

VMware Fusion Tech Preview 2017: API/CLI Getting Started Guide

This document will provide step by step instruction for getting started with the new developer focused features available in the VMware Fusion Tech Preview 2017 release.

The developer features include a new RESTful API, an improved 'vmrun' command line interface, and integration with [Photon OS](#), VMware's Cloud Native OS platform.

Fusion API: What is it?

The VMware Fusion REST API (the "Fusion API" for short) is a new feature we are testing in the VMware Fusion Tech Preview 2017 release which provides a simple HTTP interface for interacting with the core VMware hypervisor and associated processes programmatically.

Users can send standard GET, PUT, POST, and DELETE messages via HTTP to control configuration and deployment options such as: Create, Clone, and Power operations; Networking operations like create and updating virtual NIC configuration, get IP addresses and configure Port Forwarding; and configuring Shared Folders (hgfs). Response payloads are delivered in standard JSON format.

How do I use it?

The Fusion API is a daemon ('vmrest') that runs by default on port 8697. The service can be invoked from the terminal by simply typing 'vmrest' (without quotes). To run the daemon as a service in the background to avoid using a terminal session, simply append '&' to the command. ('vmrest &') and hit 'return' once the process starts.

To kill the daemon, issue a 'ctrl-c' if the program is in the foreground, or 'sudo kill <pid>' if it is running in the background. ("pgrep vmrest" to find the PID.)

The Fusion API has a [Swagger](#) based interface allowing easy access to the full API command tree. Simply visit <http://localhost:8697/> once the service is running to get a complete list of available commands, the appropriate syntax, and a quick 'test it out' capability.

For example:

```
curl -X GET --header 'Accept: application/vnd.vmware.vmw.rest-v1+json' 'http://localhost:8697/api/vms'
```

This command pasted into a Terminal window would return a complete list of VM's in JSON format:

```
> curl -X GET --header 'Accept: application/vnd.vmware.vmw.rest-v1+json'  
'http://localhost:8697/api/vms'
```

```
[  
  {  
    "id": "8VKVH8",  
    "path": "/Users/mike/Virtual Machines.localized/Windows 10 x64.vmwarevm/Windows 10  
x64.vmx"  
  },  
  {  
    "id": "Z2ROLN",  
    "path": "/Users/mike/Virtual Machines.localized/Kali.vmwarevm/Kali.vmx"  
  },  
  {  
    "id": "WYVRPR",  
    "path": "/Users/mike/Virtual Machines.localized/macOS 10.13 - HFS+.vmwarevm/macOS  
10.13 - HFS+.vmx"  
  },  
  {  
    "id": "XQDAA3",  
    "path": "/Users/mike/mydev/vagrant-  
ubuntu/.vagrant/machines/default/vmware_fusion/946264dd-3123-4195-8879-  
3cd97c4b0894/vivid64.vmx"  
  },  
  {  
    "id": "1NUWT6A",  
    "path": "/Users/mike/Virtual Machines.localized/Windows 7 Ult - x64.vmwarevm/Windows 7  
Ult - x64.vmx"  
  }  
]
```

API Considerations

This feature is experimental and should not be used in production systems. In addition, there are several specific noteworthy deployment considerations:

- The vmrest service will run as the user who started it. 'sudo vmrest' would make the root user run the service, for example.
- The service does not provide authentication. Access to the network path where the API resides should be restricted if security is a concern at this time.
- HTTP requests are not encrypted.

Updated vmrun commands

vmrun is a useful command line utility for managing local virtual machines on Fusion. With the Tech Preview 2017 release we have added 12 new calls to the core hypervisor subsystem in the following areas:

- VM NIC operations
- Host network/port forwarding operations
- Photon OS Template VM update

Category	Command name	Description
VM NIC operations	vmrun listNetworkAdapter	list NIC devices for given vmx path
	vmrun addNetworkAdapter	add NIC devices for given vmx path
	vmrun setNetworkAdapter	update NIC devices for given vmx path
	vmrun deleteNetworkAdapter	delete NIC devices for given vmx path
Host network operations	vmrun listHostnetwork	list all networks in the host
	vmrun listPortForwarding	list port forwardings for given host network
	vmrun setPortForwarding	update port forwardings for given host network
	vmrun deletePortForwarding	delete port forwardings for given host network
Photon OS	vmrun downloadTemplateVM <path>	get Photon OS VM online and download it into local directory

Photon OS Integration

Photon OS is a minimal Linux container host, optimized to run on VMware platforms. It contains all necessary drivers and tools for host-guest file integration and deployment across other VMware infrastructure products.

Photon OS is:

- **Optimized for VMware** - Leveraging more than a decade of experience validating guest operating systems, Photon OS is thoroughly validated on vSphere, Workstation and Fusion; and, because of the underlying vSphere platform, we're able to highly tune the Photon OS kernel for VMware product and provider platforms.
- **Purpose Built for Containers** - Compatible with container runtimes, like Docker, and container scheduling frameworks, like Kubernetes.
- **Efficient lifecycle management** - Contains a new, open-source, yum-compatible package manager - `dnf` - that makes the system as small as possible, but preserves robust yum package management capabilities.

Photon OS is released as Open Source, and is available on our github page. For more info or for Photon OS on other platforms such as AWS or Google Compute Engine, visit <https://vmware.github.io/photon/>

`vmrun 'downloadTemplateVM'` example:

```
> vmrun downloadTemplateVM ~/photon/
```

```
Download Photon OS VM to /Users/mike/photon/photon.ova
Opening OVA source: /Users/mike/photon/photon.ova
91.5 MB of 291.5 MB
The manifest validates
Source is signed and the certificate validates
Opening VMX target: /Users/mike/photon
Writing VMX file: /Users/mike/photon/photon-custom.vmwarevm/photon-custom.vmx
Transfer Completed
Completed successfully
Download template VM into /Users/mike/photon successfully
```

(There is currently a known issue where the `downloadTemplateVM` command does not support file paths with spaces.)