# Table of Contents

Lab Overview - HOL-1921-05-CMP - vRealize Orchestrator - Getting Started .............. 2
  Lab Guidance .................................................................................................................. 3  
Module 1 - vRealize Orchestrator Overview (30 min) .................................................. 9 
  Introduction ...................................................................................................................... 10  
  Introduction to vRealize Orchestrator Client .............................................................. 11  
  Running your first workflow ......................................................................................... 16  
  Conclusion ....................................................................................................................... 25  
Module 2 - Creating a Basic Orchestrator Workflow (30 min) .................................... 27 
  Introduction ...................................................................................................................... 28  
  Creating your First Workflow ....................................................................................... 30  
  Understanding a Workflow Token ................................................................................. 44  
  Conclusion ....................................................................................................................... 48  
Module 3 - Introduction to Parameters, Attributes and Scripting Objects (30 min) ..... 50 
  Introduction ...................................................................................................................... 51  
  Understanding Input and Output Parameters ............................................................... 53  
  Understanding Workflow Attributes ............................................................................. 56  
  Using Scripting Objects ................................................................................................. 65  
  Conclusion ....................................................................................................................... 76  
Module 4 - Using Actions in vRealize Orchestrator (30 min) ....................................... 78 
  Introduction ...................................................................................................................... 79  
  Overview of Actions ....................................................................................................... 81  
  Create an Action ............................................................................................................. 83  
  Conclusion ....................................................................................................................... 94  
Module 5 - Creating and Managing Resource Elements (30 min) ............................... 96  
  Introduction ...................................................................................................................... 97  
  Creating Resource Elements ....................................................................................... 99  
  Managing Resource Elements .................................................................................... 107  
  Using Resource Elements ............................................................................................ 119  
  Conclusion ...................................................................................................................... 133  
Module 6 - Using the new Web UI to Monitor and Manage vRealize Orchestrator (30 min) .............................................................................................................................. 135  
  Introduction .................................................................................................................... 136  
  Use Web Client to Troubleshoot ................................................................................. 139  
  Use Web Client to Explore ........................................................................................... 146  
  Conclusion ...................................................................................................................... 157
Lab Overview -
HOL-1921-05-CMP -
vRealize Orchestrator -
Getting Started
Lab Guidance

Note: It may take more than 90 minutes to complete this lab. You should expect to only finish 2-3 of the modules during your time. The modules are independent of each other so you can start at the beginning of any module and proceed from there. You can use the Table of Contents to access any module of your choosing.

The Table of Contents can be accessed in the upper right-hand corner of the Lab Manual.

Use vRealize Orchestrator to simplify the automation of IT tasks. Explore the vRealize Orchestrator development environment and learn important development concepts by building and testing basic workflows, then put those workflows to use inside vRealize Automation to create reusable processes.

Lab Module List:

- **Module 1 - vRealize Orchestrator Overview** (30 min) (Intermediate) Examine and understand the key features and architecture of the vRealize Orchestrator platform. Learn the methods for accessing and interacting with platform.
- **Module 2 - Creating a Basic Orchestrator Workflow** (30 min) (Intermediate) Learn how to create a basic workflow to resolve a simple yet routine task such as cloning a virtual machine or taking a snapshot.
- **Module 3 - Introduction to Parameters, Attributes and Scripting Objects** (30 min) (Intermediate) Explore and learn about the objects and constructs that are used to extend the functionality of workflows. These include attributes, scripting objects and decision trees as well as the parameters they use.
- **Module 4 - Using Actions in vRealize Orchestrator** (30 min) (Advanced) Reuse your JavaScript code by turning JavaScript Functions into vRealize Orchestrator Actions, then use them repeatedly in multiple workflows. Actions represent individual functions that you use as building blocks in workflows and scripts.
- **Module 5 - Creating and Managing Resource Elements** (30 min) (Advanced) Discover how to use external objects such as scripts, XML, or HTML files in workflows.
- **Module 6 - Using the new Web UI to Monitor and Manage vRealize Orchestrator** (30 min) (Intermediate) Explore the all-new vRealize Orchestrator Web UI and the new capabilities it brings, including the web-based workflow viewer, troubleshooting, statistics, performance monitoring and more!

Lab Captains:

- **Module 1-6 - Benoit Serratrice, Staff Cloud Solutions Architect**
This lab manual can be downloaded from the Hands-on Labs Document site found here:

http://docs.hol.vmware.com

This lab may be available in other languages. To set your language preference and have a localized manual deployed with your lab, you may utilize this document to help guide you through the process:


**Location of the Main Console**

1. The area in the RED box contains the Main Console. The Lab Manual is on the tab to the Right of the Main Console.
2. A particular lab may have additional consoles found on separate tabs in the upper left. You will be directed to open another specific console if needed.
3. Your lab starts with 90 minutes on the timer. The lab can not be saved. All your work must be done during the lab session. But you can click the **EXTEND** to increase your time. If you are at a VMware event, you can extend your lab time twice, for up to 30 minutes. Each click gives you an additional 15 minutes. Outside of VMware events, you can extend your lab time up to 9 hours and 30 minutes. Each click gives you an additional hour.
Alternate Methods of Keyboard Data Entry

During this module, you will input text into the Main Console. Besides directly typing it in, there are two very helpful methods of entering data which make it easier to enter complex data.

Click and Drag Lab Manual Content Into Console Active Window

You can also click and drag text and Command Line Interface (CLI) commands directly from the Lab Manual into the active window in the Main Console.

Accessing the Online International Keyboard

You can also use the Online International Keyboard found in the Main Console.

1. Click on the Keyboard Icon found on the Windows Quick Launch Task Bar.
Click once in active console window

In this example, you will use the Online Keyboard to enter the "@" sign used in email addresses. The "@" sign is Shift-2 on US keyboard layouts.

1. Click once in the active console window.
2. Click on the **Shift** key.

Click on the @ key

1. Click on the "@ key".

Notice the @ sign entered in the active console window.
Activation Prompt or Watermark

When you first start your lab, you may notice a watermark on the desktop indicating that Windows is not activated.

One of the major benefits of virtualization is that virtual machines can be moved and run on any platform. The Hands-on Labs utilizes this benefit and we are able to run the labs out of multiple datacenters. However, these datacenters may not have identical processors, which triggers a Microsoft activation check through the Internet.

Rest assured, VMware and the Hands-on Labs are in full compliance with Microsoft licensing requirements. The lab that you are using is a self-contained pod and does not have full access to the Internet, which is required for Windows to verify the activation. Without full access to the Internet, this automated process fails and you see this watermark.

This cosmetic issue has no effect on your lab.

Look at the lower right portion of the screen
Please check to see that your lab is finished all the startup routines and is ready for you to start. If you see anything other than "Ready", please wait a few minutes. If after 5 minutes you lab has not changed to "Ready", please ask for assistance.
Module 1 - vRealize Orchestrator Overview (30 min)
Introduction

In this module, you will learn how to navigate the vRealize Orchestrator client interface. Once you're familiar with the client, you will run your first workflow and verify the results.
Introduction to vRealize Orchestrator Client

The vRealize Orchestrator Client is the main interface to develop, see and run workflows. In this exercise, you will familiarize yourself with the vRealize Orchestrator Client user interface.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: **vra-01a.corp.local:443**
2. User name: **administrator@vsphere.local**
3. Password: **VMware1!**
4. Click **Login**

Review the Three Different Views

After log in to the vRealize Orchestrator Client, the Run view is selected by Default.
1. Click the dropdown menu at the top of the client to see the other views available: **Design** and **Administer**
Keep **Run** view selected

This section will cover the differences in the three views.

**Run View**

Within this view you can schedule tasks, manage policies, run workflows, access the inventory, and manage permissions. The tabs shown in the screenshot are available on the left side of the client window. Take some time to click on each of the tabs and explore what is available in each.

1. **My Orchestrator** - Summarizes the most recent activities on the Orchestrator server, such as recently modified elements, pending and running workflows, running policies, completed workflows, and workflows that are waiting for user interaction. You can use the My Orchestrator view to perform common administrative tasks such as running a workflow, importing a package, and setting root access rights.
2. **Scheduler** - Displays a list of all scheduled workflows. The workflows are sorted by name or date, together with their status. You can use the Scheduler view to create, edit, suspend, resume, and cancel scheduled workflows.
3. **Policies** - Displays existing policies. You can use the Policies view to create and apply policies. Policies in vRealize Orchestrator are a series of rules, gauges, thresholds, and event filters that run certain workflows or scripts when specific predefined events occur either in Orchestrator or in the platforms that Orchestrator can access through plug-ins.
4. **Workflows** - Provides access to the Orchestrator workflow library. You can use the Workflows view to view information about each workflow. You can also use this view to create, edit, run and interact with workflows.
5. **Inventory** - Displays the objects of the plug-ins that are enabled in Orchestrator. You can use the Inventory view to run workflows on an inventory object.
Select the Design View

1. Click the dropdown menu at the top of the client, and select the Design View

Design view

Within this view you can create and modify workflows and actions. You can also manage resources, configuration elements, and policy templates. The tabs shown in the screenshot are available on the left side of the client. Take some time to click on each of the tabs and explore what is available in each.

1. **Workflows** - Provides access to the Orchestrator workflow library. You can use the Workflows view to view information about each workflow, create, edit, and run workflows, as well as to interact with the workflows.
2. **Actions** - Provides access to the libraries of predefined actions. You can use the Actions view to duplicate actions, export them to a file, or move them to a different module in the actions hierarchical list.
3. **Resources** - Provides access to the list of resource elements. You can use the Resources view to import external objects such as images, sysprep files, HTML templates, XML templates, and custom scripts. In turn, you can then use these as resource elements within workflows.
4. **Configurations** - Provides access to the available configuration elements. You can use the Actions view to create configuration elements to define common attributes across an Orchestrator server.
5. **Packages** - Displays a list of the available packages and where a selected package is used. You can use the Packages view to add, import, export, and synchronize packages.
6. **Inventory** - Displays the objects of the plug-ins that are enabled in Orchestrator. You can use the Inventory view to run workflows on an inventory object.
Select Administer View

1. Click the dropdown menu at the top of the client, and select the Administer View.

Administer View

Within this view you can access the inventory and manage authorizations and packages. The tabs shown in the screenshot are available on the left side of the client. Take some time to click on each of the tabs and explore what is available in each.

1. **Inventory** - Displays the objects of the plug-ins that are enabled in Orchestrator. You can use the Inventory view to run workflows on an inventory object.
2. **Policy Templates** - Displays a list of the available master policies. You can use the Policy Templates view to create policy templates.
3. **Authorizations** - Displays a list of the available authorization elements. You can use the Authorizations view to create and edit authorization elements.
4. **Packages** - Displays a list of the available packages and where a selected package is used. You can use the Packages view to add, import, export, and synchronize packages.
Running your first workflow

Now that you have familiarized yourself with the vRealize Orchestrator Client, you are going to run your first workflow. This will allow you to apply some of the knowledge you have already gained around vRealize Orchestrator.

Return to the Run View

1. Click the dropdown menu at the top of the client, and select Run from the list to return to the Run view

View the Create a Snapshot Workflow
1. Select the **Workflow** tab
2. Navigate to the workflow **Library > vCenter > Virtual Machine Management > Snapshot > Create a snapshot** (Vertical scrolling might be needed)
3. On the right, select the **Schema** tab

Take a moment to look at what the workflow is doing. You will see that it is using an action called `createSnapshot` and then another action called `wim3WaitTaskEnd`. So this workflow will take a snapshot of a virtual machine and then wait for that task to end before the workflow completes.

4. Click the **Start workflow** Icon

### Choose the VM

1. Click on **Not Set** to browse for the vm
Choose the VM


2. Click the Select button
1. Click Next
Enter snapshot creation parameters

1. Type in **Test** for the snapshot name
2. Type in **My First Workflow** for the description

Leave all the other settings at the default values.

3. Click the **Submit** button

Launch the Chrome Browser

1. From the Windows Quick Launch task bar, click the **Google Chrome** icon to open the browser
Log in to the vSphere Web Client

1. Click on the RegionA folder and select the HTML5 Client bookmark
2. Check the box for Use Windows session authentication

This will log you on with the administrator@corp.local credentials

3. Click the Login button
Search for VM

1. Enter **brown** in the search box
2. Click the **brownfield-01a** machine

Manage Snapshots

1. Click the **ACTIONS** button
2. Select **Snapshots**
3. Click on **Manage Snapshots**
Manage Snapshots: Verify Test

1. Select the Test snapshot
2. Verify the snapshot. It should have My First Workflow as its description
   - Click the DELETE button
3. The snapshot can now be removed.
4. Click OK (not shown)
5. Click the DONE button

Close the Chrome Browser

1. Click the X in the top right hand corner of the browser to close it
Close the vRealize Orchestrator Client

1. Close the **vRealize Orchestrator Client** by clicking the **X** in the top right corner.

**Close the vRealize Orchestrator Client**

1. Click the **Exit** to confirm you want to close the vRealize Orchestrator Client.
Conclusion

vRealize Orchestrator allows you to perform any operation on various systems (VMware products and non-VMware products).

Its main interface is a Java client that allows to develop new workflows and run them. From its inventory a global view of all endpoints connected is provided.

You’ve finished Module 1

Congratulations on completing Module 1.

Looking for additional information on vRealize Orchestrator basics, try one of these:

- Go to http://bit.ly/2veqDW2
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 2 - Creating a Basic Orchestrator Workflow** (30 min) (Intermediate)
  Learn how to create a basic workflow to resolve a simple yet routine task such as cloning a virtual machine or taking a snapshot.

- **Module 3 - Introduction to Parameters, Attributes and Scripting Objects** (30 min) (Intermediate)
  Explore and learn about the objects and constructs that are used to extend the functionality of workflows. These include attributes, scripting objects and decision trees as well as the parameters they use.

- **Module 4 - Using Actions in vRealize Orchestrator** (30 min) (Advanced)
  Reuse your JavaScript code by turning JavaScript Functions into vRealize Orchestrator Actions, then use them repeatedly in multiple workflows. Actions represent individual functions that you use as building blocks in workflows and scripts.
• **Module 5 - Creating and Managing Resource Elements** (30 min) (Advanced) Discover how to use external objects such as scripts, XML, or HTML files in workflows.

• **Module 6 - Using the new Web UI to Monitor and Manage vRealize Orchestrator** (30 min) (Intermediate) Explore the all-new vRealize Orchestrator Web UI and the new capabilities it brings, including the web-based workflow viewer, troubleshooting, statistics, performance monitoring and more!

### How to End Lab

To end your lab click on the **END** button.
Module 2 - Creating a Basic Orchestrator Workflow (30 min)
Introduction

This module covers the basics of vRealize Orchestrator workflow creation. Our workflow will use a Script Task to execute a few lines of JavaScript.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.

Log In to vRealize Orchestrator
Use the following credentials

1. Host name: *vra-01a.corp.local:443*
2. User name: *administrator@vsphere.local*
3. Password: *VMware1!*
4. Click *Login*

**Select the Design View**

1. Click the dropdown menu at the top of the client, and select the **Design** View
Creating your First Workflow

vRealize Orchestrator workflows allow you to perform a variety of tasks. Each workflow can have multiple inputs and return multiple outputs. This lesson covers the creation and execution of a simple workflow.

Create a New Workflow

1. Make sure Design view is selected
2. Select Workflow tab
3. Right Click on HOL > 1921-05-CMP > Module 2 workflow folder
4. Click on New workflow

Naming a Workflow

1. Type in your workflow name, for example, "My First Workflow"
1. Type **My First Workflow** for the Workflow name
2. Click the **Ok** button

**Configuring the New Workflow: Schema**

1. Select the **Schema** tab

Workflow schemas are made up of elements. Take a moment to look through the list of available elements.
Configuring the New Workflow: Schema

1. In the Generic section, Select **Scriptable task**
2. Drag the **Scriptable task** element onto the **Schema** between the two existing elements on the canvas

The green arrow represents the Start workflow element and the one to the far right is the End workflow element.
These 2 elements are mandatory, and must be present for the workflow to run.
Configuring the New Workflow: Inputs

1. Select the **Inputs** tab
2. Click the **Add parameter** icon to add a parameter. The name will default to **arg_in_0**

Parameters within vRealize Orchestrator are basically input and output variables of a workflow in the simplest of terms.

Input parameters are needed for most workflows and are values that are passed into the workflow when it starts via a user, application, another workflow, or even an action.

3. Click on **arg_in_0** to change the parameter name
4. Type **vm** for the Attribute name to update the name of this parameter
5. Click the **Ok** button
Configuring the New Workflow: Inputs

1. Click on string in the Type column to open the "Select a type..." window
2. Type VC:VirtualMachine in the Filter box
3. Scroll Up
4. Select the VC:VirtualMachine Type (Note: this value should be at the top of the filtered list)
5. Click the Accept button

Configuring the New Workflow: Description

1. Double-click on the area under the Description column, and type My vm in the Description field

The description is automatically saved.
Editing a Scriptable Task

1. Select the **Schema** tab
2. Click the **Scriptable task** to open it in the lower pane
3. Click the **Undock** icon in the corner of the lower pane to open the contents in a larger window. This is used because of the low resolution in the lab.
1. Select the **Visual Bindings** tab
2. Select the `vm` input parameter
   Drag it over to the **IN** field to create the binding

**Bindings** allow you to connect workflow **inputs** or **outputs** to local parameters. This is needed so the local element can process the parameter.

In this case, the input parameter is passed to a local parameter so it can be used within the **Scriptable Task** element that is part of the workflow. It is standard practice to have all parameters bound in a workflow. If a variable is not bound anywhere, a warning will appear when the workflow is saved - failing validation. The warning message indicates the parameter that is not bound.
Local IN Parameters

1. Select the **IN** tab

Notice that now there is a local IN parameter for the Scriptable task. Bindings can be done either in Visual Binding tab or here in IN or OUT tabs.

**Scriptable Task: Scripting**
1. Select the **Scripting** tab
2. **Enter** the following code (if drag and drop is used, make sure to keep the focus in the scripting tab all the time):

```javascript
System.log(" Name: " + " " + vm.summary.config.name);
System.log(" Hostname: " + " " + vm.summary.guest.hostName);
System.log(" IP Address: " + " " + vm.summary.guest.ipAddress);
System.log(" Description for VM: " + " " + vm.summary.config.annotation);
System.log(" Number of CPU: " + " " + vm.summary.config.numCpu);
System.log(" Amount of Memory in MB: " + " " + vm.summary.config.memorySizeMB);
System.log(" Number of Network Adapters: " + " " + vm.summary.config.numEthernetCards);
System.log(" Number of Disks: " + " " + vm.summary.config.numVirtualDisks);
System.log(" Guest OS: " + " " + vm.summary.config.guestFullName);
System.log("Is template: " + " " + vm.summary.config.template);
```

The code will log information available from the API of the virtual machine entered as input into the workflow. The information logged by the workflow includes:

- The virtual machine name
- The virtual machine hostname
- The virtual machine ip address
- The virtual machine description field
- The cpu count for the virtual machine
- The memory count for the virtual machine
- The network adapter count for the virtual machine
- The disk count for the virtual machine
- The virtual machine guest OS full name
- Whether or not the virtual machine is marked as a template

It is also taking this data and formatting it in a much more readable format.

3. Click the Red X button to close the window and return it to the dock
Configuring the New Workflow: Validation

1. Click the **Validate** icon

   This allows vRealize Orchestrator to check the workflow and make sure there are not any issues, such as parameters that are not bound or used.

2. Click the **Close** button

### Save and Close

1. Click the **Save and close** button
Running Your Workflow

1. Click the **Workflow** tab
2. Browse to **HOL > 1921-05-CMP > Module 2 > My First Workflow**
3. Click on the **Start workflow** icon

1. Click **Not set** to select the virtual machine to use as input into the workflow
Running Your Workflow

1. Navigate to **vSphere vCenter Plugin** > https://vcsa-01a.corp.local:443/sdk > **Datacenters** > **RegionA01** > **vm** > **VRM** > **brownfield-01a**
2. Click the **Select** button
Running Your Workflow

1. Click the **Submit** button
Running Your Workflow

When the workflow completes, the workflow token shows a green check mark next to it. This indicates successful execution of the workflow.
Understanding a Workflow Token

A workflow token represents an instance of a workflow that is running or has run previously.

Find Workflow Token from the Last Run

1. If necessary, expand the workflow My First Workflow
2. Select the workflow token (with the green check mark) from the previous exercise.
Get the Details of the Workflow Instances

1. Select **Schema** tab
2. If necessary, click and drag the **Summary** window up to see the details.
Look at the Variables

1. Click the Variables tab on the lower right hand side of the window

Notice that the value for the vm input is the **brownfield-01a** virtual machine that was selected as input of the workflow. Variables are Inputs, outputs and attributes, and the values shown are the values at the time of completion of the workflow.

Look at the Logs

1. Click the Logs tab on the lower right hand side of the window
2. Make sure **Info** is selected

Feel free to scroll the logs to check all the lines.

Notice that the script worked and all the values were logged in the format specified in the Scriptable Task.
Look at the Events

1. Select **Events** tab

This tab shows a list of events relevant to the workflow execution, including:

- Start of the workflow
- End of the workflow
- Various instances of Server.log, Server.warn, Server.error that are called within the workflow

Note that Events are stored in database and can be seen from any Orchestrator node and at any time.
Conclusion

A vRealize Orchestrator workflow is very easy to create, and it can be executed once or multiple times. A workflow contains parameters (inputs, outputs, attributes) that can be bound to workflow elements. Each workflow run is called a workflow token.

Discover more about workflows in the next module.

You've finished Module 2

Congratulations on completing Module 2.

If you are looking for additional information on Developing workflows, try one of these:

- Go to http://bit.ly/2ugo6gy
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - vRealize Orchestrator Overview** (30 min) (Intermediate) Examine and understand the key features and architecture of the vRealize Orchestrator platform. Learn the methods for accessing and interacting with platform.
- **Module 3 - Introduction to Parameters, Attributes and Scripting Objects** (30 min) (Intermediate) Explore and learn about the objects and constructs that are used to extend the functionality of workflows. These include attributes, scripting objects and decision trees as well as the parameters they use.
- **Module 4 - Using Actions in vRealize Orchestrator** (30 min) (Advanced) Reuse your JavaScript code by turning JavaScript Functions into vRealize Orchestrator Actions, then use them repeatedly in multiple workflows. Actions represent individual functions that you use as building blocks in workflows and scripts.
• **Module 5 - Creating and Managing Resource Elements** (30 min) (Advanced) Discover how to use external objects such as scripts, XML, or HTML files in workflows.

• **Module 6 - Using the new Web UI to Monitor and Manage vRealize Orchestrator** (30 min) (Intermediate) Explore the all-new vRealize Orchestrator Web UI and the new capabilities it brings, including the web-based workflow viewer, troubleshooting, statistics, performance monitoring and more!

**How to End Lab**

To end your lab click on the **END** button.
Module 3 - Introduction to Parameters, Attributes and Scripting Objects (30 min)
Introduction

This module covers the basics to make a workflow, starting with inputs and outputs, then looking at attributes, and finally scripting objects provided by vRealize Orchestrator plugins.

It explains how all these constructs are interconnected, how values and objects flow between the various elements in a workflow, and how they are used to do real work!

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: **vra-01a.corp.local:443**
2. User name: **administrator@vsphere.local**
3. Password: **VMware1!**
4. Click **Login**

Select the Design View

1. Click the dropdown menu at the top of the client, and select the **Design** View
Understanding Input and Output Parameters

Input and output parameters are essential to understand in vRealize Orchestrator, as they are used in nearly every available workflow.

Introduction

Parameters within vRealize Orchestrator are basically input and output variables of a workflow in the simplest of terms.

**Input parameters** are needed for most workflows and are values that are passed into the workflow when it starts via a user, application, another workflow, or even an action. A good example of this might be a workflow that is used to snapshot a virtual machine. An input parameter for such a workflow may be the virtual machine name. Another thing to note about input parameters is that they are read-only, and cannot be changed during workflow execution.

**Output parameters** represent the result of the workflow and can be passed on to other elements when the workflow has completed. For that same virtual machine snapshot workflow, the output parameter is the resulting snapshot. Output parameters are write-only; to read the value of a parameter within a workflow, it cannot be an output parameter. However, this output parameter value is readable for any workflows chained after this one.
Identify Parameters

Time to explore parameters a bit more.

1. Click on the Workflows tab
2. Browse the workflows tree to Library > vCenter > Configuration > Add a vCenter Server Instance
3. Click on Inputs

Here are the input parameters for a workflow used to add vCenter objects to the vRealize Orchestrator inventory. Once a vCenter object is added, it is possible to run other workflows against it - for example, a workflow to clone a virtual machine.

Back to parameters. Observe the different input parameters being gathered. There is a "Type" column, which dictates the type of value the input parameter will accept. For example, the password input parameter type is SecureString, which allows vRealize Orchestrator to take in the string but in a secure way since it is a password and to not show this value in clear text.

The best practice is to format input parameter names in a format called camelCase. camelCase is where the first word in the name starts with a lowercase letter and all proceeding ones start with a capital letter. A good example would be the parameter "sessionPerUser". camelCase is the standard used for parameters in vRealize Orchestrator and what we will be using throughout the lab.

4. Now, click on the Output tab
Are there any output parameters listed? What is their type?

**Types of Parameters**

Each parameter has a type. This type can be simple type like string, number, array, ... but it can also be complex type such as VC:VirtualMachine, vCACAFE:CatalogItemRequest or REST:RESTHost. All of these complex types come from plugins that have been added to vRealize Orchestrator.

When an input is chosen, vRealize Orchestrator will only allow objects from that input's type to be selected in order to avoid user mistakes. A complex object can carry a lot of properties that can be accessed directly from within a scriptable task, which makes it very flexible.

The fact that vRealize Orchestrator accounts for all variable types simplifies development, and makes it more reliable.
Understanding Workflow Attributes

Workflow attributes are one of the key constructs used to carry values between elements in a workflow.

Introduction

Attributes are variables that are generally read/write, and they can be used to transfer data between the elements of a workflow. An attribute can be locked so it is read-only across all workflow elements. This is very useful to define a static value that is needed in several elements of a workflow.

Read-only workflow attributes act as global constants for a workflow. Writeable attributes act as a workflow’s global variables.

Values of attributes can be set and modified as follows:

- Set attribute value in a workflow
- Assign attribute value from a configuration element
- Attribute linked to the output parameter of a workflow element

Identify Attributes

1. Click on the Workflows tab
2. Browse the workflows tree to HOL > 1921-05-CMP > Module 3 > Playing with attributes
3. Click the **Edit** icon

### Edit Attributes

1. Click on the **General** tab
2. **Scroll down** to the bottom of the General tab

Here are the attributes listed. They can be blank or have a default value, and like Parameters, they have a type.

3. Click on the **checkbox** near **attribute2**

By doing this, the attribute is now read-only, and can be used by any workflow element only as IN.

4. Double-click into the value field for the **attribute1** attribute and type in **my value 1**

This would set the attribute's default value to **my value 1**. It will remain read/write and can used by any workflow element as IN and/or OUT.
Look at the Value of the Attributes

1. Select **Schema** tab
2. Click on **Run** button
3. Once the workflow completes, look at the logs by clicking the **Logs** tab on the right
4. The log output shows the Business Group Name, and it shows the attribute value twice (because the workflow contains two scriptable tasks that show attribute values.)

Note: if necessary, you can click to the left of the Logs window and drag to expand it.
Change the Value of an Attribute in a Workflow Element

1. Scroll to the middle of the schema
2. Select change attr value workflow element
3. Click the Undock icon to open the lower panel in a new window
1. Select **Visual Binding** tab
2. Drag and Drop **attribute1** from **Out Attributes** box to **OUT** box
3. Notice that **attribute2** is only available as **In Attributes** and not as **Out Attributes** (it is because the attribute has been made read-only in a previous step).
Update Script

1. Select **Scripting** tab
2. Enter the following code:

   ```javascript
   attribute1 = attribute1 + " - my new value";
   ```
3. Click the X to close the window and return to the Schema panel

Look at the Value of the Attributes

1. Select **Schema** tab
2. Click on **Run** button
3. Look at the logs
Notice the second time the value of attribute 1 is shown, "my new value" is visible but with null at the beginning. This does not seem to be the expected behavior - let's see in the next steps how to fix that.

**Change Value of an Attribute in a Workflow Element**

1. Scroll to the middle of the schema
2. Select **change attr value** workflow element
3. Click the **Undock** icon to open the lower panel in a new window
Change the Value of an Attribute in a Workflow Element (Continued)

1. Click the **Visual Binding** tab
2. Drag and Drop **attribute1** from **In Attributes** box to **IN** box
3. Click the **X** to close the window and return to the Schema panel
Look at the Value of the Attributes

1. Select **Schema** tab
2. Click on **Run** button
3. Look at the logs

Notice the second time the value of attribute 1 is shown, "my new value 1" has been well concatenated to " - my new value".

Because the attribute "attribute1" was not an IN of the scriptable task, its value was null. After adding it as an IN, its value is available in the scriptable task and properly concatenated.

**Save and Close**

1. Click on **Save and close**
Using Scripting Objects

**Scripting Objects** are JavaScript classes that can be used in any Scriptable Task and/or Action. They can be browsed from the API explorer and comes from vRealize Orchestrator core as well as the plugins.

Select "Show VM IP Address" workflow

1. Select **Workflow** tab
2. Expand **HOL > 1921-05-CMP > Module 3**. Select the **Show VM IP Address** workflow
3. Click on **Edit** icon
Select "show ip" scriptable task

1. Select **Schema**
2. Select **show ip** scriptable task
3. Select the **Scripting** tab

From the scripting tab you can use JavaScript code. You also have the ability to leverage the API explorer to view the vRealize Orchestrator types available and what type of data is available and can be scripted against. In your workflow we will be leveraging the VC:VirtualMachine type.

4. Click the **Search API** icon
API Explorer: API Search

1. Type `VC:VirtualMachine` in the Containing text field
2. Click the Search button
3. Click on Name column to sort the result by name.
4. Select `VC / VC:VirtualMachine`
5. Click Go to selection
6. Click the Close button

Undock Scripting tab

1. Click on Undock icon
**VC:VirtualMachine** Type should be selected in the type list, below there is a description of the object type.

1. **Scroll up** if necessary
2. Click the Scripting Object **VcVirtualMachine**

A scripting object is the javascript object that can be used in a scriptable task.
1. Click the arrow to expand **VcVirtualMachine**

This shows all properties and methods in the **VcVirtualMachine** scripting object. Take a moment to click on some of the properties. When a property or method is selected, the lower panel displays details about this property or method.
API Explorer: Browsing

1. **Scroll down** until **guest** attribute is visible
2. Select the **guest** attribute.
3. Click on the **VcGuestInfo** Return Type

Return Types are the value we can pull via the API for the given object type and property.
### Find ipAddress

1. Expand **VcGuestInfo** by clicking on the arrow.
2. Expand the panel by **drag and drop** the bottom part
3. Observe all the available attributes, **ipAddress** is listed as one of the information

### Show IP address

If we summarize, we started at the **VC:VirtualMachine** object which points toward the Scripting class **VcVirtualMachine**, followed by **guest**, then **ipAddress**. From a javascript standpoint, it is written like this: **VM.guest.ipAddress**

1. The following code should be already present:

   ```javascript
   System.log(VM.guest.ipAddress);
   ```

   This code writes the IP address of the machine in the log.
2. Click on the X icon to close the window and dock it back.

**Launch Workflow**

1. Click on **Run** button

**Set Virtual Machine input**

1. Click on **Not Set**
Select Virtual Machine

1. Navigate to **vSphere vCenter Plugin** > https://vcsa-01a.corp.local:443/sdk > **Datacenters** > **RegionA01** > **vm** > **puppet-01a**
2. Click **Select**
Submit the workflow

1. The machine selected in the previous step should be visible
2. Click **Submit**
Find the IP Address

The IP Address of the VM selected when you ran the workflow is shown in the logs.

Close the workflow editor

1. Click on **Save and close** to close the workflow editor and come back to the workflow list.
Conclusion

Inputs, outputs, and attributes are the variables available in vRealize Orchestrator. They can be passed between workflow elements within a workflow, or between workflows, and their type allows for consistency across a process.

You've finished Module 3

Congratulations on completing Module 3.

If you are looking for additional information on Key Concepts of workflows, try one of these:

- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - vRealize Orchestrator Overview** (30 min) (Intermediate) Examine and understand the key features and architecture of the vRealize Orchestrator platform. Learn the methods for accessing and interacting with platform.
- **Module 2 - Creating a Basic Orchestrator Workflow** (30 min) (Intermediate) Learn how to create a basic workflow to resolve a simple yet routine task such as cloning a virtual machine or taking a snapshot.
- **Module 4 - Using Actions in vRealize Orchestrator** (30 min) (Advanced) Reuse your JavaScript code by turning JavaScript Functions into vRealize Orchestrator Actions, then use them repeatedly in multiple workflows. Actions represent individual functions that you use as building blocks in workflows and scripts.
- **Module 5 - Creating and Managing Resource Elements** (30 min) (Advanced) Discover how to use external objects such as scripts, XML, or HTML files in workflows.
- **Module 6 - Using the new Web UI to Monitor and Manage vRealize Orchestrator** (30 min) (Intermediate) Explore the all-new vRealize Orchestrator...
Web UI and the new capabilities it brings, including the web-based workflow viewer, troubleshooting, statistics, performance monitoring and more!

**How to End Lab**

To end your lab click on the **END** button.
Module 4 - Using Actions in vRealize Orchestrator (30 min)
Introduction

This module explains in detail how Actions are created, used, and why they are so important to the development of clean workflows. They are a key component in vRealize Orchestrator, allowing you to easily reuse a piece of code in multiple workflows and even in other actions.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: **vra-01a.corp.local:443**
2. User name: **administrator@vsphere.local**
3. Password: **VMware1!**
4. Click **Login**

Select the Design View

1. Click the dropdown menu at the top of the client, and select the **Design** View
Overview of Actions

Actions in vRealize Orchestrator are generally pieces of code that can be reused often. In another language they could be referred to as functions.

The main difference between an Action and a workflow is that Actions can only return one variable and a workflow can return many. Actions are also built entirely using JavaScript, and there is no visual programming like with a workflow.

Browse Out of the Box Actions

1. Select the Actions tab

The list of folders that are displayed are called modules.

Modules are a way to group actions, like a folder but only one level is allowed - there is no possibility to create "submodules." The naming convention is quite different and looks like a URL. This is done as a way to track each module's purpose and its author or provider. For example, com.vmware.library.vc.networking is a module with actions associated with vCenter networking provided by VMware.
1. Scroll down
2. Expand the module `com.vmware.library.vc.basic`
3. Select the `cloneVM` action
4. On the right hand side select the **Scripting** tab

The scripting tab of an action gives details about the parameters used and the JavaScript code being used. Notice that tabs like schema or visual bindings do not exist for an action.

Take a few minutes to look through the other modules and actions available.

Every module and action starting with `com.vmware.library` in the name is available out of the box with vRealize Orchestrator. VMware has you covered for a lot of things that can be used right away just by dragging and dropping an action into a workflow.
Create an Action

Now that you’ve learned more about actions, let’s create a module with a basic action in it.

Creating a Module

1. Select the **Actions** tab
2. **Scroll up** to the top of the Actions list if necessary
3. Right click **Administrator @ vra-01a.corp.local**
4. Select **New module**

Naming a Module

1. Type **com.rainpole.vc.vm** for the Module name

Remember, modules are a way to group actions. One thing that makes them different from workflow folder is that only one level is allowed (there is no submodule.) Also, their naming convention tends to look like a Java class and includes information on what is included in the module.
For example, the name used (vc.vm) is included in the end to reflect that the module is going to contain modules related to vCenter virtual machines.

2. Click the **Ok** button

**Creating an Action**

1. Select the **com.rainpole.vc.vm** module (it should be selected by default after the previous step)
2. Click the **Add Action** icon
Remember, Actions in vRealize Orchestrator are generally pieces of code that can be reused often.

## Naming an Action

![Image of Action Name Dialog]

1. Type `osNameToGuestOS` for the Action name
2. Click the **Ok** button

In vRealize Orchestrator it is standard practice to name items using camelCase. camelCase means that the first word in the name starts with a lower case letter and all proceeding ones start with capital letters. Also the action name should be descriptive. In this module the action takes a OS friendly name and converts it virtual machine guest OS identifier that orchestrator and vSphere understand.

## Configuring the Action

![Image of Scripting Configuration]

4. Set attribute name to `osName`
5. Click **Ok**
1. Select the **Scripting** tab
2. Click **Add parameter** icon

This will be an input parameter that the action will take in as a string

3. Click on **arg0**
4. Type **osName** for the Attribute name

Notice that it is camelCase for the naming convention

5. Click the **Ok** button

**Return Type**

1. Click the **void** link, this is for the Return Type
2. Type **VC:VirtualMachineGuest** in the Filter field
3. Select the **VC:VirtualMachineGuestOsIdentifier** type
4. Click the **Accept** button

The Return Type the format of the output.
vRealize Orchestrator is JavaScript based, and actions are based on it. Unlike with workflows, there are no visual programming options available.

This JavaScript code takes an input parameter string that is for a Windows or CentOS OS and convert it to a type understood by Orchestrator that identifies a virtual machine's guest OS type. Take a few minutes to look over the code.

1. **Type** the code into the script window:

   ```javascript
   if (osName == "Windows" || osName == "Windows 2012" || osName == "Windows 2012 R2")
       return VcVirtualMachineGuestOsIdentifier.windows8Server64Guest;
   else if (osName == "CentOS" || osName == "CentOS 7" || osName == "CentOS 6" || osName == "CentOS 5")
       return VcVirtualMachineGuestOsIdentifier.centos64Guest;
   ```

2. Click the **Save and close** button
Using an Action in a Workflow

Add this action to a workflow to test, and make sure it takes a string value and converts it to the correct type and value.

1. Select the **Workflow** tab
2. Expand the **HOL** Folder
3. Expand the **1921-05-CMP** Folder
4. Right Click The **Module 4** Folder
5. Select **New Workflow**
Naming a Workflow

1. Type **My First Action Workflow** for the Workflow name
2. Click the **Ok** button

Adding an Action to a Workflow

1. Select the **Schema** tab
2. Type the name of the action **osNameToGuest**
3. Click the search button
4. **Drag and drop** the action on the **Schema** between the Start and End workflow elements and make sure the workflow element is **selected**
Promote Workflow Parameters

1. Click the **Setup** button (action must be selected in the workflow schema)
2. Click the **Promote** button

Accept the defaults on Promote Workflow Input/output Parameters.

vRealize Orchestrator is smart enough to create the right binding for Input and Output parameters that were defined in the action element. This step can be skipped and binding can be done manually otherwise.

Save and Close

1. Click the **Save and close** button
Running the Workflow

1. Make sure **HOL > 1921-05-CMP > Module 4 > My First Action Workflow** is selected.
2. Click the **Start workflow** button
Common Parameters

1. Type **Windows 2012 R2** for the guest os name
2. Click the **Submit** button
Verifying the Workflow Run

1. Make sure the Schema tab is selected
2. Select the Variables tab
3. The workflow returned a Type of VC:VirtualMachineGuestOsIdentifier and the correct value for Windows.

For extra credit, run the workflow again with other values like CentOS as the name of the Operating System and observe the return!
Conclusion

Actions are like "functions" in other programming languages. They are meant to be reused across multiple workflows, or even in other actions. They are also very fast to execute.

You've finished Module 4

Congratulations on completing Module 4.

If you are looking for additional information on Developing Actions, try one of these:

• Go to https://bit.ly/2ulxzwU
• Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

• Module 1 - vRealize Orchestrator Overview (30 min) (Intermediate) Examine and understand the key features and architecture of the vRealize Orchestrator platform. Learn the methods for accessing and interacting with platform.
• Module 2 - Creating a Basic Orchestrator Workflow (30 min) (Intermediate) Learn how to create a basic workflow to resolve a simple yet routine task such as cloning a virtual machine or taking a snapshot.
• Module 3 - Introduction to Parameters, Attributes and Scripting Objects (30 min) (Intermediate) Explore and learn about the objects and constructs that are used to extend the functionality of workflows. These include attributes, scripting objects and decision trees as well as the parameters they use.
• Module 5 - Creating and Managing Resource Elements (30 min) (Advanced) Discover how to use external objects such as scripts, XML, or HTML files in workflows.
• Module 6 - Using the new Web UI to Monitor and Manage vRealize Orchestrator (30 min) (Intermediate) Explore the all-new vRealize Orchestrator
Web UI and the new capabilities it brings, including the web-based workflow viewer, troubleshooting, statistics, performance monitoring and more!

How to End Lab

To end your lab click on the END button.
Module 5 - Creating and Managing Resource Elements (30 min)
Introduction

Resource Elements are a very important and useful construct in vRealize Orchestrator. From storing scripts, to HTML or XML templates, or even images, they can be easily used from within a workflow.

Each aspect of resource element management, from creation to deletion, is covered from an UI standpoint (using vRealize Orchestrator Client) and from a code standpoint (using javascript code in a workflow) in this module.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: `vra-01a.corp.local:443`
2. User name: `administrator@vsphere.local`
3. Password: `VMware1!`
4. Click **Login**

Select the Design View

1. Click the dropdown menu at the top of the client, and select the **Design** View
Creating Resource Elements

Let's see how to create resource elements from the vRealize Orchestrator Client, and also directly from a workflow.

Import Resource Element

To start, you will import an HTML page into vRealize Orchestrator as a resource element. It can be used later on in a workflow to send customized emails.

1. Select Resources tab
2. Expand the HOL Folder and select 1921-05-CMP
3. Click on Import Resources... button
4. Browse the folder C:\hol\HOL-1921-05-CMP
5. Select email-template.html
6. Click on Open button
vRealize Orchestrator allows certain types of files to be viewed (such as txt, csv, html and a few other types.) These files can be viewed in the Resources section of the Design view.

1. Select the resource from the left panel
2. Click on **Viewer** tab
Create a New Workflow

Now let's see how it is possible to create a Resource Element from within a workflow.

1. Click on **Workflows** tab
2. Expand the **HOL > 1921-05-CMP** Folder and Right Click on **Module 5**
3. Click on **New Workflow**

**Name the New Workflow**

Now let's see how it is possible to create a Resource Element from within a workflow.

1. Click on **Workflows** tab
2. Expand the **HOL > 1921-05-CMP** Folder and Right Click on **Module 5**
3. Click on **New Workflow**
1. Enter **Create Resource Element** as the workflow name
2. Click **OK**

### Add Scriptable Task

1. Select the **Schema** tab
2. Make sure the **Generic** tab is selected
3. Drag and drop the **Scriptable task** item to the arrow in the schema
1. Make sure the **Scriptable task** is selected
2. Click the **Undock** icon to open the Scriptable task pane in a new window
Add Javascript Code to Create Resource Element

1. Click on **Scripting** tab
2. Copy the following code (it is recommended to drag and drop it from the manual)

   ```javascript
   var myMimeAttachment = new MimeAttachment();
   myMimeAttachment.content = "vro generated content\r\n1921-05-CMP";
   myMimeAttachment.mimeType = "text/plain";
   Server.createResourceElement("HOL/1921-05-CMP" , "my-vro-generated-resource.txt" , myMimeAttachment);
   
   The javascript code creates an object MimeAttachment which could be assimilated to a file and assign a content and a content type (line 1-3). Then the method createResourceElement (line 4) creates a new resource in the given folder, with the given name containing the MimeAttachment object.

3. Click on the **X** to close the window, and then click on **Save and close** (not shown)
Launch the Workflow

1. Select the workflow **Create Resource Element** that was just created
2. Click on **Start Workflow** button

Wait for the workflow to finish (green check mark)
View the Newly Created Resource Element

1. Click on Resources tab
2. Expand HOL > 1921-05-CMP folder
3. Click on my-vro-generated-resource.txt Resource Element
4. Click on Viewer tab

The content matches what was in the code from the previous step. This can be easily changed to integrate user inputs or data from other systems.
Managing Resource Elements

After a Resource Element is added to vRealize Orchestrator, it can be updated. This can be done from the Client or from a workflow.

**Update Resource with vRealize Orchestrator Client**

1. Click on **Resources** tab
2. Expand the **HOL > 1921-05-CMP** Folder. Select **email-template.html**
3. Click the **Edit** button
Change Version of the Resource

Each object in vRealize Orchestrator is versioned (Workflows, Configuration Elements, Resource Elements, and so on.) However, when an update is done from the Client the version number is not automatically incremented. That is why it needs to be done manually before uploading the new content.

1. Select **General** tab
2. Click on the last number in the Version row
3. Add a **comment** for the version change
4. Click **OK**
5. Click on **Save and close** (Not shown)
Update the Resource Element with vRealize Orchestrator Client

1. **Right-Click** the `email-template.html` resource and select **Update resource...** (not shown)
2. Browse the folder `C:\hol\HOL-1921-05-CMP`
3. Select `email-template-2.html`
4. Click **Open**
5. Click **Update Resource** when prompted (not shown in the screenshot)
Update Resource with Workflow

1. Click on **Workflows** tab
2. Expand the **HOL > 1921-05-CMP > Module 5 Folder**
3. Right Click and select **New Workflow**

Name the New Workflow

1. Enter "Update Resource Element" as workflow name
2. Click **OK**
Add Input: resource

1. Select **Inputs** tab
2. Click the **Add parameter** button
3. Click the newly created input **arg_in_0**
4. Enter the name of the input: **resource**
5. Click the **Ok** button
Assign Type

1. Click the type of the input: **string**
2. Type **ResourceElement** in the Filter box
3. Select **ResourceElement** from the list
4. Click **Accept**
Add Scriptable Task

1. Click to **Schema** tab
2. Select **Generic** tab
3. Drag and Drop **Scriptable task** item to the arrow in the schema
Undock Scriptable Task configuration

1. Make sure the **Scriptable task** workflow element is selected
2. Click the **Undock** icon to open the lower pane in a new window

Bind Input

1. Click on the **Visual Binding** tab
2. Drag and drop **resource** from In Parameters to IN

### Add Javascript Code to Create Resource Element

1. Click on **Scripting** tab
2. Copy the following code (drag and drop from the manual is recommended):

   ```javascript
   var mimeAttachment = resource.getContentAsMimeAttachment();
   mimeAttachment.content += "\n\nthis is new content";
   resource.setContentFromMimeAttachment(mimeAttachment);
   
3. Click on the X to close the window, and then click **Save and close** (not shown)
Launch the Workflow

1. Select the workflow **Create Resource Element** that was just created
2. Click on **Start Workflow** button
Fill Parameters

1. Click on Not Set [Not shown]
2. Type my-vro
3. Select my-vro-generated-resource.txt by double clicking it - without a double click, the resource will not be selected!
4. Click Submit

Wait for the workflow to finish (green checkmark.)
**View the Newly Created Resource Element**

1. Click on **Resources** tab
2. Expand the **HOL > 1921-05-CMP Folder**
3. Click on **my-vro-generated-resource.txt** Resource Element
4. Click on **Viewer** tab

The content has been successfully updated.
Using Resource Elements

Now let's use a Resource Element in a workflow. Previously, an email template was added as a resource element. Let's see how it can be used to send an actual email.

Create a New Workflow

Now let's see how it is possible to create a Resource Element from within a workflow.

1. Click on Workflows tab
2. Expand the HOL > 1921-05-CMP Folder and Right Click on Module 5
3. Click on New Workflow
Name the New Workflow

1. Enter **Use Resource Element** as workflow name
2. Click **OK**

Add Input: resource

1. Select **Inputs** tab
2. Click the **Add parameter** button
3. Click the newly created input **arg_in_0**
4. Enter the name of the input: **resource**
5. Click the **Ok** button
Assign Type

1. Click the type of the input: **string**
2. Type **ResourceElement** in the Filter box
3. Select **ResourceElement** from the list
4. Click **Accept**
1. Click the **Add parameter** button
2. Click the newly created input **arg_in_0**
3. Enter the name of the input: **VM**
4. Click the **Ok** button
Assign Type

1. Click the type of the input: **string**
2. Type **VC:VirtualMachine** in the Filter box
3. Scroll up
4. Select **VC:VirtualMachine** from the list
5. Click **Accept**
Add Scriptable Task

1. Click to **Schema** tab  
2. Select **Generic** tab  
3. Drag and Drop **Scriptable task** item to the arrow in the schema

Add Workflow

1. Enter **send em** in the search box
2. Click on the **search** button to find the Send Email to Dev User workflow
3. Drag and drop the workflow **Send Email to Dev User** after the scriptable task

## Bind Workflow Inputs

1. Click on **Setup...** (if Setup... is not visible make sure, the workflow element is selected)
2. Click on **Value** (at the top)
3. Click on **Promote**

This automatically creates new attributes and bind them to the workflow element. It is way quicker than doing it one by one.
Select Scriptable Task Item

1. Click on **Scriptable task** workflow element
2. Click the **Undock** icon to open the lower pane in a new window
Bind Input

Ensure that the **Visual Binding** tab is selected.

1. Drag and Drop **resource** from **In Parameters** to **IN**
2. Drag and Drop **VM** from **In Parameters** to **IN**
3. Drag and Drop **subject** from **Out Attributes** to **OUT**
4. Drag and Drop **content** from **Out Attributes** to **OUT**
Add Javascript Code to Create Resource Element

1. Click on **Scripting** tab
2. Copy the following code (drag and drop from the manual is recommended):

```javascript
subject = "Machine info " + VM.name;
content = resource.getContentAsMimeAttachment().content;
content = content.replace("{{name}}", VM.name);
content = content.replace("{{cpu}}", VM.summary.config.numCpu);
content = content.replace("{{memory}}", VM.summary.config.memorySizeMB + " MB");
content = content.replace("{{ipAddress}}", VM.guest.ipAddress);
```

3. Click on the **X** to close the window, and then click **Save and close** (not shown)
Launch the Workflow

1. Select the workflow **Use Resource Element** that was just created
2. Click on **Start Workflow** button

Enter Resource Parameter

1. For resource parameter (first one), click on **Not Set** (Not shown)
2. Type **email-te**
3. Select **email-template.html** (making sure to double-click)
**Enter VM Parameter**

1. For VM parameter (second one), click on **Not Set** (Not shown).
2. Expand and Select **vSphere vCenter Plugin** >
   **https://vcsa-01a.corp.local:443/sdk > Datacenters > RegionA01 > vm > puppet-01a**
3. Click **Select**
4. Click **Submit**

Wait for the workflow to finish (green checkmark.)

**Open Chrome Browser from Windows Quick Launch Task Bar**
1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar.

**Launch Webmail**

1. Click the bookmark folder **HOL Admin**
2. Click the **Webmail** bookmark

**Log In to devuser Mailbox**

1. Enter username: **devuser@rainpole.com**
2. Enter password: **VMware1!**
3. Click on **Login**
View Email from Workflow

1. Click Refresh until an email appears in the inbox (This can take up to a minute)
2. Select the email in the email list
3. Check the value in the email

Information from the machine has been populated into the email template! This show how a static vRO resource object can be easily used to generate dynamic content.

When finished reading the new email message, close the Chrome window (not pictured)
Conclusion

Resource Elements are very useful to store templates (HTML or XML,) scripts, or even images. It is very easy to use them from a workflow, to add content to them, and to use them in other systems like Emails, REST calls, and more.

You've finished Module 5

Congratulations on completing Module 5.

If you are looking for additional information on Resource Elements, try one of these:

- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most. [Add any custom/optional information for your lab manual.]

- **Module 1 - vRealize Orchestrator Overview** (30 min) (Intermediate) Examine and understand the key features and architecture of the vRealize Orchestrator platform. Learn the methods for accessing and interacting with platform.
- **Module 2 - Creating a Basic Orchestrator Workflow** (30 min) (Intermediate) Learn how to create a basic workflow to resolve a simple yet routine task such as cloning a virtual machine or taking a snapshot.
- **Module 3 - Introduction to Parameters, Attributes and Scripting Objects** (30 min) (Intermediate) Explore and learn about the objects and constructs that are used to extend the functionality of workflows. These include attributes, scripting objects and decision trees as well as the parameters they use.
- **Module 4 - Using Actions in vRealize Orchestrator** (30 min) (Advanced) Reuse your JavaScript code by turning JavaScript Functions into vRealize Orchestrator Actions, then use them repeatedly in multiple workflows. Actions represent individual functions that you use as building blocks in workflows and scripts.
• **Module 6 - Using the new Web UI to Monitor and Manage vRealize Orchestrator** (30 min) (Intermediate) Explore the all-new vRealize Orchestrator Web UI and the new capabilities it brings, including the web-based workflow viewer, troubleshooting, statistics, performance monitoring and more!

**How to End Lab**

To end your lab click on the **END** button.
Module 6 - Using the new Web UI to Monitor and Manage vRealize Orchestrator (30 min)
Introduction

As a Cloud Admin, it is important to be able to quickly monitor the current status of infrastructure, running workflows and integrations within vRealize Orchestrator. The standard vRealize Orchestrator interface is great for this, however, the previous iteration ran exclusively on Java which is showing its age.

New with the latest version of vRealize Orchestrator, an HTML5 Web UI has been added to allow monitoring and management of vRealize Orchestrator.

This module gives an overview of the possibilities with this new Orchestrator Web UI.

However, before getting started, something needs to happen in vRealize Orchestrator (workflow running, failed, ...), so the first step here will be to generate some content using a vRO workflow.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: vra-01a.corp.local:443
2. User name: administrator@vsphere.local
3. Password: VMware1!
4. Click Login
Launch workflow "Generate content"

1. Select Workflows tab
2. Navigate to HOL > 1921-05-CMP > Module 6
   Select workflow Generate content
3. Click the Start workflow... button
Use Web Client to Troubleshoot

The Web Client can be very useful for troubleshooting workflow failures.

Open Chrome Browser from Windows Quick Launch Task Bar

1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar.

Open the monitoring client

![Open the monitoring client](image_url)
1. Click the bookmark vRealize Orchestrator
2. Click the link OPEN THE MONITORING CLIENT

**Login to monitoring client**

![Login screen](image)

The monitoring client is using the same authentication as vRealize Automation

1. Enter **administrator** as the username
2. Enter **VMware1!** as the password
3. Click the Sign in button
Main dashboard

1. The **Dashboard** button allows you to return to this page any time.
2. The **Workflow Runs** gives a quick overview of what is currently happening. It allows an administrator to easily identify the failure rate as well as the current load by looking at the running workflow.
3. The Recent workflow runs view gives a quick sneak pick of the latest workflow launched. By looking at it, it is easy to find out the most recent workflows failed or are still running.
4. User interactions is also important as it may be used for Approval and other actions waiting for cloud admin attention.

Overall this dashboard gives a great overview of what is going on in vRealize Orchestrator and many links to dig deeper into the current state of your Orchestrator environment.
Workflow failed

From the main dashboard,

1. Click the RED part of the diagram to be redirected to the page with all failed workflows

List of workflow failed

This page will shows the list of recent failed workflows. Note that this is actually the "Workflow Runs" page with a filter on the tag state:"failed".

By looking at the finished date, it gives hints to know if it is an isolated incident or a massive failure.

1. **Click** the first failed workflow to get more details.
Workflow failure details

1. The schema gives an overview on where the workflow failed
2. The exception message is shown on the right
3. If the failure occurred in a sub workflow, it is possible to see the path to the parent workflow
4. Click the Logs tab to get more information
**Logs tab**

1. The main part show all the logs
2. A dropdown allows to filter the logs by severity (ALL, INFO, WARNING, ERROR, DEBUG)
3. Two button are there to choose between logs from **Server** or **System**, it references how the logs has been created
   Either with Server.log() (stored in database) or System.log() (stored in log file)

**Variables tab**

1. Click the **Variables** tab to show the list of variables (inputs, outputs or attributes)

We can use the variables tab to dig deeper into currently assigned variables in a workflow run. This is helpful for troubleshooting as you can start to identify if variables are assigning in the way you would normally expect.
Show related Runs

1. Click the link **SHOW RELATED RUNS** to see all runs for the same workflow

Workflow Runs for a specific workflow

Returning to the Workflow Runs page

1. We can place a filter here based on workflowId. All workflow runs for a given workflow are shown below in the list after you apply the filter as indicated in the screenshot above.
2. As shown in the screenshot, we can quickly see that all runs have failed on this workflow.
Use Web Client to Explore

The Web Client is also an awesome tool to quickly explore the vRealize Orchestrator configuration and environment. Since there is no client that needs to be installed, its suited perfectly for non-developers who expect a web interface for interfacing with their platforms. A good example of teams like this are operations teams who don't need full access but still have an interest into looking at how vRealize Orchestrator is configured, and what a workflow contains.

Explore the workflows

1. Expand Workflows
2. Select the Library tab
3. Each workflow is represented as a card with its name, its version and a list of tags.

First thing to notice is the folder structure has disappeared to be replaced by tags only.
1. **Click** the list button to switch to another type of view.

The key difference here is that we are able to see a description of the workflow in place as well, as shown above
Let's search a workflow to start a vCenter virtual machine.

1. Enter **vcenter** in the search bar and **Press Return**
2. Enter **start** in the search bar and **Press Return**
3. **Click** the name of the workflow to look at it.

Searching a workflow is now way easier than with a folder structure, with only few tags it is very easy to find what needed.
Look at the workflow

1. Select the **Schema** tab

   The workflow schema is now visible.

2. Select the **ADD TO FAVORITE** button
1. Select **Dashboard** tab

Now the workflow is easily accessible from the Favorite workflows list

2. **Click** the workflow link to come back to the workflow details
Look at a workflow element

1. Select the **Schema** tab
2. Select the workflow element **VM is poweredOn?**
3. Select **Scripting** tab

The code for the workflow element can be seen and exported.

4. Double-click on the workflow element **Wait for task** to drill down to this sub workflow
Drill down to a sub workflow

1. **Select** the first workflow element
2. Select the **In** tab
   Observe the workflow element inputs
3. Observe the path from the parent workflow to the current workflow
   This is very convenient to easily come back to the parent workflow
Explore the configurations

1. Select the Configurations tab
2. Enter hol in the search bar and Press return
3. Click the DETAILS link
1. Select the **Attributes** tab
2. Observe the list of attribute and their values
1. Select the **Inventory** tab
2. **Scroll** down
3. **Expand** vSphere vCenter Plug-in
1. Navigate to https://vcsa-01a.corp.local:443/sdk > Datacenters > RegionA01 > vm
   Select base-linux-cli
2. Scroll down
3. Observe the details of the virtual machine, like ipAddress, number of vCPUs, and other key details of the machine
**Conclusion**

*The new Web UI provides a nice way to visualize workflows and actions without having to log in to the vRealize Orchestrator client. It is very convenient for Operation team as well, troubleshooting can be done directly from a web browser.*

You've finished Module 6

Congratulations on completing Module 6.

If you are looking for additional information on Web UI, try one of these:

- Click on this link
- Or go to [https://tinyurl.com/yb44xqpd](https://tinyurl.com/yb44xqpd)
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - vRealize Orchestrator Overview** (30 min) (Intermediate) Examine and understand the key features and architecture of the vRealize Orchestrator platform. Learn the methods for accessing and interacting with platform.
- **Module 2 - Creating a Basic Orchestrator Workflow** (30 min) (Intermediate) Learn how to create a basic workflow to resolve a simple yet routine task such as cloning a virtual machine or taking a snapshot.
- **Module 3 - Introduction to Parameters, Attributes and Scripting Objects** (30 min) (Intermediate) Explore and learn about the objects and constructs that are used to extend the functionality of workflows. These include attributes, scripting objects and decision trees as well as the parameters they use.
- **Module 4 - Using Actions in vRealize Orchestrator** (30 min) (Advanced) Reuse your JavaScript code by turning JavaScript Functions into vRealize Orchestrator Actions, then use them repeatedly in multiple workflows. Actions represent individual functions that you use as building blocks in workflows and scripts.
• **Module 5 - Creating and Managing Resource Elements** (30 min) (Advanced)
  Discover how to use external objects such as scripts, XML, or HTML files in workflows.

**How to End Lab**

To end your lab click on the **END** button.
Conclusion

Thank you for participating in the VMware Hands-on Labs. Be sure to visit http://hol.vmware.com/ to continue your lab experience online.

Lab SKU: HOL-1921-05-CMP

Version: 20181113-014644