set some system properties of the vCO server by editing the vmo.properties and js-io-rights.conf files, located in :

- If you installed vCO with the VC installer - `install_directory\VMware\Infrastructure\Orchestrator\app-server\server\vm\conf`
- If you installed the standalone vCO - `install_directory\VMware\Orchestrator\app-server\server\vm\conf`.

To set JavaScript access to OS commands, add to vmo.properties the following line: `com.vmware.js.allow-local-process=true`

To set JavaScript access to Java classes

1. Create a configuration file and list the Java packages to which to allow JavaScript access.
2. Add to vmo.properties the following line : `com.vmware.scripting.rhino-class-shutter-file=path_to_your_configuration_file`

To set server file system access for JavaScript

1. Create `c:/orchestrator` folder at the root of the vCO server system.
2. Add the necessary lines to the `js-io-rights.conf` file to allow or deny access to parts of the file system. For example, to deny the execution rights in the `c:/orchestrator/noexec` directory, type `-x c:/orchestrator/noexec`.

Basic Scripting Examples

Access XML Documents From Javascript by using E4X

```
var people = <people>
  <person id="1">
    <name>Moe</name>
  </person>
  <person id="2">
    <name>Larry</name>
  </person>
</people>;
System.log("'people' = " + people);
System.log("'people' is of type : $ + typeof(people));
people.person.length () + "persons");
System.log("whose first element is : " + people.person[0]);
people.person[0].@id='47';
System.log("Moe's id is now : " + people.person.(name=='Moe').@id);
delete people.person[0];
System.log("Moe is now removed.");
people.person[1] = new XML("<person id='3'><name>James</name></person>");
System.log("Added James to the list, which is now :");
for each(var person in people..person)
for each(var person in people..person){
System.log("- " + person.name + " (id=" + person.@id + ")");
}
```

Setting and Obtaining Properties from a Hashtable

```
//key is a String and the value is an object, a number, a Boolean, or a String
var table = new Properties() ;
table.put("myKey",new Date()) ;
var myDate= table.get("myKey") ;
System.log("Date is : "+myDate) ;
```

Replace the Contents of a String

```
var str1 = "hello" ;
var reg = new RegExp("l", "g");
var str2 = str1.replace(reg,"\\");
System.log(""+str2) ; // result : '\\ello'
```

Run a Command in the Orchestrator Server

```
var cmd = new Command("ls -al") ;
cmd.execute(true) ;
System.log(cmd.output) ;
```

File System

Add Content to a Simple File

```
var tempDir = System.getTempDirectory() ;
var fileWriter = new FileWriter(tempDir + "/readme.txt") ;
fileWriter.open() ;
fileWriter.writeLine("File written at : "+new Date()) ;
fileWriter.writeLine("Another line") ;
fileWriter.close() ;
```

Obtain the Contents of a File

```
var tempDir = System.getTempDirectory() ;
var fileReader = new FileReader(tempDir + "/readme.txt") ;
fileReader.open() ;
var fileContentAsString = fileReader.readAll();
fileReader.close() ;
```

Networking

Obtain Text from a URL

- `var url = new URL("http://www.vmware.com") ;`
- `var htmlContentAsString = url.getContent() ;`

LDAP

Convert LDAP Objects to Active Directory Objects

```
var ldapGroup ;
var adGroup = ActiveDirectory.search("UserGroup",ldapGroup.commonName) ;
var ldapElement = Server.getLdapElement(adGroup.distinguishedName) ;
```

Logging

Persistent Logging

```
Server.log("This is a persistent message", "enter a long description here");
Server.warn("This is a persistent warning", "enter a long description here");
Server.error("This is a persistent error", "enter a long description here");
```

Non-Persistent Logging

```
System.log("This is a non-persistent log message");
System.warn("This is a non-persistent log warning");
System.error("This is a non-persistent log error");
```

Workflows

Return All Workflows Run by the Current User

```
var allTokens = Server.findAllForType('WorkflowToken');
var currentUser = Server.getCredential().username;
var res = [];
for(var i = 0; i<res.length; i++){
if(allTokens[i].runningUserName == currentUser){
res.push(allTokens[i]);
}
}
return res;
```

Access the Current Workflow

```
System.log("Current workflow run ID: " + workflow.id);
System.log("Current workflow run start date: "+workflow.startDate);
```

Schedule a Workflow

```
var workflowToLaunch = myWorkflow ;
var workflowParameters = new Properties() ;
workflowParameters.put("name","John Doe") ;
workflowParameters.put("__taskName","Workflow for John Doe") ;
var workflowScheduleDate = new Date() ;
var time = workflowScheduleDate.getTime() + (60*60*1000) ;
workflowScheduleDate.setTime(time) ; var scheduledTask =
workflowToLaunch.schedule(workflowParameters,workflowScheduleDate);
```

Run a Workflow on a Selection of Objects in Loop

```
var len=VMs.length;
for (var i=0; i < len; i++ ) {
var VM = VMs[i];
var workflowToLaunch = workflowToRun;
if (workflowToLaunch == null) {
throw "Workflow not found"; }
var workflowParameters = new Properties();
workflowParameters.put("vm",VM);
var wfToken = workflowToLaunch.execute(workflowParameters);
System.log ("Ran workflow on " +VM.name); }
```