

VMware vRealize Orchestrator 8.6.1 Release Notes

vRealize Orchestrator 8.6.1

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2023 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

- 1** Introduction 4
- 2** Release Versions 5
- 3** What's New in vRealize Orchestrator 8.6.1 6
- 4** Deploying the vRealize Orchestrator Appliance 7
- 5** Upgrading and Migrating vRealize Orchestrator 8
- 6** Plug-ins Installed with vRealize Orchestrator 8.6.1 9
- 7** Earlier Releases of vRealize Orchestrator 10
- 8** Resolved Issues 11
- 9** Known Issues 13
- 10** Previously Known Issues 14

Introduction

1

vRealize Orchestrator 8.6.1 | 19 NOV 2021

Check for additions and updates to these release notes.

Release Versions

2

vRealize Orchestrator 8.6.1 | 19 November 2021

- vRealize Orchestrator 8.6.1 Appliance build 18937479
- VRealize Orchestrator 8.6.1 Update Repository build 18937479

Updates made to this document:

| Date | Description of update | Type |
|------------|---|------|
| 11/19/2021 | Initial publishing. | |
| 03/02/2022 | Added link to KB workaround used to resolve upgrade failure related to the log4j vulnerabilities. | |

IMPORTANT

Upgrade failure after performing steps in KB 87120

Performing the instructions used to address the CVE-2021-44228 and CVE-2021-45046 log4j vulnerabilities described in KB 87120 can cause upgrade failures for vRealize Automation and vRealize Orchestrator 8.6.2 or earlier. For a workaround, see [KB 87794](#).

What's New in vRealize Orchestrator 8.6.1

3

Newly supported Node.js and PowerCLI versions.

vRealize Orchestrator now supports the following versions of the Node.js and PowerCLI runtimes:

- Node.js version 14
- PowerCLI version 12.3.0 for PowerShell 7.1

Changes to cluster health and synchronization status monitoring.

In previous releases of vRealize Orchestrator, you could monitor the health and synchronization status of your cluster from the **Orchestrator Cluster Management** and **Validate Configuration** pages of the Control Center. You can no longer use these pages to monitor the status of your cluster.

The new recommended approach for monitoring the health and synchronization status of your cluster is to log in to the vRealize Orchestrator Client and navigate to the **System** tab of the dashboard. Alternatively, if the vRealize Orchestrator Client is unavailable, you can also check the status of your cluster by running the `kubectl get pods -n prelude` command from the vRealize Orchestrator Appliance command line.

Deploying the vRealize Orchestrator Appliance

4

The vRealize Orchestrator Appliance is a VMware Photon OS-based appliance distributed as an OVA file. It is prebuilt and preconfigured with an internal PostgreSQL database, and it can be deployed with vCenter Server 6.0 or later.

The vRealize Orchestrator Appliance is a fast, easy to use, and more affordable way to integrate the VMware cloud stack, including vRealize Automation and vCenter Server, with your IT processes and environment.

For instructions about deploying the vRealize Orchestrator Appliance, see [Download and Deploy the vRealize Orchestrator Appliance](#).

For information about configuring the vRealize Orchestrator Appliance server, see [Configuring a Standalone vRealize Orchestrator Server](#).

Upgrading and Migrating vRealize Orchestrator

5

You can upgrade a standalone or clustered vRealize Orchestrator 8.x deployment to the latest product version by using a mounted ISO image.

For more information about upgrading the vRealize Orchestrator Appliance, see [Upgrading vRealize Orchestrator](#).

You can migrate a standalone vRealize Orchestrator instance authenticated with vSphere or vRealize Automation to vRealize Orchestrator 8.6.1. Product versions of vRealize Orchestrator 7.x supported for migration include versions 7.3 to 7.6. The migration of clustered vRealize Orchestrator 7.x deployments is not supported.

For more information about migrating the vRealize Orchestrator Appliance, see [Migrating vRealize Orchestrator](#).

Plug-ins Installed with vRealize Orchestrator 8.6.1

6

The following plug-ins are installed by default with vRealize Orchestrator 8.6.1:

- vRealize Orchestrator vCenter Server Plug-In 7.0.0
- vRealize Orchestrator Mail Plug-In 8.0.0
- vRealize Orchestrator SQL Plug-In 1.1.8
- vRealize Orchestrator SSH Plug-In 7.3.0
- vRealize Orchestrator SOAP Plug-In 2.0.6
- vRealize Orchestrator HTTP-REST Plug-In 2.4.1
- vRealize Orchestrator Plug-In for Microsoft Active Directory 3.0.11
- vRealize Orchestrator AMQP Plug-In 1.0.6
- vRealize Orchestrator SNMP Plug-In 1.0.3
- vRealize Orchestrator PowerShell Plug-In 1.0.20
- vRealize Orchestrator Multi-Node Plug-In 8.6.1
- vRealize Orchestrator Dynamic Types 1.3.6
- vRealize Orchestrator vCloud Suite API (vAPI) Plug-In 7.5.2

Earlier Releases of vRealize Orchestrator

7

Features and issues from earlier releases of vRealize Orchestrator are described in the release notes for each release. To review release notes for earlier releases of vRealize Orchestrator, click one of the following links:

- [vRealize Orchestrator 8.6](#)
- [vRealize Orchestrator 8.5.1](#)
- [vRealize Orchestrator 8.5](#)
- [vRealize Orchestrator 8.4.2](#)
- [vRealize Orchestrator 8.4.1](#)
- [vRealize Orchestrator 8.4](#)
- [vRealize Orchestrator 8.3](#)
- [vRealize Orchestrator 8.2 Patch 1](#)
- [vRealize Orchestrator 8.2](#)
- [vRealize Orchestrator 8.1 Patch 3](#)
- [vRealize Orchestrator 8.1 Patch 2](#)
- [vRealize Orchestrator 8.1 Patch 1](#)
- [vRealize Orchestrator 8.1](#)
- [vRealize Orchestrator 8.0.1](#)
- [vRealize Orchestrator 8.0](#)
- [vRealize Orchestrator 7.6.0](#)
- [vRealize Orchestrator 7.5.0](#)
- [vRealize Orchestrator 7.4.0](#)

Resolved Issues

8

- **You encounter workflow regressions.**

The default **RequestConfig** object used after refactoring was unintentionally changed and this introduced compression by default, which, for example, causes the Apache HTTP Client to remove the **Content-Length** header from the response. Such differences might cause regressions in workflows.

- **After upgrading from vRealize Orchestrator 8.4.0 to vRealize Orchestrator 8.5.1, you are unable to load the vRealize Orchestrator Client when you are logged in as a non-default tenant but you can log in as as a default tenant.**

A subtenant is unable to use the embedded vRealize Orchestrator Client in vRealize Automation even if the client is registered as an integration for that particular vRealize Automation tenant. When attempting to log into the vRealize Orchestrator Client, subtenants receive the following error message: **O, Unknown error.**

- **After migrating from vRealize Orchestrator 7.6 to vRealize Orchestrator 8.6, the **vco-app** pod fails to start because of a SQL script failure.**

The **vco-app** pod fails to start because the **install_rpms init** container exits with a non zero status. The SQL script **merge_duplicates** fails with the following error:

```
[ERROR: column "createdat" does not exist  Where: PL/pgSQL function
merge_duplicates(text,text,text) line 105 at EXECUTESQL statement "SELECT
merge_duplicates('vmo_workflowcategory', 'vmo_workflow', 'workflowcategoryid')"PL/pgSQL
function inline_code_block line 4 at PERFORM]
```

- **HTTP-REST plug-in must handle HTTP header names in a case-insensitive way.**

Based on [RFC 2616](#), header field names must be treated as case-insensitive. Prior versions of the HTTP-REST plug-in made a distinction between headers with the same name, but with different casings and this can lead to unwanted results. The HTTP-REST plug-in packaged with vRealize Orchestrator 8.6.1 and above has the following methods refactored so they can work regardless of casing:

- **RESTRequest#setHeader**
- **RESTResponse#getHeaderValues**

- **RESTOperation#addMandatoryHeaderParameter**
- **RESTOperation#isHeaderParamMandatory**

The **RESTResponse#getAllHeaders** method still returns an ordinary dictionary with no guarantee for the casings of the headers. If you want to get a particular header that is case-insensitive, use **RESTResponse#getHeaderValues(headerName)** instead. Old host configurations created with previous versions of the HTTP-REST plug-in might still behave case-sensitively, particularly for the mandatory header parameters of REST operations.

- The following error message is displayed above the workflow input form when using the **RUN AGAIN** functionality to start a new workflow run: **Some data cannot be retrieved. If the problem persists, contact your system administrator.**

The value of an input field consisting of simple types of non-object arrays, such as **Array/number** or **Array/string**, is incorrectly evaluated and passed to an external source action that is used to populate the value of another input form field.

Known Issues

9

- **Pushing commits to a protected Git branch fails.**

If the configured Git branch is protected, the push operation fails consistently, but the message that appears indicates that the push is successful.

Workaround: The decision was taken not to fix this issue in the current or upcoming releases of vRealize Orchestrator. This known issue entry is going to be deleted from the release notes for the next release.

- **Local changes are not available after duplicating and deleting a workflow.**

You duplicate a workflow and then delete it. In the **Git History** page, there is no local change for the deleted workflow.

No workaround.

- The interaction form for resuming a failed workflow does not display fields for all defined variables when one of the variables is of the type **Properties**. Inputs for variables coming after the **Properties** type variable will be hidden. The input fields of a **Properties** type variable will not be populated.

The variable type was not interpreted for the **Properties** type, which caused issues during the input form rendering. The type is now used while generating the input form.

No workaround.

Previously Known Issues

10

This section contains previously known issues (known issues remaining from older releases of vRealize Orchestrator that still exist in the product).

- **During the installation of a plug-in in the vRealize Orchestrator Control Center, an error message appears.**

When you install a plug-in from the **Manage Plug-Ins** page in Control Center, the following error message appears: **Plug-in 'name_of_the_plug-in' (plug-in_file_name) is not compatible with the current platform version. Supported platform versions are 'names_of_the_supported_versions'.** Clicking on the 'Install' button will install it anyway.

Workaround: You can safely ignore this error and proceed with the installation of the plug-in.

- **The vRealize Orchestrator authentication configuration might become invalid, if the authentication provider certificate changes or regenerates.**

When the SSL certificate of the vRealize Automation or vSphere instance that is configured as the authentication provider in Control Center is changed or regenerated, the vRealize Orchestrator authentication configuration becomes invalid, and the vRealize Orchestrator server cannot start.

Workaround: Import the new authentication provider certificate:

- a Log in to Control Center as **root**.
- b Click **Certificates**.
- c Click the **Import on the Trusted Certificates** tab.
- d Load the SSL certificate from a URL or a file.
- e Click **Import**.

- **The SOAP plug-in cannot connect through an authenticated proxy server.**

When attempting to run the **Add a SOAP host** workflow, you encounter an issue with the proxy server authentication.

Workaround: When running the workflow, use a proxy server that does not require authentication.

- **If you experience issues connecting to a SOAP or a REST host, or importing a certificate, you might have to explicitly enable certain versions of SSL or TLS.**

For information about this issue, see [the Java Secure Socket Extension \(JSSE\) Reference Guide](#).

Workaround: For information about explicitly enabling SSLv3 and TLSv1 for outgoing HTTPS connections, see [How to enable TLSv1.1 and TLSv1.2 for outgoing HTTPS connections in vRO 6.0.x \(KB 2144316\)](#).

- **The SSH plug-in encounters firewall connectivity issues.**

The SSH plug-in cannot connect to a Cisco Adaptive Security Appliance (ASA) firewall.

Workaround: The SSH plug-in for vRealize Orchestrator 7.1 does not support connectivity to a Cisco Adaptive Security Appliance (ASA) firewall.

- **Problems handling non-ASCII characters in certain contexts.**

Using non-ASCII characters in input parameters results in incorrect behavior in the following situations:

- If you run the **SCP put** or **SCP get** workflows from the SSH folder on a file with a name that contains non-ASCII characters, the workflow runs but the name of the resulting file on the destination machine is unreadable.
- If you try to insert non-ASCII characters into variable names, the characters do not appear. This issue occurs for workflow variables and action variables.

No workaround.

- **The Storage VSAN workflows of the vCenter Server plug-in do not support adding Solid-State Drive (SSD) disks to an ESXi host.**

The **Add disks to disk group** and **Remove disks from disk groups** workflows do not support adding SSD disks as capacity disks to ESXi hosts.

No workaround.

- **The RESTOperation ID does not initialize properly if the REST host instance is created by using a Swagger spec.**

In the HTTP-REST plug-in, when the REST host instance is created by a Swagger spec, the **RESTOperation ID** does not initialize properly and the **getOperation** method of the **RESTHost** object does not work.

No workaround.

- Adding values to vCenter Server data object properties of the **Array** type is not possible.

When vRealize Orchestrator runs scripts, the vCenter Server plug-in converts JavaScript arrays to Java arrays of a fixed size. As a result, you cannot add new values to vCenter Server data objects that take arrays as property values. You can create an object that takes an array as a property if you instantiate that object by passing it a pre-filled array. However, after you instantiate the object, you cannot add values to the array.

For example, the following code does not work:

```
var spec = new VcVirtualMachineConfigSpec();
spec.deviceChange = [];
spec.deviceChange[0] = new VcVirtualDeviceConfigSpec();
System.log(spec.deviceChange[0]);
```

In the above code, vRealize Orchestrator converts the empty **spec.deviceChange** JavaScript array into the fixed-size Java array **VirtualDeviceConfigSpec[]** before it calls **setDeviceChange()**. When calling **spec.deviceChange[0] = new VcVirtualDeviceConfigSpec()**, vRealize Orchestrator calls **getDeviceChange()** and the array remains a fixed, empty Java array. Calling **spec.deviceChange.add()** results in the same behavior.

Workaround: Declare the array as a local variable:

```
var spec = new VcVirtualMachineConfigSpec();
var deviceSpec = [];
deviceSpec[0] = new VcVirtualDeviceConfigSpec();
spec.deviceChange = deviceSpec;
System.log(spec.deviceChange[0]);
```