



Customizing vm-support in vSphere™ 5.0

TECHNICAL WHITE PAPER

Table of Contents

The vm-support tool - An Overview	2
The vm-support tool in vSphere™ 5.0	3
Introduction:.....	3
Exploring groups, manifests and tags	3
Groups	4
Manifest files	4
Exploring the Manifest file	6
Additional features.....	11
Additional Documentation	12
Providing Feedback	12



The vm-support tool - An Overview

The vm-support is an intelligent and useful tool provided by VMware™ to gather diagnostics information, troubleshoot issues, and understand the setup from the support side. The vm-support tool has been bundled with ESX® since ESX® 2.5.

vm-support is a simple script which gathers required logs from various files, core-dump if present, and information on the state of the virtual machines.

The vm-support tool works using two steps:

- i) Collecting the diagnostic logs from various locations. The support tool will collect logs from different components like vmkernel™, host, CIM, virtual machines, security, vpxa, cronjobs, dmesg, update logs and many other diagnostic logs which are helpful in understanding the state of the system.
- ii) Collecting configuration information of the NICs, switches, storage adapters, NAS mounts, multi-path setup, and many other details.

To obtain such data, the vm-support tool triggers a list of commands which obtain the required results and stores them in the respective file under the vm-support bundle.

For Example: /usr/sbin/esxcfg-swiscsi command is triggered to retrieve details about the software iscsi configuration and the output is stored in the esxcfg-swiscsi.txt of the vm-support bundle.

The main objective of running a vm-support is troubleshooting a system which might have faced unresponsiveness, different behavior than expected and for collecting performance snapshots. From a support side, these logs provide adequate information to debug and resolve the server failure.

The vm-support tool is a simple shell script from ESX 2.5® release and has been modified to support new platforms, servers, additional commands that are introduced with new releases of ESX/ESXi. While vm-support works on both Classic and Embedded versions of ESX®, the commands triggered and logs collected vary depending upon the release.

Once completed, vm-support provides a tar ball bundle that states the date and time at which the script was run. This tar file contains the diagnostic logs and the core-dumps for the files to be traversed by in an easier method.

An important feature of the vm-support tool is that of running snapshots. The snapshots here do not refer to the snapshot feature available to a virtual machine. While running vm-support in snapshot mode, it gathers data from key areas like network packets, memory utilization, CPU allocation and disk usage.

On getting the tar bundle from such method, support personnel can replay the vm-support, observe the values being played back and conclude on a possible root cause of an issue.

The vm-support tool can be used with the command line options to manipulate the logs that are needed.

For Example: vm-support -x lists out the world ids for running VMs while -X grabs the debug information of a hung VM. A list of these options can be found by vm-support -h

This paper explains the new features of vm-support in vSphere™ 5.0 which helps the user to customize and use the tool as per their requirement.

The vm-support tool in vSphere™ 5.0

Introduction:

The vm-support tool is re-written entirely in python framework from vSphere™ 5.0. This allows for a plugin type framework and users can run individual modules without the need to change the tool.

As explained in the tool overview in the preceding section, the vm-support tool gathers and triggers commands to get diagnostic data across different components of the ESX® server. The vm-support tool has been modified with this in mind and now addresses these components as “Groups”. Almost every component comes under this group and allows the user to run the support tool on these groups either individually or by combining multiple groups.

vm-support tool in vSphere™ 5.0 introduces the concept of manifest files. A manifest file is the setup or configuration file that the vm-support tool refers to run any command or gather any diagnostic log and store it in the vm-support tar bundle. A manifest file has the details of the group it belongs to, the set of actions that have to be executed and additional options like timeout value in executing a command. By default, a set of manifest files are made available to the user during the installation of ESXi® 5.0. The manifest files are available under the /etc/vmware/vm-support directory.

Tags are another feature introduced in the vm-support tool in vSphere™ 5.0. Tags refer to a number of groups or a list of manifest files with common components and trigger commands around similar groups.

Like previous versions, the support tool also provides the user to run performance snapshots, gather information about a particular VM, and also list out the files that have been gathered by the support tool.

Exploring groups, manifests and tags

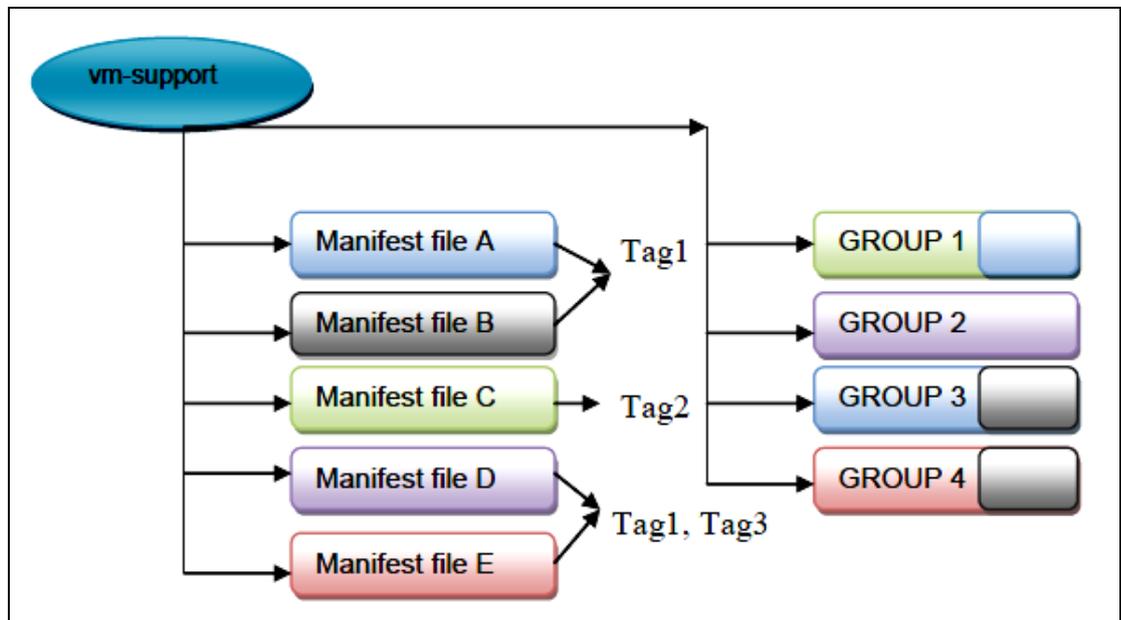


Figure 1.1: Illustrating the vm-support components in vSphere 5.0

Groups

Groups are the different components of an ESX[®] system and help to identify the nature of a manifest file. By default, a set of 19 groups are made available to the user.

These groups can be listed by executing `vm-support -g`

- ActiveDirectory
- CIM
- Configuration
- Crash
- Fault
- FileSystem
- Hardware
- HungVM
- Installer
- IntegrityChecks
- Logs
- Network
- Performance Snapshot
- Storage
- System
- Testing
- Userworld
- Virtual
- Host Profiles

The above groups cover the overall components in ESXi[®] server and the users can set their customized manifest file by declaring it to one of the groups. New groups can be created and added to the manifest file helping engineers to run their specific Group.

The vm-support tool can be triggered by the individual group level. This helps the user in troubleshooting specific areas. For example, if a user is facing an issue with networking in the server, the user can trigger vm-support only to run on the manifest files that are included in the group "Network"
`vm-support -g Network`

Likewise, the support tool can be triggered for storage, CIM, etc. based upon the diagnosis needed for troubleshooting. Executing multiple groups by triggering the vm-support tool once is currently not supported.

Manifest files

The manifest files are the key configuration/setup files which the vm-support tool refers during execution. The manifest files explain the nature of the commands the tool has to execute, the logs that are to be collected, time out value while executing a command, and error handling if a command or log collection has encountered an error.

With the default installation of ESXi[®] 5.0 the users are given a set of manifest files which comprise all the components and key areas for collection of diagnostic logs to be populated in the vm-support tar bundle.

These manifest files are located under `/etc/vmware/vm-support` and have the extension of `.mfx`.

The default manifests available can be listed by

```
vm-support --listmanifests
```

<ul style="list-style-type: none"> • <code>ad.mfx</code> • <code>base.mfx</code> • <code>cim.mfx</code> • <code>config-system.mfx</code> • <code>esxssoftware.mfx</code> • <code>fault-tolerance.mfx</code> • <code>filesystem.mfx</code> • <code>hardware.mfx</code> • <code>hostProfiles.mfx</code> • <code>hostd.mfx</code> • <code>installer.mfx</code> • <code>md5sums.mfx</code> • <code>modules.mfx</code> • <code>multipath.mfx</code> • <code>network.mfx</code> • <code>networking-dvs.mfx</code> • <code>networking-fence.mfx</code> • <code>networking-tcpip.mfx</code> • <code>performance.mfx</code> • <code>resource.mfx</code> • <code>storage-fcoe.mfx</code> 	<ul style="list-style-type: none"> • <code>storage-iscsi.mfx</code> • <code>storage-nas.mfx</code> • <code>storage.mfx</code> • <code>system-logs.mfx</code> • <code>test-esx.mfx</code> • <code>usb.mfx</code> • <code>userworld-cores.mfx</code> • <code>userworld.mfx</code> • <code>vm-coredump.mfx</code> • <code>vm-cores.mfx</code> • <code>vm-disks.mfx</code> • <code>vm-logs.mfx</code> • <code>vm-nmi.mfx</code> • <code>vm-snapshots.mfx</code> • <code>vm-stats.mfx</code> • <code>vm-suspend.mfx</code> • <code>vm.mfx</code> • <code>vmfs-header.mfx</code> • <code>vmkernel-cores.mfx</code> • <code>vmkernel-old-cores.mfx</code> • <code>vmkernel.mfx</code>
---	---

While running `vm-support` without any additional command line options, the `vm-support` tool executes the manifest files that are enabled by default. The manifest files load the commands to be triggered to the `vm-support@` tool and gathers the output in the same format. This helps the user diagnose the output of a particular command.

`vm-support` tool customizations are heavily dependent on manifest files. To customize a `vm-support@`, it is mandatory that these manifest files are correctly configured and appropriately tagged and categorized.

Exploring the Manifest file

A typical manifest file will have the following fields that explain its nature:

For example: Consider the example of storage.mfx file:

```
cat /etc/vmware/vm-support/storage.mfx
```

- Manifest name: base
- Manifest group: Storage
- Manifest default: Enabled
- Manifest tag: storage
- Action Options file/command

The manifest name assigns a name to the manifest file that can be used across multiple manifest files. The manifest file is unique and is identified by the file name to which it has been created. In this case, the storage.mfx is the identifier for the manifest file.

Group is the next important field for creating a manifest file. It tells the vm-support tool that this manifest file is categorized under a particular group and the file needs to be called when the vm-support tool is executed by a Group name.

The Enable/Disable field ensures whether this manifest file is indeed authorized to be executed during run-time of the support tool. If a manifest file is set to Disabled, then the support tool ignores the manifest file and moves on to the next one. While running vm-support, without any additional command line options, the vm-support tool executes the manifest files that are set to be enabled.

The tag field assists the user to categorize multiple manifest files while running the vm-support tool based on the tag fields. Multiple tags can be provided to a manifest file which allows the same manifest file to be used by the engineer to have an idea of the components that it comprises.

For example, considering the vm.mfx manifest file,

```
# cat vm.mfx
% Manifest name: base
% Manifest group: Virtual Machines
% Manifest default: Enabled
% Manifest tag: vm,logs
# action Options file/command
```

In the above example, a manifest file uses multiple tags to be categorized under different components to identify its area of execution. Also, a manifest file can be used to address one particular component. Considering the userworld.mfx manifest file,

```
# cat userworld.mfx
% Manifest name: Process Information
% Manifest group: Userworld
% Manifest default: Enabled
% Manifest tag: userworld
# action Options file/command
run IGNORE_ERROR ps -cPTgjst
run - /sbin/memstats -r uw-stats
```

The last field “action Options file/command” contains the actual set of commands to be executed and the corresponding logs to be collected for diagnosis. The vm-support[®] tool provides the user pre-defined commands that can be used to trigger system commands or to gather any logs.

The set of pre-defined commands available to the users from the manifest files are:

- Copy

Copies the mentioned file to the bundle and has options like IGNORE_MISSING, IGNORE_ERROR, RECURSE, TIMEOUT=N[Seconds] to be used with the command.

```
Copy RECURSE, IGNORE_MISSING, IGNORE_ERROR /etc/vmware/
```

- Prune

Prunes some common files that have sensitive information

```
prune - /etc/vmware/vmkiscsid/.#vmkiscsid.db
```

- Run

Executes local cli commands or supported system commands in the ESXi server and has multiple options like IGNORE_ERROR, TIMEOUT=N[Seconds], RELOCATE, DUMP_JSON, IGNORE_MISSING

```
run IGNORE_ERROR, TIMEOUT=10 fdisk -lu
run DUMP_JSON /usr/sbin/localcli storage core adapter
```

- Safecopy

Allows the user to copy the contents of a file and not to obtain a lock on it. While the file being used by the system and changes dynamically. The options available for this command are RECURSE, IGNORE_MISSING, IGNORE_ERROR.

```
safecopy RECURSE /proc/*
```

- Vmcmd, vmfile

Trigger a VM related command and to collect a file residing in the VM's directory. It is advisable to use the IGNORE_ERROR, option to these commands, as there could be vmfs locks on the VM files.

```
vmfile IGNORE_ERROR, IGNORE_MISSING *.vmtx
vmcmd - /bin/ls -la "{VMDIR}"
```

- Vmfscmd

Used for triggering the vmkfstools command and has options like IGNORE_ERROR, TIMEOUT=N[Seconds]

```
vmfscmd TIMEOUT=10, IGNORE_ERROR vmkfstools -P -v 10
"/vmfs/volumes/{VMFS}"
```



Creating a custom manifest file

A customized manifest file helps a support personnel or an engineer to extract the diagnostic logs from the specified components instead of having to browse through the entire vm-support bundle which has logs from components that might not be required at the time of troubleshooting.

In a production environment, if the engineers detect low performance of the storage read/writes, then while debugging, they can trigger a custom created manifest file that will allow them to include storage, hardware, VM and kernel logs.

These logs would be sufficient for debugging purpose, when compared to the debugging of the entire vm-support bundle that has gathered logs on ActiveDirectory™, CIM, host profiles or Installers and many other components that are not required at the particular time.

The customization of a manifest file allows the engineer to combine multiple commands that they require in one manifest file. This covers the operation state of the server in one manifest file and can be used across components.

Any command that is not collected by the default manifest files can be added to the existing manifest file. Additionally, it can also be added to the set of new commands, which the engineers feel is required can be added to a manifest file and executed with the vm-support tool.

The following section will cover the process of creating a custom manifest file with the options explained in the “Exploring the Manifest file” section.

For the following example, a custom manifest file is created that incorporates a custom list of commands across different manifest files and new commands and logs.

```
cd /etc/vmware/vm-support
cat CustomManifest.mfx
% Manifest name: CustomManifest
% Manifest group: Custom
% Manifest default: Enabled
% Manifest tag: storage, base, network, filesystem, esxsoftware
# action Options file/command
run IGNORE_ERROR,TIMEOUT=10 dmesg
run IGNORE_ERROR,TIMEOUT=10 vdf -h
safecopy IGNORE_MISSING /vmfs/devices/char/vmkdriver/usbdevices
run IGNORE_ERROR /usr/lib/vmware/vm-support/bin/nicinfo.sh
run DUMP_JSON localcli software vib get
copy IGNORE_MISSING /var/log/*.*
safecopy IGNORE_MISSING /etc/vmware/esx.conf
run DUMP_JSON /usr/sbin/localcli network firewall get
run DUMP_JSON /usr/sbin/localcli storage vmfs snapshot list
run DUMP_JSON /usr/sbin/localcli storage core adapter list
run DUMP_JSON /usr/sbin/localcli storage core device vaai status get
run DUMP_JSON /usr/sbin/localcli storage core plugin registration list
run TIMEOUT=10 /usr/lib/vmware/vm-support/bin/swfw.sh
```



In the above custom created manifest file, commands of various groups like storage, network, system, software level are combined and the logs of different components are also collected under one execution of the manifest file. With the above manifest file set to enabled, the vm-support tool can be triggered to execute the created manifest file alone by `vm-support -a Custom:CustomManifest` or triggered by the custom group created by

```
vm-support -g Custom
```

The output of the vm-support tar bundle contains the logs of the respective commands that have been included in the manifest file and any errors that have been encountered while executing the commands. With the introduction of creating custom manifest file, the support personnel can narrow down the requirement of logs to the corresponding components. This will be very helpful in increasing the logs of any commands required by focusing on the exact error messages or behavior faced by the customers.

The manifest file created above will also be available from a vSphere™ client for exporting the system logs. [Fig 1.2]

From a VI Client, select File, select Export and select Export logs.

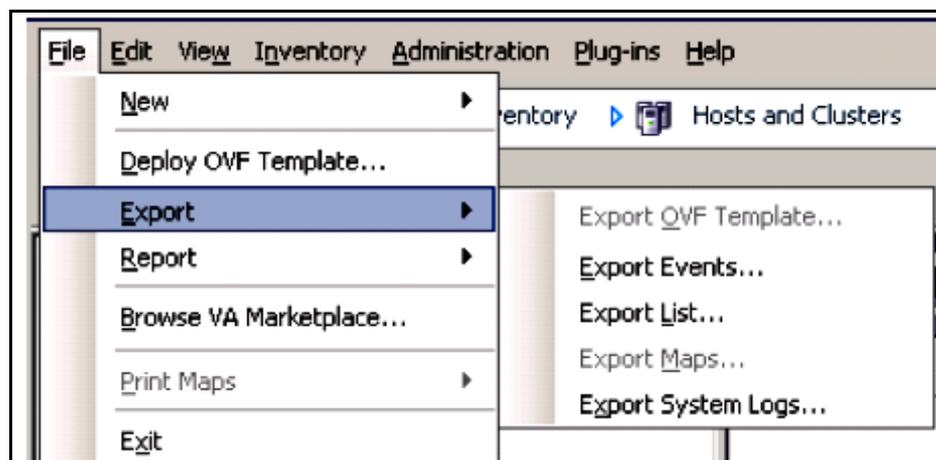


Figure 1.2a From a VI client, export options being displayed

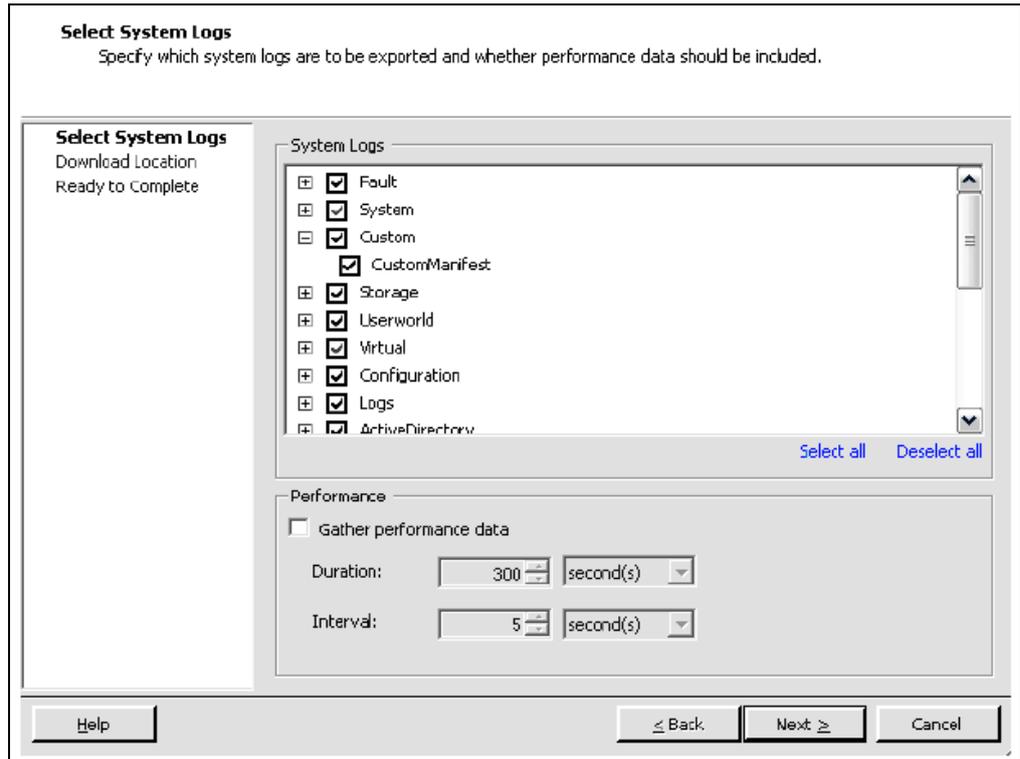


Figure 1.3b: From a VI client, export options being displayed

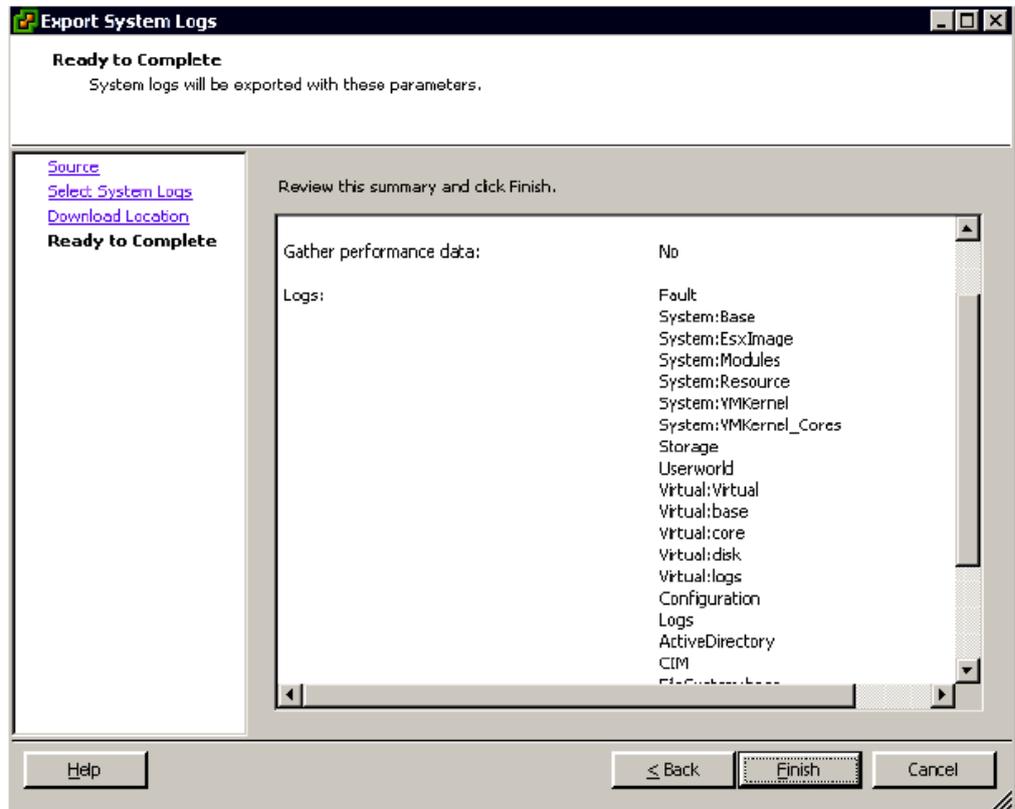


Figure 1.4c: Indicating the logs to be collected from the selected manifests.

Additional features

Apart from providing users to create customized vm-support execution, vSphere™ 5 also brings in two new features that help engineers troubleshoot and identify root-cause issues.

- Browser support of diagnostic logs
- Reconstruct.sh file

With a browser support, engineers can obtain exhaustive information of the system by utilizing the inbuilt .cgi files similar to:

- esxcfg-info.cgi
- esxcli.cgi
- vm-support.cgi

These cgi's can be accessed from a browser by the following path;

- <https://<ESXIP or HOSTNAME>/cgi-bin/esxcfg-info.cgi> or
- <https://<ESXIP or HOSTNAME>/vm-support.cgi/>

Another important script for debugging or recreating the system with the collected data is the reconstruct.sh file in vSphere™ 5.0. This script file is created in the root directory of the support bundle.

Certain commands in vm-support generate a large file that consumes more resources and is likely to result in a timeout error or takes a considerable amount of time to execute. To control the creation of larger files, the reconstruct.sh file breaks down the larger file into fragments when added into the vm-support bundle. Upon completion of the support tool, users can recreate the larger file by executing the reconstruct.sh file above the extracted bundle directory.



Additional Documentation

For more information about VMware™ vSphere™ 5, visit the product pages at

- [Product Overview](#)
- [Product Documentation](#)

Below are some of the links from VMware™ knowledge base articles related to vm-support:

- [Collection of diagnostic logs using vSphere™ Client](#)
 - [Collecting Performance snapshots using vm-support](#)
 - [Collecting diagnostic logs using vm-support® command](#)
-

Providing Feedback

VMware appreciates your feedback on the material included in this guide, and in particular, would be grateful for any guidance on the following topics:

1. How useful was the information in this guide?
2. What other specific topics would you like to see covered?

Please send your feedback to feedback@vmware.com, with “Customizing vm-support in vSphere™ 5.0” in the subject line.

