

vSphere SDK for Perl 6.7 Release Notes

Released 17 Apr 2018

Build 8156551 is the 6.7 release of the vSphere SDK for Perl.

Check frequently for additions and updates to these release notes.

vSphere SDK for Perl 6.7 Release Notes

Welcome to the vSphere SDK for Perl 6.7 release notes.

The vSphere SDK for Perl provides an easy-to-use Perl scripting interface to the vSphere Web Services API.

See the *vSphere Command-Line Interface Release Notes* for information about vSphere CLI.

About the vSphere SDK for Perl

The vSphere SDK for Perl is a client-side Perl framework that provides an easy-to-use scripting interface to the vSphere Web Services SDK. Administrators and developers who are familiar with Perl can use the vSphere SDK for Perl to automate a wide variety of administrative, provisioning, and monitoring tasks in the vSphere environment. The vSphere SDK for Perl includes ready-to-use utility applications, which you can immediately put to use in your virtual data center.

The vSphere SDK for Perl installation includes the WS-Management Perl Library, which allows you to write scripts that retrieve CIM data from the ESX host using CIMOM, a service that provides standard CIM management functions over a WBEM (Web-Based Enterprise Management). The installation also includes the vSphere CLI command set.

You can use the vSphere SDK for Perl to manage ESXi 5.x, ESXi 6.x, vCenter Server 5.x, and vCenter Server 6.x.

What's New in vSphere SDK for Perl 6.7

OS support — This release adds support for Red Hat Enterprise Linux (RHEL) 7.3 (Server) 64-bit.

Linux installation — This release consolidates the Linux installation process. There is no separate installation process for RHEL platforms.

Updated libraries — This release includes updated libraries that add support for ESXi 6.7 and vCenter Server 6.7. If you have an earlier version installed on your system, you must replace the existing libraries with the new libraries.

Supported Platforms

For this release, vCLI is supported on the following Linux platforms:

- Red Hat Enterprise Linux (RHEL) 6.6 (Server) — 64-bit
- Red Hat Enterprise Linux (RHEL) 7.1 (Server) — 64-bit
- Red Hat Enterprise Linux (RHEL) 7.3 (Server) — 64-bit
- Ubuntu 12.04 (LTS) — 64-bit
- Ubuntu 14.04 (LTS) — 64-bit
- Ubuntu 15.10 (LTS) — 64-bit
- Ubuntu 16.04 (LTS) — 64-bit
- SLES 11 SP3 — 64-bit
- SLES 12 — 64-bit

For this release, vCLI is supported on the following Windows platforms:

- Windows 8 (64-bit)
- Windows 10 (64-bit)
- Windows 2008 (64-bit)
- Windows 2012 R2 (64-bit)

Resolved Issues

- **Running VMware vSphere Automation SDK for Perl 6.5 samples on Windows fails with an error**

If you try to run VMware vSphere Automation SDK for Perl 6.5 samples on Windows, you receive an error message of type `(pythondll) failed. The specified module could not be found.`

Known Issues

- **Obtaining objects of different types requires multiple calls to the `Vim::get_views()` subroutine**

The `Vim::get_views()` subroutine takes a reference to an array of managed object references and returns a reference to an array of view objects. Although the array can contain multiple managed object types, objects of only one type can be obtained at the same time.

Workaround: You can call `get_views()` multiple times to obtain multiple object types. Or you can specify the appropriate base class (rather than a specific subclass) as the view type to access values of properties that are common across all objects based on the superclass. For example, `view_type = 'ManagedEntity'` allows you to retrieve all objects that extend from this base class (`VirtualMachine`, `HostSystem`, `Folder`, `ResourcePool`, and so on).

- **Must use encoded representation when constructing filters that recognize names containing certain special characters**

Based on what the vSphere Web Services API returns, a few property values, such as `ManagedEntity.name`, `VirtualMachineConfigInfo.name`, and `VirtualMachineConfigSpec.name`, are returned with the characters '/', '\', and '%' encoded as %2f, %5c, and %25.

For example, if you have a Folder named `My%Folder`, its name is returned as `My%25Folder`. If you want to construct a filter that recognizes names containing slashes or percent signs, you must use the encoded representation.

Workaround: To match a name that contains a percent sign, write:

```
filter => { name => qr/%25/ }
```

You **cannot** perform a match on percent sign alone, because a percent sign alone also matches other characters that are preceded by a percent escape character.

```
filter => { name => qr/%/ } # wrong;
```

You must use case-insensitive matching for escape character sequences that contain alpha characters:

```
filter => { name => qr/%2f/i } # matches %2f and %2F
```