



VMware® Ruby vSphere Console Command Reference for Virtual SAN

Joe Cook and Cormac Hogan
Storage and Availability Business Unit
VMware
v 1.0a/March 2015

PREFACE	3
ABSTRACT	3
INTENDED AUDIENCE	3
OVERVIEW	4
BACKGROUND.....	4
FEATURES	4
ADVANTAGES.....	4
SETUP, CONFIGURATION, AND ACCESS	5
SETUP AND CONFIGURATION.....	5
ACCESSING AND LOGGING IN	6
LOGIN CREDENTIAL FORMATS	7
USAGE	8
RVC COMMAND STRUCTURE.....	8
TAB-COMPLETION	8
WILDCARDS	8
MARKS.....	8
NAVIGATING THE vSPHERE AND VIRTUAL SAN INFRASTRUCTURE	9
RVC COMMAND BASICS.....	11
VIRTUAL SAN RVC COMMANDS OVERVIEW	14
DETAILED VIRTUAL SAN RVC COMMANDS	18
VSAN.APPLY_LICENSE_TO_CLUSTER	18
VSAN.CHECK_LIMITS.....	19
VSAN.CHECK_STATE.....	21
VSAN.CLEAR_DISKS_CACHE.....	23
VSAN.CLUSTER_CHANGE_AUTOCLAIM	24
VSAN.CLUSTER_CHANGE_CHECKSUM.....	25
VSAN.CLUSTER_INFO	26
VSAN.CLUSTER_SET_DEFAULT_POLICY.....	28
VSAN.CMMDS_FIND	30
VSAN.DISABLE_VSAN_ON_CLUSTER.....	32
VSAN.DISKS_INFO	33
VSAN.DISK_OBJECT_INFO	35
VSAN.DISKS_STATS.....	37
VSAN.ENABLE_VSAN_ON_CLUSTER.....	39
VSAN.ENTER_MAINTENANCE_MODE.....	40
VSAN.FIX_RENAMED_VMS	42
VSAN.HOST_CLAIM_DISKS_DIFFERENTLY.....	43
VSAN.HOST_CONSUME_DISKS.....	44
VSAN.HOST_EVACUATE_DATA.....	45
VSAN.HOST_EXIT_EVACUATION	46
VSAN.HOST_INFO.....	47
VSAN.HOST_WIPE_NON_VSAN_DISKS.....	48
VSAN.HOST_WIPE_VSAN_DISKS.....	49
VSAN.LLDPNETMAP	51
VSAN.OBJECT_STATUS_REPORT	52

VSAN.OBJECT_INFO	54
VSAN.OBJECT_RECONFIGURE.....	56
VSAN.OBSERVER.....	57
VSAN.OBSERVER_PROCESS_STATSFILE	58
VSAN.PROACTIVE_REBALANCE	59
VSAN.PROACTIVE_REBALANCE_INFO	61
VSAN.PURGE_INACCESSIBLE_VSWP_OBJECTS	63
VSAN.REAPPLY_VSAN_VMKNIC_CONFIG	64
VSAN.RECOVER_SPBM.....	66
VSAN.RESYNC_DASHBOARD	68
VSAN.SCRUBBER_INFO.....	69
VSAN.SUPPORT_INFORMATION	70
VSAN.V2_ONDISK_UPGRADE.....	71
VSAN.VM_OBJECT_INFO	74
VSAN.VM_PERF_STATS	76
VSAN.VMDK_STATS	77
VSAN.WHATIF_HOST_FAILURES.....	78
REFERENCE	80

Preface

Abstract

This document details the vSphere Ruby Console, its benefits and usage in Virtual SAN environments.

Intended Audience

This white paper is intended for vSphere architects, administrators, developers and any others who are interested in deploying, managing or maintaining a Virtual SAN infrastructure. To glean the most out of this document, it will help to be familiar with vSphere infrastructure, Virtual SAN hardware as well as VM provisioning workflows.

Overview

Background

The **Ruby vSphere Console (RVC)** is an interactive command-line console user interface for VMware vSphere and Virtual Center. The Ruby vSphere Console is based on the popular *RbVmomi Ruby interface to the vSphere API* has been an open source project for the past 2-3 years. RbVmomi was created with the goal to dramatically decrease the amount of coding required to perform routine tasks, as well as increase the efficiency of task execution, all while still allowing for the full power of the API when needed.

The Ruby vSphere Console comes bundled with both the vCenter Server Appliance (VCSA) and the Windows version of vCenter Server. RVC is quickly becoming one of the primary tools for managing and troubleshooting Virtual SAN environments.

Features

RVC has a lot of the capabilities you'd expect from a modern command-line interface.

- Tab Completion
- Wildcards
- Marks
- Ruby Mode
- Python Mode
- VMODL Introspection
- Multiple Connections

Extensibility

- Single line Ruby scripts
- Use Cases and Advantages of RVC
- Virtual SAN Functionality covered
- Configuration of VSAN and Storage Policies
- Monitoring/ and Troubleshooting commands
- Performance monitoring via VSAN Observer

Advantages

- More detailed Virtual SAN insights vSphere Web Client
- Cluster view of VSAN while esxcli can only offer host perspective
- Mass operations via wildcards
- Works against ESX host directly, even if VC is down

Setup, Configuration, and Access

Setup and Configuration

The Ruby vSphere Console is free of charge and comes bundled with both the vCenter Server Appliance (VCSA) and vCenter Server for Windows. We recommend deploying a vCenter Server Appliance (minimum version 5.5u2) to act as a dedicated server for the Ruby vSphere Console and Virtual SAN Observer. This will mitigate any potential performance or security issues from the primary production vCenter Server.

To begin using the Ruby vSphere Console to manage your vSphere infrastructure, simply deploy the vCenter Server Appliance and configure network connectivity for the appliance. Afterwards, SSH to the dedicated vCenter Server Appliance and login as a privileged user. No additional configuration is required to begin.

Note: In light of the recommendation to leverage a dedicated vCenter Server Appliance for the Ruby vSphere Console and Virtual SAN Observer, we will use this recommendation as the context for the rest of this document.

[VMware KB Article 2007619](#) provides a walkthrough on deploying the vCenter Server Appliance 5.x/6.x:

Accessing and Logging In

Below you will find the steps to login and begin using the Ruby vSphere Console (RVC):

1. SSH to the VCSA dedicated for RVC and Virtual SAN Observer usage.

```
login as: root
VMware vCenter Server Appliance
root@192.168.1.99's password:
Last login: Thu Dec 22 22:29:15 UTC 2014 from 192.168.2.2 on ssh
```

2. Login to the VCSA as a privileged OS user (e.g. root or custom privileged user).
3. Login to RVC using a privileged user from vCenter.

Syntax: `rvc [options] [username[:password]@]hostname`

Login Example:

IP ADDRESS	DESCRIPTION
192.168.2.2	Workstation
192.168.1.99	Dedicated VCSA for RVC and Virtual SAN Observer
192.168.1.100	Primary vCenter Server

```
vcsa:~ # rvc root@192.168.1.100
password:
0 /
1 192.168.1.100/
```

Login Credential Formats

RVC credentials are directly related to the default domain setting in SSO (Single Sign-On). Verify the default SSO Identity Source is set to the desired entity.

Validate current Identity Source

Below you will find the steps to validate the default SSO Identity Source in the vCenter Web Client:

1. Login to the web client as **administrator@vsphere.local**
2. Navigate to Administration > Single Sign-On > Configuration > Identity Sources
3. There you will see entries for the
 - vsphere.local domain
 - Local Operating System
 - Active Directory (if configured)
4. One of these will be the **default**, and this has a direct bearing on which administrator password should be passed to RVC when attempting to login. Set the default to **vsphere.local** to use the **administrator@vsphere.local** credentials.

Note: *Using version 5.5u2 or higher of vCenter Server is recommended for both the vCenter Server Appliance and the Windows implementation.*

Usage

RVC Command Structure

RVC commands exist as Ruby modules within the RVC software. On the vCenter Server Appliance (VCSA), these modules can be found in the directory path `/opt/vmware/rvc/lib/rvc/modules`. Custom RVC commands can be created using the Ruby programming language. Once the module is created, upload it to the RVC modules directory, and then login to RVC to begin using the new, custom commands.

RVC commands are in the form of `<namespace>.<command>` where `<command>` is a function within the Ruby module named `<namespace>`. For instance, “`vsan.rb`” is a Ruby module within the RVC software and is located in “`/opt/vmware/rvc/lib/rvc/modules`”.

The command “`vsan.enable_vsan_on_cluster`” refers to the “`enable_vsan_on_cluster`” function within the `vsan.rb` Ruby module. All RVC namespaces, exist as separate Ruby modules wherein the RVC commands are located as separate Ruby functions within the Ruby Modules.

Tab-completion

Commands and paths can be tab completed as is typical in most command line interfaces. Whitespace characters will need to be escaped with a backslash.

Wildcards

Many commands such as “`vm.on`” can operate on multiple objects at once. RVC supports simple globbing (pattern matching based on wildcard characters) using “`*`” as well as advanced regular expression (regex) syntax. To use a regex, prefix the path element with “`%`”. For example: “`vm.on myvms/%^(linux|windows)`” will power on VMs whose names start with “`linux`” or “`windows`” in the “`myvms`” folder. It is necessary to explicitly anchor the pattern with “`^`” and “`$`” if you wish to match the whole path element.

Marks

RVC allows the use of marks, which can then be referenced by placing a tilde ‘`~`’ in front of the variable to reference it. For example, on can mark a path with `x` as shown here: “`mark x ~/computer/<foo>`” and then use `~x` to reference that path as follows: “`vsan.observer ~x --run-webserver --force`”.

Navigating the vSphere and Virtual SAN infrastructure

The vSphere and Virtual SAN infrastructure is presented to the user as a virtual file system that can be navigated with traditional directory listing (`ls`) and change directory (`cd`) commands. This virtual file system mirrors the hierarchy of the vSphere infrastructure and allows RVC commands to be issued on each of the manageable entities and their individual components (i.e. vCenter, Datacenter, Cluster, Storage, Hosts, Networks, Datastores, VMs).

In the example below, we have completed the steps detailed above and logged in via SSH to the dedicated vCenter Server Appliance (VCSA). We then launched RVC from the dedicated VCSA and pointed it to the production vCenter server as a data source. Afterwards we issued the `ls` command to obtain a directory listing from the root directory. Below we see a `/` for the root directory and then `192.168.1.100/` illustrating the directory for our production vCenter server.

```
> ls
0 /
1 192.168.1.100/
```

RVC can even connect to more than one vCenter or vSphere host at the same time. There are two options to connect to multiple servers. First, if you are outside of the RVC shell, simply add more hosts to the command line and separate them by a space (e.g. `rvc root@192.168.1.100 root@192.168.1.101`). If you are inside of the RVC shell use the command `connect`. Each server connection will be represented as a top-level node in the virtual file system. This will allow you to interact with multiple environments simultaneously, which can be useful for reporting, comparing configurations, etc.

Remember that the RVC command is looking to connect to an active vCenter installation so if you deploy a dedicated VCSA for RVC and Virtual SAN Observer management and follow the suggestion of not configuring vCenter, the command `rvc root@localhost` will fail with errors because there is no active vCenter server.

Note: Depending upon your security requirements, you can also avoid the password prompt by including the password in your connect command (e.g. `rvc root:vmware@localhost root:vmware@192.168.1.100`).

```
vcsa:~ # rvc root:vmware@localhost root:vmware@192.168.1.100
Connecting to localhost...
Connecting to 192.168.1.100...
0 /
1 localhost/
2 192.168.1.100/
```

Notice that each line of output in the directory listing above contains a single digit number to the left of the item listed almost as though it is a line number. This

number actually serves as a variable that can be used to issue commands against rather than requiring the full line item text to be entered. In the example below, instead of typing out “cd 192.168.1.100” to change directories and drill down into that vCenter, an administrator can simply type “cd 1” and achieved the same results. This can quickly speed up interacting with RVC as it significantly cuts down the number of keystrokes required while it also cuts down on the number of *fat finger typing errors*.

```
> cd 1
/192.168.1.100> ls
0 vsanDC (datacenter)
```

The vSphere and Virtual SAN infrastructure is presented to the user as a virtual file system that can be navigated with traditional directory listing (`ls`) and change directory (`cd`) commands. These commands are actually functions within the RVC Ruby module “basic” and in their formal form would be “basic.ls” and “basic.cd”. As these are very common commands, they have been aliased to “ls” and “cd” for ease of use. In the example below you will see how to drill down through the vSphere and Virtual SAN infrastructure to begin interacting with their manageable components.

```
> ls
0 /
1 192.168.1.100/
> cd 1
/192.168.1.100> ls
0 vsanDC (datacenter)
/192.168.1.100> cd 0
/192.168.1.100/lab> ls
0 storage/
1 computers [host]/
2 networks [network]/
3 datastores [datastore]/
4 vms [vm]/
```

The RVC virtual file system mirrors the hierarchy of the vSphere infrastructure. It displays the vSphere infrastructure as **vCenter** ⇒ **Datacenter** ⇒ (**Storage, Hosts, Networks, Datastores, VMs**).

The vSphere environment is broken up into 5 areas:

- **Storage:** vSphere Storage Profiles
- **Computers:** ESXi Hosts
- **Networks:** Networks and network components
- **Datastores:** Datastores and datastore components
- **VMs:** Virtual Machines and virtual machine components

RVC Command Basics

In the 1.8.0 release of RVC there are multiple Namespaces and multiple commands built-in to interact with the vSphere managed entities. Below are a few initial commands that can help get you started using RVC. To obtain a full listing of namespaces, type “help” at the RVC command line. To obtain a full listing of commands within a particular namespace, type “help <namespace>” at the RVC command line.

```
> help
Namespaces:
basic
vm
device ...
```

All RVC commands exist in Ruby modules and some of the more commonly used commands may have aliases. For example, the command to power off a VM is “**vm.off**”. This command exists within the Ruby module named “**vm**”. Since it is a rather common operation, it has been aliased to simply “**off**” for the sake of brevity.

Namespace: Basic - “basic.info” and “basic.show”

The “basic.info” and “basic.show” commands are aliased as “info” and “show” respectively. These are great commands to use when you would like to get a simple overview of a managed entity.

Viewing SPBM Information: “show vmprofiles”

In our example environment, there are 2 Storage Policies configured. In the example below, We see the “ls” and “show” commands to list all of the Storage Policies we have configured in this environment.

Note: When listing the contents of the `vmprofiles` directory, it may be necessary to include the trailing “/” character in order for the results to display. Without it you may see the “*Errno::EPIPE: Broken pipe*” message. Keep this in mind when navigating through the vSphere infrastructure using the RVC shell.

```
/192.168.100.1/vsanDC/storage> ls vmprofiles/
0 FTT1
1 FTT2

/192.168.100.1/vsanDC/storage/vmprofiles> show 0
path: /192.168.100.1/vsanDC/storage/vmprofiles/FTT1
class: RbVmomi::PBM::PbmCapabilityProfile
Name: FTT1
Description:
ProfileId: 4df09cb2-60e7-488c-9a5b-5d6a3b36443a
Type: STORAGE - REQUIREMENT
Rule-Sets:
Rule-Set #1:
VSAN.hostFailuresToTolerate: 1

/192.168.100.1/vsanDC/storage/vmprofiles> show 1
path: /192.168.100.1/vsanDC/storage/vmprofiles/FTT2
class: RbVmomi::PBM::PbmCapabilityProfile
Name: FTT2
Description:
Failures to Tolerate = 2
ProfileId: 90d00d53-a588-4789-84b5-ccf9eacff67d
Type: STORAGE - REQUIREMENT
Rule-Sets:
Rule-Set #1:
VSAN.hostFailuresToTolerate: 2
```

Viewing Virtual SAN Datastore Capacity: “show vsanDatastore” Here is an example of using “ls” to list out datastores within the infrastructure and then using “show” to obtain high level information on the “vsanDatastore”. Notice the capacity and free space of the vsanDatastore.

```
/192.168.100.1/vsanDC/datastores> ls
0 datastore1: 992.14GB 0.7%
1 datastore1 (1): 992.14GB 0.1%
2 vsanDatastore: 2999.77GB 17.7%
3 datastore1 (3): 992.14GB 0.1%
4 datastore1 (2): 992.14GB 0.1%

/192.168.100.1/vsanDC/datastores> show vsanDatastore/
path: /192.168.100.1/vsanDC/datastores/vsanDatastore
type: vsan
url: ds:///vmfs/volumes/vsan:5207cb725036c9fc-3e560cb2fb96f36d/
multipleHostAccess: true
capacity: 2999.77GB
free space: 2469.14GB
```

In the following example, we can see the navigation of the datastore directory. Within it we can find listings of all of the datastores in the vSphere infrastructure. By drilling down into a specific datastore, we can then see all of the files and virtual machines along with the hosts consuming storage as well.

The capability sets directory does not currently list out its contents.

```
/192.168.100.1/vsanDC/datastores> ls
0 datastore1: 992.14GB 0.7%
1 datastore1 (1): 992.14GB 0.1%
2 vsanDatastore: 2999.77GB 17.7%
3 datastore1 (3): 992.14GB 0.1%
4 datastore1 (2): 992.14GB 0.1%

/192.168.100.1/vsanDC/datastores/vsanDatastore> ls
0 files/
1 vms/
2 hosts/
3 capabilitysets/

/192.168.100.1/vsanDC/datastores/vsanDatastore> ls 0
0 vc_backup/
1 551eae53-03fa-9e3c-fa78-a0d3c1039ba8/
2 vSphere Data Protection 5.5/
3 eac4b153-91a6-de41-68e7-a0d3c1039ba8/
4 Tiny Linux template/
5 4fb1b553-a893-bcbb-9c34-a0d3c1039ba8/
6 vcsa5ulc/
7 93952f53-1367-4cc1-c4f0-a0d3c1045888/
8 VMware vCenter Server Appliance/
9 d58e2d53-a9c4-a3f1-36b3-a0d3c1045888/

/192.168.100.1/vsanDC/datastores/vsanDatastore> ls 1
0 VMware vCenter Server Appliance: poweredOn
/10.144.106.87/vsanDC/datastores/vsanDatastore> ls 2
0 10.144.97.178 (host): cpu 2*12*2.09 GHz, memory 137.00 GB
1 10.144.97.179 (host): cpu 2*12*2.09 GHz, memory 137.00 GB
2 10.144.97.180 (host): cpu 2*12*2.09 GHz, memory 137.00 GB
```

Virtual SAN RVC Commands Overview

The RVC Virtual SAN (VSAN) v1.8.0 namespace contains **42** commands to interact with a Virtual SAN infrastructure. More detail on how to use each command, and examples, are included in the next section.

COMMAND	DESCRIPTION
<code>vsan.apply_license_to_cluster</code> ~cluster~	Apply license to the VSAN cluster. The argument to the command is a reference to the cluster and a license key. -k License key
<code>vsan.check_limits</code> ~host~ ~cluster~	Gather and validate counters against limits. Can be ran either entire cluster or single host.
<code>vsan.check_state</code> ~cluster~	Check if VMs and Virtual SAN objects are valid and accessible. -e Un-register and re-register VMs in inventory -r Refresh state and then check state
<code>vsan.clear_disks_cache</code>	This command clears RVC disk cache. It does not impact VSAN or physical disk caching.
<code>vsan.cluster_change_autoclaim</code> ~cluster~	Change the Disk claim model on the cluster from Manual to Automatic or vice-versa -e Enable autoclaim -d Disable autoclaim
<code>vsan.cluster_change_checksum</code> ~cluster~	Enable or disable the checksum on the cluster. This command is for future use. -e Enable checksum -d Disable checksum
<code>vsan.cluster_info</code> ~cluster~	Prints cluster, storage and network information from all hosts in the cluster
<code>vsan.cluster_set_default_policy</code> ~cluster~	Set a default policy for all object types on the cluster. Capabilities include: ("hostFailuresToTolerate") ("forceProvisioning") ("stripeWidth") ("proportionalCapacity") ("cacheReservation") Values for the capabilities are integers specifies as i0, i1, etc.
<code>vsan.cmmds_find</code> ~cluster~ ~host~	Query the CMMDS Database directly to return information about objects, components and entities -t Type -u UUID -o Owner
<code>vsan.disable_vsan_on_cluster</code> ~cluster~	Disables Virtual SAN on the cluster
<code>vsan.disks_info</code> ~host~	Displays information about the disks resident on a host. SSD and HDD. -s Include adapter information
<code>vsan.disk_object_info</code> ~cluster~ ~host~ diskuid	Display information on all VSAN objects that reside on a physical disk

RVC Command Reference Guide for Virtual SAN

<code>vsan.disks_stats ~cluster~ ~host~</code>	Show stats on all disks in VSAN cluster on one host
<code>vsan.enable_vsan_on_cluster ~cluster~</code>	Enables Virtual SAN on the cluster in Automatic mode.
	-d Disable autoclaim - Manual mode claiming of disks
	-e Enable vsan checksum enforcement – this is for future use.
<code>vsan.enter_maintenance_mode ~host~</code>	Put host into maintenance mode
	-v Evacuation mode is one of: <ul style="list-style-type: none"> ▪ ensureObjectAccessibility (default) ▪ evacuateAllData ▪ noAction
	-t Timeout, in seconds
	-n Immediate action – no wait
<code>vsan.fix_renamed_vms ~vm~</code>	Rename VM to the name of its configuration file without the full path and .vmx extension.
	-f Force. Required to perform actual deletion
<code>Vsan.health.*</code>	These commands are only available when the Health Services are installed. If you wish to learn more about the Virtual SAN Health Services, please see the <i>VSAN 6.0 Health Services Guide</i> .
<code>vsan.host_claim_disks_differently ~host~</code>	Tag devices as capacity_flash, HDD or SSD. Needed with All Flash VSAN configurations
	-m Model of disk to claim as capacity tier
	-d Disk name to claim as capacity tier
	-c Claim/tag types
<code>vsan.host_consume_disks ~host~</code>	Consume all eligible disks on host for Virtual SAN
	-f Filter for SSD disk
	-i Filter for HDD disks
<code>vsan.host_evacuate_data ~host~</code>	Evacuate hosts from a Virtual SAN cluster
	-a Remove the need for free space for rebuilding
	-n Do not evacuate data
	-t Time out for evacuation in seconds (default: no timeout)
<code>vsan.host_exit_evacuation ~host~</code>	Bring evacuated hosts back into cluster
<code>vsan.host_info ~host~</code>	Display VSAN information about a host
<code>vsan.host_wipe_non_vsan_disks ~host~</code>	Delete all contents from disks that contain non-Virtual SAN partitions.
	-d Specify a disk for wiping
	-i Run in interactive mode
	-f Force. Required to perform actual deletion
<code>vsan.host_wipe_vsan_disks ~host~</code>	Delete all contents from disks consumed by Virtual SAN.
	-d Specify a disk for wiping
	-i Run in interactive mode
	-a Remove the need for free space for rebuilding
	-n No action
	-f Force. Required to perform actual deletion
<code>vsan.lldpnetmap ~cluster~</code>	Gather LLDP mapping information from a set of hosts
<code>vsan.obj_status_report ~cluster~</code>	Print component status for objects in a cluster or on a host (e.g. Health)
<code>vsan.object_info ~cluster~ ~obj_uuid~</code>	Display information about a VSAN object
	-s Omit extra attribute info
	-i Include detailed usage info

RVC Command Reference Guide for Virtual SAN

<pre>vsan.object_reconfigure ~object_uuid~ -p ~policy~</pre>	Reconfigures the policy on a VSAN object. Policy parameters include: <ul style="list-style-type: none"> ▪ hostFailuresToTolerate ▪ forceProvisioning ▪ stripeWidth ▪ proportionalCapacity ▪ cacheReservation 																				
<pre>vsan.observer ~cluster~</pre>	Start the VSAN Observer monitoring and troubleshooting utility. <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">-f</td> <td>Dump the metric to a file</td> </tr> <tr> <td>-r</td> <td>Run a web server to view the metrics via a web browser</td> </tr> <tr> <td>-p</td> <td>Port to run web server (default:8010)</td> </tr> <tr> <td>-g</td> <td>Generate an html bundle of raw stats</td> </tr> <tr> <td>-m</td> <td>Max runtime, in hours (default: 2)</td> </tr> <tr> <td>-o</td> <td>Force</td> </tr> <tr> <td>-e</td> <td>Run forever</td> </tr> <tr> <td>-n</td> <td>Don't use HTTPS (no login required)</td> </tr> <tr> <td>-a</td> <td>Max disk space (in GB) to use (default: 5)</td> </tr> <tr> <td>-i</td> <td>Collection interval, in seconds (default: 60)</td> </tr> </table>	-f	Dump the metric to a file	-r	Run a web server to view the metrics via a web browser	-p	Port to run web server (default:8010)	-g	Generate an html bundle of raw stats	-m	Max runtime, in hours (default: 2)	-o	Force	-e	Run forever	-n	Don't use HTTPS (no login required)	-a	Max disk space (in GB) to use (default: 5)	-i	Collection interval, in seconds (default: 60)
-f	Dump the metric to a file																				
-r	Run a web server to view the metrics via a web browser																				
-p	Port to run web server (default:8010)																				
-g	Generate an html bundle of raw stats																				
-m	Max runtime, in hours (default: 2)																				
-o	Force																				
-e	Run forever																				
-n	Don't use HTTPS (no login required)																				
-a	Max disk space (in GB) to use (default: 5)																				
-i	Collection interval, in seconds (default: 60)																				
<pre>vsan.observer_process_statsfile ~statsfile~ ~outputpath~</pre>	Create HTML viewable version from VSAN.Observer JSON file <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">-m</td> <td>Max number of traces to process</td> </tr> </table>	-m	Max number of traces to process																		
-m	Max number of traces to process																				
<pre>vsan.proactive_rebalance ~cluster~</pre>	Proactively rebalance the VSAN cluster's objects and components across all nodes and disks <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">-s</td> <td>Start proactive rebalance</td> </tr> <tr> <td>-t</td> <td>How long in seconds, to run proactive rebalance for</td> </tr> <tr> <td>-v</td> <td>Variance threshold at which a disk's contents are considered for balance</td> </tr> <tr> <td>-i</td> <td>Length of time, in seconds, for variance threshold to be continuously exceeded before disk's contents are considered for balance</td> </tr> <tr> <td>-r</td> <td>Amount of data, in MB, to be moved per hour</td> </tr> <tr> <td>-o</td> <td>Stop proactive rebalance</td> </tr> </table>	-s	Start proactive rebalance	-t	How long in seconds, to run proactive rebalance for	-v	Variance threshold at which a disk's contents are considered for balance	-i	Length of time, in seconds, for variance threshold to be continuously exceeded before disk's contents are considered for balance	-r	Amount of data, in MB, to be moved per hour	-o	Stop proactive rebalance								
-s	Start proactive rebalance																				
-t	How long in seconds, to run proactive rebalance for																				
-v	Variance threshold at which a disk's contents are considered for balance																				
-i	Length of time, in seconds, for variance threshold to be continuously exceeded before disk's contents are considered for balance																				
-r	Amount of data, in MB, to be moved per hour																				
-o	Stop proactive rebalance																				
<pre>vsan.proactive_rebalance_info ~cluster~</pre>	Monitor the proactive rebalance activity																				
<pre>vsan.purge_inaccessible_vswp_objects ~cluster~</pre>	Cleanup stranded VM swap objects – only used as part of the on-disk format upgrade from v1 to v2 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">-f</td> <td>Force</td> </tr> </table>	-f	Force																		
-f	Force																				

vsan.reapply_vsan_vmknics_config ~host~	Unbinds and rebinds VSAN to its VMKNICs	
	-v	Specify a specific NIC
	-d	Dry-run - Test this to see what changes would be made.
vsan.recover_spbm ~cluster~ ~host~	Recover the storage policy based management configuration on a cluster or host	
	-f	Force
	-d	Dry-run - Test this to see what changes would be made.
vsan.resync_dashboard ~cluster~ ~host~	Resynchronize dashboard for all objects in cluster or a host	
	-r	Refresh rate, in seconds
vsan.scrubber_info	Check for latent sector errors. Command implemented for future use.	
vsan.support_information	Gather support information. Use when directed by GSS.	
vsan.v2_ondisk_upgrade	Upgrade the on-disk format and objects from v1 to v2	
	-i	Ignore objects, simply upgrade on disk format
	-d	Downgrade format from v2 back to v1
	-a	Allow reduced redundancy
	-f	Force
vsan.vm_object_info ~vm~	Shows all object information about a VM	
	-c	Cluster
	-p	Host perspective
	-i	Included detailed usage
vsan.vm_perf_stats ~vm~	Query info on selected VM and displays a table of average IOPS, throughput and latency for the VM over 2 samples 20 seconds apart.	
	-i	Configure interval
	-s	Show VM's objects
vsan.vmdk_stats ~vm~	Display cache and capacity stats for VMs	
vsan.what_if_host_failures ~cluster~	Simulates how host failures impact VSAN resource usage compared to current usage. Can only simulate 1 host failure for now.	
	-n	Number of failures to simulate (default: 1)
	-s	Show current resource usage per host

Detailed Virtual SAN RVC Commands

This section provides detailed information about each of the RVC commands for VSAN, including command line options, examples and information regarding the output produced from each of the commands.

vsan.apply_license_to_cluster

Applies a VSAN license to a VSAN cluster. The command runs against a cluster object, and takes a license key as an argument.

Usage:

```
vsan.apply_license_to_cluster {cluster} {-k, --license-key} {-h, --help}
```

Examples:

- Display help:

```
> vsan.apply_license_to_cluster 0 -h
usage: apply_license_to_cluster [opts] cluster
Apply license to VSAN
  cluster: Path to a ClusterComputeResource
  --license-key, -k <s>: License key to be applied to the cluster
  --null-reconfigure, -r: (default: true)
  --help, -h: Show this message
```

- Apply a license key to a cluster:

```
> vsan.apply_license_to_cluster 0 -k aaaaa-bbbbb-cccc-d-dddd-eeee
VSAN60: Applying VSAN License on the cluster...
VSAN60: Null-Reconfigure to force auto-claim...
ReconfigureComputeResource VSAN60: success
cs-ie-h01.ie.local: success
cs-ie-h02.ie.local: success
cs-ie-h03.ie.local: success
cs-ie-h04.ie.local: success
/localhost/IE-VSAN-DC/computers>
```

vsan.check_limits

This command displays resource information and is useful for ensuring that Virtual SAN is operating within its resource limits. The command runs against a cluster object.

Usage:

```
vsan.check_limits {host|cluster} {-h, --help}
```

Examples:

- Display help:

```
> vsan.check_limits -h
usage: check_limits hosts_and_clusters...
Gathers (and checks) counters against limits
  hosts_and_clusters: Path to a HostSystem or ClusterComputeResource
--help, -h: Show this message
>
```

- Display the current limits of a cluster:

This output is taken from a cluster upgraded to version 6.0 and the on-disk format has been upgraded to v2, so the number of components supported per host is shown as 9000.

```
> vsan.check_limits 0
2014-11-27 14:52:25 +0000: Querying limit stats from all hosts ...
2014-11-27 14:52:27 +0000: Fetching VSAN disk info from cs-ie-h03 (may take a moment) ..
2014-11-27 14:52:27 +0000: Fetching VSAN disk info from cs-ie-h02 (may take a moment) ..
2014-11-27 14:52:27 +0000: Fetching VSAN disk info from cs-ie-h01 (may take a moment) ..
2014-11-27 14:52:27 +0000: Fetching VSAN disk info from cs-ie-h04 (may take a moment) ..
2014-11-27 14:52:29 +0000: Done fetching VSAN disk infos
+-----+-----+-----+
| Host | RDT | Disks |
+-----+-----+-----+
| cs-ie-h01.ie.local | Assocs: 355/45000 | Components: 111/9000 | |
| | | Sockets: 27/10000 | naa.600508b1001ccd5d506e7ed19c40a64c: 59% |
| | | Clients: 0 | naa.600508b1001c16be6e256767284eaf88: 67% |
| | | Owners: 69 | naa.600508b1001c2ee9a6446e708105054b: 66% |
| | | | naa.600508b1001c3ea7838c0436dbe6d7a2: 67% |
| | | | naa.600508b1001c61cedd42b0c3fbf55132: 0% |
| | | | naa.600508b1001c388c92e817e43fcd5237: 65% |
| | | | naa.600508b1001c64816271482a56a48c3c: 65% |
| | | | naa.600508b1001c79748e8465571b6f4a46: 62% |
| cs-ie-h02.ie.local | Assocs: 103/45000 | Components: 75/9000 |
| | | Sockets: 27/10000 | naa.600508b1001c0cc0ba2a3866cf8e28be: 64% |
| | | Clients: 0 | naa.600508b1001c19335174d82278dee603: 68% |
| | | Owners: 9 | naa.600508b1001c07d525259e83da9541bf: 45% |
| | | | naa.600508b1001c64b76c8ceb56e816a89d: 0% |
| | | | naa.600508b1001ca36381622ca880f3aacd: 53% |
| | | | naa.600508b1001cb2234d6ff4f7b1144f59: 71% |
| cs-ie-h03.ie.local | Assocs: 121/45000 | Components: 81/9000 |
| | | Sockets: 27/10000 | naa.600508b1001c9c8b5f6f0d7a2be44433: 0% |
| | | Clients: 0 | naa.600508b1001cd259ab7ef213c87eaad7: 53% |
| | | Owners: 13 | naa.600508b1001c1a7f310269ccd51a4e83: 59% |
| | | | naa.600508b1001c9b93053e6dc3ea9bf3ef: 76% |
| | | | naa.600508b1001c2b7a3d39534ac6beb92d: 66% |
```

```

|                                     | naa.600508b1001ceefc4213ceb9b51c4be4: 69% | |
|                                     | naa.600508b1001cb11f3292fe743a0fd2e7: 60% |
| cs-ie-h04.ie.local | Assocs: 133/45000 | Components: 86/9000 |
|                                     | Sockets: 27/10000 | naa.600508b1001c29d8145d6cc1925e9fb9: 0% |
|                                     | Clients: 0         | naa.600508b1001c258181f0a088f6e40dab: 74% |
|                                     | Owners: 15        | naa.600508b1001cadff5d80ba7665b8f09a: 43% |
|                                     |                   | naa.600508b1001c846c000c3d9114ed71b3: 62% |
|                                     |                   | naa.600508b1001c51f3a696fe0bbcb5096: 65% |
|                                     |                   | naa.600508b1001c4b820b4d80f9f8acfa95: 73% |
|                                     |                   | naa.600508b1001c6a664d5d576299cec941: 62% |
+-----+-----+-----+
/ie-vcsa-03.ie.local/vsan-dc/computers>

```

RDT relates to networking limits and Disks relates to storage limits. RDT is Reliable Datagram Transport and is the Virtual SAN network transport. RDT has a number of limits listed. These are Associations (Assocs) and Sockets. Additional information regarding Clients and Owners is also displayed. For an explanation on RDT Assocs/Sockets/Client/Owners, please refer to the Virtual SAN 6.0 Troubleshooting Reference Manual. A link can be found at the end of this document.

vsan.check_state

There are 3 checks that this command implements:

- Check for inaccessible Virtual SAN objects
- Check for invalid/inaccessible VMs
- Check for VMs for which VC/hostd/vmx are out of sync

Inaccessible Virtual SAN objects are an indication that there is probably a failure somewhere in the cluster, but that Virtual SAN is still able to track the virtual machine. An invalid or inaccessible object is when the VM has objects that have lost the majority of its components or votes, again due to hardware failures. Note that for a VM's object to be accessible, it must have a full, intact mirror and greater than 50% of its components/votes available.

The next check is for invalid or inaccessible VMs. These are VMs that, most likely due to the fact that the failure(s) that have occurred in the cluster, have been impacted so much that it is no longer accessible by the vCenter server or the ESXi hosts. This is likely be due to the fact that the VM Home Namespace, where the .vmx file resides, is no longer online. Common causes are clusters that have had multiple failures, but the virtual machines have been configured to tolerate only one failure, or network outages.

Finally, the command checks to ensure that the vCenter Server and the ESXi hosts are in agreement with regards to the state of the cluster.

Usage:

```
vsan.check_state {host|cluster} {-r, refresh-state} {-e, --reregister-vms}
                 {-f, --force}{-h, --help}
```

Examples:

- Display help:

The command takes a cluster as an argument. There are additional arguments that can be used to resolve state issues if objects or VMs are found to be out of sync.

```
> vsan.check_state -h
usage: check_state [opts] cluster_or_host
Checks state of VMs and VSAN objects
  cluster_or_host: Path to a ClusterComputeResource or HostSystem
  --refresh-state, -r: Not just check state, but also refresh
  --reregister-vms, -e: Not just check for vms with VC/hostd/vmx out of sync
but also fix them by un-registering and re-registering them
  --force, -f: Force to re-register vms, without confirmation
  --help, -h: Show this message
```

- Check the state of a cluster when everything is OK:

```
> vsan.check_state 0
2014-10-19 16:03:39 +0000: Step 1: Check for inaccessible VSAN objects
Detected 0 objects to be inaccessible

2014-10-19 16:03:39 +0000: Step 2: Check for invalid/inaccessible VMs

2014-10-19 16:03:39 +0000: Step 3: Check for VMs for which VC/hostd/vmx are
out of sync
Did not find VMs for which VC/hostd/vmx are out of sync
```

- Check the state of a cluster when there are inaccessible objects:

```
> vsan.check_state vsan
2014-11-27 14:51:24 +0000: Step 1: Check for inaccessible VSAN objects
Detected 19 objects to be inaccessible
Detected 34723e54-7840-c72e-42a5-0010185def78 on cs-ie-h02.ie.local to be inaccessible
Detected 4a743e54-f452-4435-1d15-001f29595f9f on cs-ie-h02.ie.local to be inaccessible
Detected 3a743e54-a8c2-d13d-6d0c-001f29595f9f on cs-ie-h02.ie.local to be inaccessible
Detected 6e713e54-4819-af51-edb5-0010185def78 on cs-ie-h02.ie.local to be inaccessible
Detected 2d6d3e54-848f-3256-b7d0-001b21168828 on cs-ie-h02.ie.local to be inaccessible
Detected f0703e54-4404-c85b-0742-001f29595f9f on cs-ie-h02.ie.local to be inaccessible
Detected 76723e54-74a3-0075-c1a9-001b21168828 on cs-ie-h02.ie.local to be inaccessible
Detected e4c33b54-1824-537c-472e-0010185def78 on cs-ie-h02.ie.local to be inaccessible
Detected ef713e54-186d-d77c-bf27-001b21168828 on cs-ie-h02.ie.local to be inaccessible
Detected 77703e54-0420-3a81-dc1a-001f29595f9f on cs-ie-h02.ie.local to be inaccessible
Detected 30af3e54-24fe-4699-f300-001b21168828 on cs-ie-h02.ie.local to be inaccessible
Detected 58723e54-047e-86a0-4803-001b21168828 on cs-ie-h02.ie.local to be inaccessible
Detected 85713e54-dcbe-fea6-8205-001b21168828 on cs-ie-h02.ie.local to be inaccessible
Detected c2733e54-ac02-78ca-f0ce-001f29595f9f on cs-ie-h02.ie.local to be inaccessible
Detected 94713e54-08e1-18d3-ffd7-001b21168828 on cs-ie-h02.ie.local to be inaccessible
Detected f0723e54-18d2-79f5-be44-001b21168828 on cs-ie-h02.ie.local to be inaccessible
Detected 3b713e54-9851-31f6-2679-001f29595f9f on cs-ie-h02.ie.local to be inaccessible
Detected fd743e54-1863-c6fb-1845-001f29595f9f on cs-ie-h02.ie.local to be inaccessible
Detected 94733e54-e81c-c3fe-8bfc-001b21168828 on cs-ie-h02.ie.local to be inaccessible

2014-11-27 14:51:25 +0000: Step 2: Check for invalid/inaccessible VMs

2014-11-27 14:51:25 +0000: Step 3: Check for VMs for which VC/hostd/vmx are out of sync
Did not find VMs for which VC/hostd/vmx are out of sync

/ie-vcsa-03.ie.local/vsan-dc/computers>
```

If objects are inaccessible, the *Virtual SAN 6.0 Troubleshooting Reference Guide* and the *Virtual SAN Health Services* can be utilized to locate the root cause.

vsan.clear_disks_cache

Clears the disks cache within RVC. *It does not have any effect on the Virtual SAN datastore or its physical or virtual disks.* RVC keeps a cache of all disks a host has, so it only needs to retrieve this information once. RVC automatically clears this cache if disks are added or removed using RVC. However, if disks are added or removed outside of a given RVC session, like in the vSphere Web Client, then RVC may display UUIDs instead of full disk information in commands like `vsan.disks_stats`. In those cases, one can manually clear the RVC cache using this command. It does not take any arguments.

Usage:

```
vsan.clear_disks_cache {cluster} {-h, --help}
```

Examples:

- Display help:

```
> vsan.clear_disks_cache -h
usage: clear_disks_cache
Clear cached disks information
--help, -h: Show this message
```

- Clear the disks cache:

```
> vsan.clear_disks_cache
>
```


vsan.cluster_change_autoclaim

Changes the disk auto claim mechanism. When enabled, Virtual SAN automatically claims any local, empty disks. If it is disabled, it does not claim disks automatically. It takes a cluster object as an argument.

Usage:

```
vsan.cluster_change_autoclaim {cluster} {-e, --enable} {-d, --disable}
                                {-h, --help}
```

Examples:

- Display help:

```
> vsan.cluster_change_autoclaim -h
usage: cluster_change_autoclaim [opts] cluster
Enable/Disable autoclaim on a VSAN cluster
cluster: Path to a ClusterComputeResource
--enable, -e: Enable auto-claim
--disable, -d: Disable auto-claim
--help, -h: Show this message
```

- Enable autoclaim on the cluster:

```
> vsan.cluster_change_autoclaim -e 0
ReconfigureComputeResource VSAN60: success
cs-ie-h01.ie.local: success
cs-ie-h02.ie.local: success
cs-ie-h03.ie.local: success
cs-ie-h04.ie.local: success
cs-ie-h01.ie.local: success
cs-ie-h02.ie.local: success
cs-ie-h03.ie.local: success
cs-ie-h04.ie.local: success
```

- Disable autoclaim of the cluster:

```
> vsan.cluster_change_autoclaim -d 0
ReconfigureComputeResource VSAN60: success
cs-ie-h01.ie.local: success
cs-ie-h02.ie.local: success
cs-ie-h03.ie.local: success
cs-ie-h04.ie.local: success
cs-ie-h01.ie.local: success
cs-ie-h02.ie.local: success
cs-ie-h03.ie.local: success
cs-ie-h04.ie.local: success
```

vsan.cluster_change_checksum

This command enables or disables checksum enforcement on the Virtual SAN cluster. It is reserved for *future use*, when 520 byte sector disk drives are supported with Virtual SAN.

Usage:

```
vsan.cluster_change_checksum {cluster} {-e, --enable} {-d, --disable}
                               {-h, --help}
```

Examples:

- Display help:

```
> vsan.cluster_change_checksum -h
usage: cluster_change_checksum [opts] cluster
Enable/Disable VSAN checksum enforcement on a cluster
cluster: Path to a ClusterComputeResource
  --enable, -e:  Enable checksum enforcement
  --disable, -d: Disable checksum enforcement
  --help, -h:   Show this message
```

- Enable checksum of the cluster:

If a cluster has hosts which do not support checksumming, the following error is displayed:

```
> vsan.cluster_change_checksum 0 -e
RuntimeError: unknown VMODL type NotSupportedHostForChecksum
```

vsan.cluster_info

Produces detailed information for each node in the cluster, so for very large clusters, the amount of information produced by the commands starts to get quite large.

Usage:

```
vsan.cluster_info {cluster} {-h, --help}
```

Examples:

- Display help:

```
> vsan.cluster_info -h
usage: cluster_info hosts_and_clusters...
Print VSAN config info about a cluster or hosts
  hosts_and_clusters: Path to a HostSystem or ClusterComputeResource
--help, -h: Show this message
```

- Display information about the cluster:

The command takes a cluster as an argument. In this output, there is a 4-node cluster, but the output is truncated to show the first two hosts only. This output shows if Virtual SAN is enabled, whether the role is master, backup or agent, the UUIDs of the other nodes in the cluster, disk mappings and network information

```
> vsan.cluster_info 0
2014-11-27 14:44:02 +0000: Fetching host info from cs-ie-h04.ie.local (may take a
moment) ...
2014-11-27 14:44:02 +0000: Fetching host info from cs-ie-h03.ie.local (may take a
moment) ...
2014-11-27 14:44:02 +0000: Fetching host info from cs-ie-h02.ie.local (may take a
moment) ...
2014-11-27 14:44:02 +0000: Fetching host info from cs-ie-h01.ie.local (may take a
moment) ...
Host: cs-ie-h02.ie.local
  Product: VMware ESXi 6.0.0 build-2305723
  VSAN enabled: yes
  Cluster info:
    Cluster role: agent
    Cluster UUID: 529ccbe4-81d2-89bc-7a70-a9c69bd23a19
    Node UUID: 54196e13-7f5f-cba8-5bac-001517a69c72
    Member UUIDs: ["54188e3a-84fd-9a38-23ba-001b21168828", "545ca9af-ff4b-fc84-dcee-
001f29595f9f", "5460b129-4084-7550-46e1-0010185def78", "54196e13-7f5f-cba8-5bac-
001517a69c72"] (4)
    Node evacuated: no
  Storage info:
    Auto claim: no
    Checksum enforced: no
    Disk Mappings:
```

```

SSD: HP Serial Attached SCSI Disk (naa.600508b1001c577e11dd042e142a583f) - 186 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001c19335174d82278dee603) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001ca36381622ca880f3aacd) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001cb2234d6ff4f7b1144f59) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001c0cc0ba2a3866cf8e28be) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001c07d525259e83da9541bf) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001c10548f5105fc60246b4a) - 136 GB, v1
FaultDomainInfo:
  Not configured
NetworkInfo:
  Adapter: vmk2 (172.32.0.2)

Host: cs-ie-h03.ie.local
Product: VMware ESXi 6.0.0 build-2305723
VSAN enabled: yes
Cluster info:
  Cluster role: agent
  Cluster UUID: 529ccbe4-81d2-89bc-7a70-a9c69bd23a19
  Node UUID: 5460b129-4084-7550-46e1-0010185def78
  Member UUIDs: ["54188e3a-84fd-9a38-23ba-001b21168828", "545ca9af-ff4b-fc84-dcee-
001f29595f9f", "5460b129-4084-7550-46e1-0010185def78", "54196e13-7f5f-cba8-5bac-
001517a69c72"] (4)
  Node evacuated: no
Storage info:
  Auto claim: no
  Checksum enforced: no
  Disk Mappings:
SSD: HP Serial Attached SCSI Disk (naa.600508b1001c9c8b5f6f0d7a2be44433) - 186 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001ceefc4213ceb9b51c4be4) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001cd259ab7ef213c87eaaad7) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001c2b7a3d39534ac6beb92d) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001cb11f3292fe743a0fd2e7) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001c1a7f310269ccd51a4e83) - 136 GB, v1
MD: HP Serial Attached SCSI Disk (naa.600508b1001c9b93053e6dc3ea9bf3ef) - 136 GB, v1
FaultDomainInfo:
  Not configured
NetworkInfo:
  Adapter: vmk2 (172.32.0.3)
  <<truncated>>

```

This is a useful command to get a “big picture” of the cluster. Useful information such as the number of nodes in the cluster (4 as per Member UUIDs) is displayed. This command in 6.0 has some additional information not in the 5.5 versions; namely information on whether the node is evacuated and whether fault domains has been configured.

Note: Although the output also reports emulated checksums in version 6.0, emulated checksums are not yet supported.

vsan.cluster_set_default_policy

Allows an administrator to create a new default policy and apply it cluster wide. It takes a cluster and a set of policy settings as arguments.

The policy settings can be one or more of the following:

- hostFailuresToTolerate
- forceProvisioning
- stripeWidth
- proportionalCapacity
- cacheReservation

These policy settings take an integer argument that is specified as i0, i1, i2 and so on. The syntax is rather complex. Please refer to the examples below for guidance.

Usage:

```
vsan.cluster_set_default_policy {cluster} {policy} {-h, --help}
```

Examples:

- Display help:

```
> vsan.cluster_set_default_policy -h
usage: cluster_set_default_policy cluster policy
Set default policy on a cluster
  cluster: Path to a ClusterComputeResource
  policy:
  --help, -h: Show this message
```

- Set a default policy with FTT=1 and SW=2:

Here is an example of setting a default policy to tolerate 1 failure in the cluster and to deploy objects with a stripe width of 2:

```
> vsan.cluster_set_default_policy 0 "((\" hostFailuresToTolerate \" i1) (\
stripeWidth \" i2))"
>
```

- Set a default policy with FTT=1 and SW=1:

Here is an example of setting a default policy to tolerate 1 failure and deploy objects with a stripe width of 1:

```
> vsan.cluster_set_default_policy 0 "(\\" hostFailuresToTolerate \" i1) (\\"  
stripeWidth \" i1)\"  
>
```

The command does not return any output when it successfully completes.

vsan.cmmds_find

Display additional information about an object or component on Virtual SAN, when only the UUID is known. It provides low-level access to the "cmmds-tool find" from RVC.

Usage:

```
vsan.cmmds_find {cluster|host} {, -t, --type} {-u, uuid} {-o, --owner}
{-h, --help}
```

Types:

DISK – represents a magnetic disk or flash device
 DOM_OBJECT – represents a composite objects
 POLICY type – represents a policy
 LSOM_OBJECT – represents a component

Examples:

- Display help:

```
> vsan.cmmds_find -h
usage: cmmds_find [opts] cluster_or_host
CMMDS Find
  cluster_or_host: Path to a ClusterComputeResource or HostSystem
  --type, -t <s>:  CMMDS type, e.g. DOM_OBJECT, LSOM_OBJECT, POLICY, DISK
etc.
  --uuid, -u <s>:  UUID of the entry.
  --owner, -o <s>:  UUID of the owning node.
  --help, -h:     Show this message
/localhost/IE-VSAN-DC/computers>
```

- Display information about an LSOM Object (component):

```
> vsan.cmmds_find 0 -t LSOM_OBJECT -u 3d69db54-0ad6-64f2-b95a-001517a69c72
+-----+-----+-----+-----+
| # | Type          | UUID                                     | Owner                               |
+-----+-----+-----+-----+
| 1 | LSOM_OBJECT   | 3d69db54-0ad6-64f2-b95a-001517a69c72 | cs-ie-h01.ie.local                 |
|   |               |                                         |                                     |
|   |               |                                         |                                     |
|   |               |                                         |                                     |
+-----+-----+-----+-----+
```

Health	Content
Healthy	{ "diskUuid"=>"528f27f4-7847-5f25-6d60-d01441f9a23d", "compositeUuid"=>"c6eb8a54-7ac4-c85f-a3de-001b21168828", "capacityUsed"=>21428699136, "physCapacityUsed"=>21428699136}

- Display information about a physical disk

```
> vsan.cmmnds_find 0 -t DISK -u 528f27f4-7847-5f25-6d60-d01441f9a23d
```

#	Type	UUID	Owner
1	DISK	528f27f4-7847-5f25-6d60-d01441f9a23d	cs-ie-h01.ie.local

Health	Content
Healthy	{ "capacity"=>146502844416, "iops"=>100, "iopsWritePenalty"=>10000000, "throughput"=>200000000, "throughputWritePenalty"=>0, "latency"=>3400000, "latencyDeviation"=>0, "reliabilityBase"=>10, "reliabilityExponent"=>15, "mtbf"=>1600000, "l2CacheCapacity"=>0, "l1CacheCapacity"=>16777216, "isSsd"=>0, "ssdUuid"=>"52070d2e-48bf-d11b-2516-9199a24969b6", "volumeName"=>"NA", "formatVersion"=>2, "devName"=>"naa.600508b1001c388c92e817e43fcd5237:2", "ssdCapacity"=>0, "rdtMuxGroup"=>175231866576640, "isAllFlash"=>0}

vsan.disable_vsan_on_cluster

Disable Virtual SAN. It takes the cluster as an argument. Note that this command does not prompt for confirmation, but simply goes ahead and disables Virtual SAN. Use with caution.

Usage:

```
vsan.disable_vsan_on_cluster {cluster} {-h, --help}
```

Examples:

- Display help:

```
> vsan.disable_vsan_on_cluster -h
usage: disable_vsan_on_cluster cluster
Disable VSAN on a cluster
  cluster: Path to a ClusterComputeResource
  --help, -h: Show this message
```

- Disable Virtual SAN on a cluster:

```
> vsan.disable_vsan_on_cluster 0
ReconfigureComputeResource VSAN60: success
cs-ie-h01.ie.local: success
cs-ie-h02.ie.local: success
cs-ie-h03.ie.local: success
cs-ie-h04.ie.local: success
```

Note: There is a corresponding RVC command called *vsan.enable_vsan_on_cluster* that enables Virtual SAN on a cluster. This will be covered shortly.

vsan.disks_info

Provide a host's view of its disks, and thus takes a host as an argument.

Usage:

```
vsan.disks_info {host} {-s, --show-adapters} {-h, --help}
```

Examples:

- Display help:

```
> vsan.disks_info -h
usage: disks_info [opts] host...
Print physical disk info about a host
host: Path to a HostSystem
--show-adapters, -s: Show adapter information
--help, -h: Show this message
```

- Disable information about the disks on this host. This command displays information about every disk on the host, both magnetic disks (MD) and solid state disks (SSD). *The output shown here has been modified to make it more readable.*

```
> vsan.disks_info 0
2015-02-27 11:32:10 +0000: Gathering disk information for host cs-ie-h01
2015-02-27 11:32:12 +0000: Done gathering disk information
Disks on host cs-ie-h01.ie.local:
+-----+
| DisplayName |
+-----+
| HP Serial Attached SCSI Disk (naa.600508b1001c16be6e256767284eaf88) |
| HP LOGICAL VOLUME |
| |
| |
+-----+

+-----+
| isSSD |
+-----+
| MD |
| |
| |
+-----+

+-----+
| Size |
+-----+
```

```
| 136 GB |
|         |
|         |
+-----+
```

```
+-----+
| State |
+-----+
| inUse |
| VSAN Format Version: v2 |
|         |
| Checksum Enabled: false |
+-----+
```

This level of information is repeated for all disks. If a disk is not used by Virtual SAN, an explanation is given as to why. Existing partition information is one such reason.

```
+-----+
| DisplayName |
+-----+
| Local USB Direct-Access (mpx.vmhba32:C0:T0:L0) |
| Kingston DataTraveler II+ |
|         |
+-----+
```

```
+-----+
| isSSD |
+-----+
| MD |
|         |
|         |
+-----+
```

```
+-----+
| Size |
+-----+
| 1 GB |
|         |
|         |
+-----+
```

```
+-----+
| State |
+-----+
| ineligible (Existing partitions found on disk 'mpx.vmhba32:C0:T0:L0'. |
|         |
| Partition table: |
| 5: 0.24 GB, type = vfat |
| 6: 0.24 GB, type = vfat |
| 7: 0.11 GB, type = coredump |
| 8: 0.28 GB, type = vfat |
|         |
| Adapters: |
| vmhba32 (usb-storage) |
|   USB |
|         |
| Checksum Enabled: false |
+-----+
```

vsan.disk_object_info

Display all of the components that reside on a physical disk.

This command takes two arguments. The first argument corresponds to either a host or cluster. The second argument is the *disk_uuid*. This is the same as the NAA id. This can be found from the *displayName* section of the previous command, *vsan.disks_info*.

Usage:

```
vsan.disk_object_info {cluster|host} {disk_uuid} {-h, --help}
```

Examples:

- Display help:

```
> vsan.disk_object_info -h
usage: disk_object_info cluster_or_host disk_uuid...
Fetch information about all VSAN objects on a given physical disk
  cluster_or_host: Cluster or host on which to fetch the object info
  disk_uuid:
  --help, -h:    Show this message
```

- Display the contents of a disk. Once again, the output can be quite long, so it has been truncated for display in this document.

```
> vsan.disk_object_info 0 naa.600508b1001c3ea7838c0436dbe6d7a2
2015-02-27 11:58:19 +0000: Fetching VSAN disk info from cs-ie-h01.ie.local
(may take a moment) ...
2015-02-27 11:58:20 +0000: Done fetching VSAN disk infos
Physical disk naa.600508b1001c3ea7838c0436dbe6d7a2 (52f630d6-eee1-130b-c34c-
47724c54bd25):
  DOM Object: c6eb8a54-7ac4-c85f-a3de-001b21168828 (v2, owner: cs-ie-
h01.ie.local, policy: forceProvisioning = 0, hostFailuresToTolerate = 1,
spbmProfileId = aa6d5a82-1c88-45da-85d3-3d74b91a5bad, proportionalCapacity = 0,
spbmProfileGenerationNumber = 0, cacheReservation = 0, stripeWidth = 1)
  Context: Part of VM ch-vsan-desktop: Disk: [vsanDatastore] b4eb8a54-d47c-
cd2d-4dae-001b21168828/ch-vsan-desktop.vmdk
    RAID_1
      RAID_0
        Component: 3d69db54-0ad6-64f2-b95a-001517a69c72 (state: ACTIVE (5),
host: cs-ie-h01.ie.local, md: naa.600508b1001c388c92e817e43fcd5237, ssd:
naa.600508b1001c61cedd42b0c3fbf55132,
                                                                    usage: 20.0 GB)
        Component: e487db54-1058-a981-84e8-001b21168828 (state: ACTIVE (5),
host: cs-ie-h01.ie.local, md: naa.600508b1001c2ee9a6446e708105054b, ssd:
naa.600508b1001c61cedd42b0c3fbf55132,
                                                                    usage: 20.0 GB)
```

```

Component: 3d69db54-6613-6af2-607a-001517a69c72 (state: ACTIVE (5),
host: cs-ie-h01.ie.local, md: naa.600508b1001c79748e8465571b6f4a46, ssd:
naa.600508b1001c61cedd42b0c3fbf55132,
usage: 19.9 GB)
Component: 3d69db54-3af4-6bf2-f56c-001517a69c72 (state: ACTIVE (5),
host: cs-ie-h01.ie.local, md: **naa.600508b1001c3ea7838c0436dbe6d7a2**, ssd:
naa.600508b1001c61cedd42b0c3fbf55132,
usage: 20.0 GB)
RAID_0
Component: 3d69db54-56e1-6df2-2877-001517a69c72 (state: ACTIVE (5),
host: 54188e3a-84fd-9a38-23ba-001b21168828, md: 523dd6fb-513a-ea7f-2b4d-
adc7e134ef66, ssd: 521b0bec-c6ce-b7c0-0742-aa428b81c192,
usage: 20.0 GB)
Component: 2289db54-9459-d10c-1e81-001b21168828 (state: ACTIVE (5),
host: 54196e13-7f5f-cba8-5bac-001517a69c72, md: 52f1eb0d-81e3-60ee-d918-
cde90690cb26, ssd: 528b1084-4fa6-7cc1-1d5a-093707258235,
usage: 19.9 GB)
Component: bd89db54-e06d-9cf3-1d81-001b21168828 (state: ACTIVE (5),
host: 54196e13-7f5f-cba8-5bac-001517a69c72, md: 52dc222f-908a-961b-e63b-
810545a6d6cb, ssd: 528b1084-4fa6-7cc1-1d5a-093707258235,
usage: 19.9 GB)
Component: 3d69db54-ea8b-74f2-82ef-001517a69c72 (state: ACTIVE (5),
host: 5460b129-4084-7550-46e1-0010185def78, md: 527226c0-0389-07db-3ad3-
135abe8e58ca, ssd: 52e3ed4c-1f98-fa06-e233-d64fb37b4476,
usage: 20.0 GB)

```

Each object displayed begins with DOM Object. Policy information is displayed for the object, including forceProvisioning, hostFailuresToTolerate, proportionalCapacity, StripeWidth and cacheReservation. The next piece of information is the list of components that go to make up the object. Even when other components that make up the object reside on different hosts and disk, all component are displayed. The components that are make with ** refer to components that are on this particular disk that was queried. Note that the component shown above is part of a replica (RAID_1) which is in turn striped (RAID_0) across a number of magnetic disks.

vsan.disks_stats

Display information about the disks in a host or cluster, including whether or not it is a magnetic disk or solid state drive, how many components reside on the disk, disk capacity, how much is used, if any of it is reserved via the *ObjectSpaceReservation* policy setting, if it's health is OK and what is the version of the on-disk format.

Usage:

```
vsan.disk_stats {cluster|host} {-h, --help}
```

Examples:

- Display help:

```
> vsan.disks_stats -h
usage: disks_stats hosts_and_clusters...
Show stats on all disks in VSAN
hosts_and_clusters: Path to a HostSystem or ClusterComputeResource
--help, -h: Show this message
```

- Display information about the disks from a host perspective. Note that when this command is run against a host, all other disks on the remaining hosts in the cluster appear as N/A in the DisplayName and Host columns:

```
> vsan.disks_stats 0
```

DisplayName	Host	isSSD	Num Comp	Capacity Total	Used	Reserved	Status Health
N/A	N/A	SSD	0	186.27 GB	0 %	0 %	OK (v2)
N/A	N/A	SSD	0	186.27 GB	0 %	0 %	OK (v2)
N/A	N/A	SSD	0	186.27 GB	0 %	0 %	OK (v2)
N/A	N/A	MD	13	136.44 GB	66 %	66 %	OK (v2)
N/A	N/A	MD	12	136.44 GB	64 %	64 %	OK (v2)
N/A	N/A	MD	15	136.44 GB	69 %	54 %	OK (v2)
N/A	N/A	MD	12	136.44 GB	69 %	68 %	OK (v2)
N/A	N/A	MD	27	136.44 GB	44 %	42 %	OK (v2)
N/A	N/A	MD	16	136.44 GB	53 %	52 %	OK (v2)
N/A	N/A	MD	15	136.44 GB	63 %	54 %	OK (v2)
N/A	N/A	MD	11	136.44 GB	60 %	60 %	OK (v2)
N/A	N/A	MD	13	136.44 GB	72 %	13 %	OK (v2)
N/A	N/A	MD	21	136.44 GB	46 %	36 %	OK (v2)
N/A	N/A	MD	11	136.44 GB	74 %	74 %	OK (v2)
N/A	N/A	MD	10	136.44 GB	63 %	55 %	OK (v2)
N/A	N/A	MD	10	136.44 GB	76 %	76 %	OK (v2)
N/A	N/A	MD	16	136.44 GB	59 %	44 %	OK (v2)
N/A	N/A	MD	14	136.44 GB	73 %	66 %	OK (v2)
N/A	N/A	MD	17	136.44 GB	54 %	52 %	OK (v2)
N/A	N/A	MD	9	136.44 GB	66 %	66 %	OK (v2)
naa.600508b1001c61cedd42b0c3fbf55132	cs-ie-h01.ie.local	SSD	0	186.27 GB	0 %	0 %	OK (v2)
naa.600508b1001c16be6e256767284eaf88	cs-ie-h01.ie.local	MD	12	136.44 GB	67 %	67 %	OK (v2)
naa.600508b1001c3ea7838c0436dbe6d7a2	cs-ie-h01.ie.local	MD	18	136.44 GB	67 %	67 %	OK (v2)
naa.600508b1001c2ee9a6446e708105054b	cs-ie-h01.ie.local	MD	12	136.44 GB	67 %	66 %	OK (v2)
naa.600508b1001c388c92e817e43fcd5237	cs-ie-h01.ie.local	MD	32	136.44 GB	66 %	65 %	OK (v2)
naa.600508b1001c64816271482a56a48c3c	cs-ie-h01.ie.local	MD	13	136.44 GB	66 %	66 %	OK (v2)
naa.600508b1001c79748e8465571b6f4a46	cs-ie-h01.ie.local	MD	12	136.44 GB	63 %	63 %	OK (v2)
naa.600508b1001ccd5d506e7ed19c40a64c	cs-ie-h01.ie.local	MD	12	136.44 GB	60 %	59 %	OK (v2)

- Display information about the disks from a cluster perspective. Here is the same output run at the cluster level, which displays all disk and host information missing from the previous output:

```
> vsan.disks_stats 0
2015-02-27 12:12:02 +0000: Fetching VSAN disk info from cs-ie-h03.ie.local (may take a moment) ...
2015-02-27 12:12:02 +0000: Fetching VSAN disk info from cs-ie-h02.ie.local (may take a moment) ...
2015-02-27 12:12:02 +0000: Fetching VSAN disk info from cs-ie-h04.ie.local (may take a moment) ...
2015-02-27 12:12:05 +0000: Done fetching VSAN disk infos
```

DisplayName	Host	isSSD	Num Comp	Capacity Total	Used	Reserved	Status Health
naa.600508b1001c61cedd42b0c3fbf55132	cs-ie-h01.ie.local	SSD	0	186.27 GB	0 %	0 %	OK (v2)
naa.600508b1001c3ea7838c0436dbe6d7a2	cs-ie-h01.ie.local	MD	18	136.44 GB	67 %	67 %	OK (v2)
naa.600508b1001ccd5d506e7ed19c40a64c	cs-ie-h01.ie.local	MD	12	136.44 GB	60 %	59 %	OK (v2)
naa.600508b1001c388c92e817e43fcd5237	cs-ie-h01.ie.local	MD	32	136.44 GB	66 %	65 %	OK (v2)
naa.600508b1001c79748e8465571b6f4a46	cs-ie-h01.ie.local	MD	12	136.44 GB	63 %	63 %	OK (v2)
naa.600508b1001c16be6e256767284eaf88	cs-ie-h01.ie.local	MD	12	136.44 GB	67 %	67 %	OK (v2)
naa.600508b1001c2ee9a6446e708105054b	cs-ie-h01.ie.local	MD	12	136.44 GB	67 %	66 %	OK (v2)
naa.600508b1001c64816271482a56a48c3c	cs-ie-h01.ie.local	MD	13	136.44 GB	66 %	66 %	OK (v2)
naa.600508b1001c64b76c8ceb56e816a89d	cs-ie-h02.ie.local	SSD	0	186.27 GB	0 %	0 %	OK (v2)
naa.600508b1001c0cc0ba2a3866cf8e28be	cs-ie-h02.ie.local	MD	12	136.44 GB	64 %	64 %	OK (v2)
naa.600508b1001c19335174d82278dee603	cs-ie-h02.ie.local	MD	12	136.44 GB	69 %	68 %	OK (v2)
naa.600508b1001cb2234d6fff4f7b1144f59	cs-ie-h02.ie.local	MD	13	136.44 GB	72 %	13 %	OK (v2)
naa.600508b1001c07d52529e83da9541bf	cs-ie-h02.ie.local	MD	21	136.44 GB	46 %	36 %	OK (v2)
naa.600508b1001ca36381622ca880f3aacd	cs-ie-h02.ie.local	MD	17	136.44 GB	54 %	52 %	OK (v2)
naa.600508b1001c9c8b5f6f0d7a2be44433	cs-ie-h03.ie.local	SSD	0	186.27 GB	0 %	0 %	OK (v2)
naa.600508b1001ceefc4213ceb9b51c4be4	cs-ie-h03.ie.local	MD	15	136.44 GB	69 %	54 %	OK (v2)
naa.600508b1001c1a7f310269ccd51a4e83	cs-ie-h03.ie.local	MD	16	136.44 GB	59 %	44 %	OK (v2)
naa.600508b1001c2b7a3d39534ac6beb92d	cs-ie-h03.ie.local	MD	13	136.44 GB	66 %	66 %	OK (v2)
naa.600508b1001cd259ab7ef213c87eaad7	cs-ie-h03.ie.local	MD	16	136.44 GB	53 %	52 %	OK (v2)
naa.600508b1001cb11f3292fe743a0fd2e7	cs-ie-h03.ie.local	MD	11	136.44 GB	60 %	60 %	OK (v2)
naa.600508b1001c9b93053e6dc3ea9bf3ef	cs-ie-h03.ie.local	MD	10	136.44 GB	76 %	76 %	OK (v2)
naa.600508b1001c29d8145d6cc1925e9fb9	cs-ie-h04.ie.local	SSD	0	186.27 GB	0 %	0 %	OK (v2)
naa.600508b1001c846c000c3d9114ed71b3	cs-ie-h04.ie.local	MD	15	136.44 GB	63 %	54 %	OK (v2)
naa.600508b1001c6a664d5d576299cec941	cs-ie-h04.ie.local	MD	10	136.44 GB	63 %	55 %	OK (v2)
naa.600508b1001c4b820b4d80f9f8acfa95	cs-ie-h04.ie.local	MD	14	136.44 GB	73 %	66 %	OK (v2)
naa.600508b1001cadff5d80ba7665b8f09a	cs-ie-h04.ie.local	MD	27	136.44 GB	44 %	42 %	OK (v2)
naa.600508b1001c258181f0a088f6e40dab	cs-ie-h04.ie.local	MD	11	136.44 GB	74 %	74 %	OK (v2)
naa.600508b1001c51f3a696fe0bbbcb5096	cs-ie-h04.ie.local	MD	9	136.44 GB	66 %	66 %	OK (v2)

vsan.enable_vsan_on_cluster

Similar to how the `vsan.disable_vsan_on_cluster` command behaves, this command does the opposite and enables Virtual SAN on a cluster. It takes a cluster as an argument.

Note: The `-e`, option to enable checksum enforcement, is reserved for *future use*, when 520 byte sector disk drives are supported with Virtual SAN.

Usage:

```
vsan.enable_vsan_on_cluster {cluster} {-d, --disable-storage-auto-claim}
                               {-e, --enable-vm-checksum-enforcement}
                               {-h, --help}
```

Examples:

- Display help:

```
> vsan.enable_vsan_on_cluster -h
usage: enable_vsan_on_cluster [opts] cluster
Enable VSAN on a cluster
  cluster: Path to a ClusterComputeResource
           --disable-storage-auto-claim, -d:  Disable auto disk-claim
           --enable-vm-checksum-enforcement, -e:  enable vsan checksum enforcement
           --help, -h:  Show this message
```

- Enable VSAN on a cluster:

```
> vsan.enable_vsan_on_cluster 0
ReconfigureComputeResource VSAN60: success
cs-ie-h01.ie.local: success
cs-ie-h02.ie.local: success
cs-ie-h03.ie.local: success
cs-ie-h04.ie.local: success
>
```


vsan.enter_maintenance mode

Place a host into maintenance mode. It also provides the ability through the '-e' option to evacuate powered off VMs, and also offers the three data evacuation choices that are offered to administrators when they choose to enter maintenance mode via the vSphere web client. These choices are:

- ensureObjectAccessibility
- evacuateAllData
- noAction

These different options and the different maintenance mode behaviors are explained in detailed in the *Virtual SAN 6.0 Administrators Guide*.

If there are running VMs on the host, DRS must be enabled so that the VMs are automatically vMotion'ed from the host that is being placed into maintenance mode. If DRS is not enabled, administrators will have to manually migrate the VMs before the host can successfully enter maintenance mode.

Usage:

```
vsan.enter_maintenance_mode {cluster} {-t, --timeout}
                             {-e, --evacuate-powered-off-vm}
                             {-n, --no-wait} {-v, --vsan-mode} {-h, --help}
```

Examples:

- Display help:

```
> vsan.enter_maintenance_mode -h
usage: enter_maintenance_mode [opts] host...
Put hosts into maintenance mode
Choices for vsan-mode: ensureObjectAccessibility, evacuateAllData, noAction
host: Path to a HostSystem
      --timeout, -t <i>:   Timeout (in seconds) (default: 0)
--evacuate-powered-off-vm, -e:   Evacuate powered off vms
      --no-wait, -n:       Don't wait for Task to complete
      --vsan-mode, -v <s>:   Actions to take for VSAN backed storage
                             (default: ensureObjectAccessibility)
      --help, -h:         Show this message
```

- Enter maintenance mode, explicitly specifying *ensureObjectAccessibility*. Here is an example of placing a host into maintenance mode, not evacuating powered off VMs, and selecting *ensureObjectAccessibility* (even though that is already the default action).

```
> vsan.enter_maintenance_mode 3 -v ensureObjectAccessibility  
EnterMaintenanceMode cs-ie-h04.ie.local: success
```

When the operation succeeds, it reports success. Note that Virtual SAN didn't change exiting maintenance mode in any way, so the RVC command `host.exit_maintenance_mode` can be used. There is no VSAN specific maintenance mode exit command.

Other options that can be included with this command are:

- Setting a timeout on the enter maintenance mode operation (-t)
- Returning from the command immediately without waiting for the task to complete (-n)

vsan.fix_renamed_vms

There have been occasions where, after an outage, virtual machines get referred to by their full path name rather than their actual names. If storage inaccessibility occurs, it is possible for vCenter server to rename VMs to their individual vmx file paths (e.g. "/vmfs/volumes/vsanDatastore/foo/foo.vmx").

At the current time, the original name of the VM is irretrievable so the command sets the name of the virtual machines to the name of the .vmx file. The `-help (-h)` option to the command provides additional details.

Usage:

```
vsan.fix_renames_vms {cluster} {vms} {-f, --force} {-h, --help}
```

Examples:

- Display help:

```
> vsan.fix_renamed_vms -h
usage: fix_renamed_vms [opts] vms...
This command can be used to rename some VMs which get renamed by the VC in
case of storage inaccessibility. It is possible for some VMs to get renamed to
vmx file path. eg. "/vmfs/volumes/vsanDatastore/foo/foo.vmx". This command
will rename this VM to "foo". This is the best we can do. This VM may have
been named something else but we have no way to know. In this
best effort command, we simply rename it to the name of its config file
(without the full path and .vmx extension ofcourse!).
vms: Path to a VirtualMachine
--force, -f: Force to fix name
--help, -h: Show this message
```

This time the argument required is a virtual machine, not a host or a cluster. However, just like hosts and clusters, you can navigate to the VMs folder and use the numeric reference for a virtual machine.

vsan.host_claim_disks_differently

Tag a particular device or set as devices as a flash device, magnetic disk or capacity device. Tagging devices as flash devices is often necessary with SAS controllers in RAID-0 mode as they may hide the characteristics of devices from ESXi, including the fact that these devices are SSDs and not magnetic disks. If you mistakenly tag the incorrect device as flash, you can easily re-tag them back as magnetic disks (HDD).

Note that this functionality is now in the vSphere web client UI in vSphere 6.0.

Another use of this command is only applicable to all-flash Virtual SAN configurations (AF-VSAN). With AF-VSAN, the capacity layer is made up of flash devices. If all flash devices used for the capacity layer are a common model, this command enables all devices of a particular model to be tagged as capacity devices for AF-VSAN.

Usage:

```
vsan.host_claim_disks_differently {host} {-m, --model} {-d, --disk}
                                {-c, --claim-type} {-h, --help}
```

Examples:

- Display help:

```
> vsan.host_claim_disks_differently -h
usage: host_claim_disks_differently [opts] hosts...
Tags all devices of a certain model as certain type of device
  hosts: Path to a HostSystem
  --model, -m <s>:   Model of disk to be claimed as capacity tier
  --disk, -d <s>:   Disk name to be claimed as capacity tier
  --claim-type, -c <s>: Claim types: capacity_flash, hdd, ssd
  --help, -h:      Show this message
```

vsan.host_consume_disks

Allows Virtual SAN to consume disks on a host. In some cases, disks might be marked as remote and cannot be automatically consumed by Virtual SAN, which required disks to be marked as local.

This commands overcomes this issue and allows a host to consume disks, even though they may not be presented in a way for Virtual SAN to automatically consume.

Usage:

```
vsan.host_consume_disks {host} {-f, --filter-ssd-by-model}
                               {-i, --filter-hdd-by-model} {-h, --help}
```

Examples:

- Display help:

```
> vsan.host_consume_disks -h
usage: host_consume_disks [opts] host_or_cluster...
Consumes all eligible disks on a host
  host_or_cluster: Path to a ComputeResource or HostSystem
  --filter-ssd-by-model, -f <s>:  Regex to apply as SSD model filter
  --filter-hdd-by-model, -i <s>:  Regex to apply as HDD model filter
  --help, -h:  Show this message
>
```

vsan.host_evacuate_data

This command is the data evacuation part of entering maintenance mode, but it does not do any of the compute/vSphere HA/etc. checks that one gets with maintenance mode.

The command will evacuate the data on the host and ensure that VM objects are rebuilt elsewhere in the cluster to maintain full redundancy. However, this can be overridden with the “--allow-reduced-redundancy” option, which evacuates the host, but doesn’t initiate a rebuild due to lack of resources, i.e. a 3 ==node cluster.

Another option is “--no-action”, which will run the command but not actually evacuate the host.

Usage:

```
vsan.host_evacuate_data {host} {-a, --allow-reduced-redundancy}
                        {-n, --no-action} {-t, --time-out}
                        {-h, --help}
```

Examples:

- Display help:

```
> vsan.host_evacuate_data -h
usage: host_evacuate_data [opts] hosts...
Evacuate hosts from VSAN cluster
  hosts: Path to a HostSystem
  --allow-reduced-redundancy, -a: Removes the need for nodes worth of free
                                   space, by allowing reduced redundancy
  --no-action, -n: Do not evacuate data during host evacuation
  --time-out, -t <i>: Time out for single node evacuation
                       (default: 0)
  --help, -h: Show this message
```

vsan.host_exit_evacuation

This command exits the host evacuation state, and allows the disks on the host in question to be reused for virtual machine objects.

Usage:

```
vsan.host_exit_evacuation {host} {-h, --help}
```

Examples:

- Display help:

```
> vsan.host_exit_evacuation -h
usage: host_exit_evacuation hosts...
Exit hosts' evacuation, bring them back to VSAN cluster as data containers
hosts: Path to a HostSystem
--help, -h: Show this message
```

vsan.host_info

This command produces detailed information for about a host in the Virtual SAN cluster.

Usage:

```
vsan.host_info {host} {-h, --help}
```

Examples:

- Display help:

```
> vsan.host_info -h
usage: host_info host
Print VSAN info about a host
host: Path to a HostSystem
--help, -h: Show this message
```

- Display information about a host in the Virtual SAN cluster:

```
> vsan.host_info 0
2015-02-27 14:04:27 +0000: Fetching host info from cs-ie-h01.ie.local (may take a
moment) ...
Product: VMware ESXi 6.0.0 build-2391873
VSAN enabled: yes
Cluster info:
Cluster role: master
Cluster UUID: 529ccbe4-81d2-89bc-7a70-a9c69bd23a19
Node UUID: 545ca9af-ff4b-fc84-dcee-001f29595f9f
Member UUIDs: ["545ca9af-ff4b-fc84-dcee-001f29595f9f", "54188e3a-84fd-9a38-23ba-
001b21168828", "5460b129-4084-7550-46e1-0010185def78", "54196e13-7f5f-cba8-5bac-
001517a69c72"] (4)
Node evacuated: no
Storage info:
Auto claim: yes
Checksum enforced: no
Disk Mappings:
SSD: HP Serial Attached SCSI Disk (naa.600508b1001c61cedd42b0c3fbf55132) - 186 GB, v2
MD: HP Serial Attached SCSI Disk (naa.600508b1001c16be6e256767284eaf88) - 136 GB, v2
MD: HP Serial Attached SCSI Disk (naa.600508b1001c64816271482a56a48c3c) - 136 GB, v2
MD: HP Serial Attached SCSI Disk (naa.600508b1001c388c92e817e43fcd5237) - 136 GB, v2
MD: HP Serial Attached SCSI Disk (naa.600508b1001ccd5d506e7ed19c40a64c) - 136 GB, v2
MD: HP Serial Attached SCSI Disk (naa.600508b1001c79748e8465571b6f4a46) - 136 GB, v2
MD: HP Serial Attached SCSI Disk (naa.600508b1001c2ee9a6446e708105054b) - 136 GB, v2
MD: HP Serial Attached SCSI Disk (naa.600508b1001c3ea7838c0436dbe6d7a2) - 136 GB, v2
FaultDomainInfo:
Not configured
NetworkInfo:
Adapter: vmk2 (172.32.0.1)
```


vsan.host_wipe_non_vsan_disks

Wipe a disk that was previously used for some other non-Virtual SAN purpose. This is useful if there are other filesystems (e.g. VMFS, FAT, vFRC) and you now wish to repurpose the disk for use by Virtual SAN.

Usage:

```
vsan.host_wipe_non_vsan_disks {host} {-d, --disk} {-f, --force}
                               {-i, --interactive} {-h, --help}
```

Examples:

- Display help:

```
> vsan.host_wipe_non_vsan_disk -h
usage: host_wipe_non_vsan_disk [opts] hosts...
Wipe disks with partitions other than VSAN partitions
hosts: Path to a HostSystem
  --disk, -d <s>:   Disk to be wiped clean (multiple allowed)
  --force, -f:     Do it for real
  --interactive, -i: Select disks to wipe from given disk list, cannot be
                   set together with parameter 'disks'
  --help, -h:     Show this message
>
```

- Here is an attempt to wipe a disk that is actually in use by Virtual SAN:

```
> vsan.host_wipe_non_vsan_disk 0 -d naa.600508b1001c16be6e256767284eaf88
2015-03-02 14:23:38 +0000: Gathering disk information for host cs-ie-
h01.ie.local
2015-03-02 14:23:39 +0000: Done gathering disk information
Disks on host cs-ie-h01.ie.local:
Disk: HP Serial Attached SCSI Disk (naa.600508b1001c16be6e256767284eaf88)
Host: cs-ie-h01.ie.local
Make/Model: HP LOGICAL VOLUME
Type: HDD
Size: 136 GB
Detected to be a VSAN disk, skipping
>
```

vsan.host_wipe_vsan_disks

Wipe a disk that was previously used by Virtual SAN purpose. This is useful if there are VSAN filesystems on the disk and you now wish to repurpose the disk for some other use (e.g. VMFS, vFRC). The command will evacuate the data on the disk before wiping it. However, this can be overridden with the “--allow-reduced-redundancy” option. Another option is “--no-action”, which will run the command but not actually wipe the disk. Note that disks cannot be wiped when auto claim mode is enabled.

Usage:

```
vsan.host_wipe_vsan_disks {host} {-d, --disk} {-i, ---interactive} {-f, --force}
    {-a, --allow-reduced-redundancy}
    {-n,--no-action} {-h, --help}
```

Examples:

- Display help:

```
> vsan.host_wipe_vsan_disks -h
usage: host_wipe_vsan_disks [opts] hosts...
Wipes content of all VSAN disks on hosts, by default wipe all disk groups
hosts: Path to a HostSystem
    --disk, -d <s>:   Disk's canonical name, as identifier of
                    disk to be wiped
    --interactive, -i: Select disks to wipe from given disk list,
                    cannot be set together with parameter
                    'disks'
    --allow-reduced-redundancy, -a: Removes the need for disks worth of free
                    space, by allowing reduced redundancy
                    during disk wiping
    --no-action, -n:   Take no action to protect data during disk
                    wiping
    --force, -f:       Forcely wipe disks without any
                    confirmation
    --help, -h:       Show this message
```

- Wipe a disk clean (prevented due to auto claim mode on):

```
> vsan.host_wipe_vsan_disks 0 -d naa.600508b1001c16be6e256767284eaf88
2015-03-02 14:14:29 +0000: Checking status on host cs-ie-h01.ie.local

Disks cannot be wiped when storage auto claim mode is enabled
Please disable it and try again
Wipe disk operation is aborted
>
```

- Wipe a disk clean (evacuate all data):

```
> vsan.host_wipe_vsan_disks 0 -d naa.600508b1001c16be6e256767284eaf88
2015-03-02 14:16:29 +0000: Checking status on host cs-ie-h01.ie.local
2015-03-02 14:16:29 +0000: Done checking status on host cs-ie-h01.ie.local

Disks to be wiped:
+-----+-----+-----+-----+
| Index | DisplayName                               | isSSD |
+-----+-----+-----+-----+
| 1     | naa.600508b1001c16be6e256767284eaf88 | MD    |
+-----+-----+-----+-----+
2015-03-02 14:16:29 +0000: Data evacuation mode during disk wiping:
evacuateAllData
All data will be evacuated to other disks, to keep data's
integrity and compliance
Are you willing to wipe above disks?[Y/N]
```

- Wipe a disk clean (ensure object accessibility):

```
> vsan.host_wipe_vsan_disks 0 -d naa.600508b1001c16be6e256767284eaf88 --allow-
reduced-redundancy
2015-03-02 14:19:00 +0000: Checking status on host cs-ie-h01.ie.local
2015-03-02 14:19:00 +0000: Done checking status on host cs-ie-h01.ie.local

Disks to be wiped:
+-----+-----+-----+-----+
| Index | DisplayName                               | isSSD |
+-----+-----+-----+-----+
| 1     | naa.600508b1001c16be6e256767284eaf88 | MD    |
+-----+-----+-----+-----+
2015-03-02 14:19:00 +0000: Data evacuation mode during disk wiping:
ensureObjectAccessibility
Data compliance may be broken, to speed up data evacuation,
but data won't get lost
Are you willing to wipe above disks?[Y/N]
```

vsan.lldpnetmap

This command takes either a host or a cluster as an argument. If there are non-Cisco switches with Link Layer Discovery Protocol (LLDP) enabled in the environment, there is an RVC command to display uplink <-> switch <-> switch port information.

Usage:

```
vsan.lldpnetmap {host|cluster} {-h, --help}
```

Examples:

- Display help:

```
> vsan.lldpnetmap -h
usage: lldpnetmap hosts_and_clusters...
Gather LLDP mapping information from a set of hosts
  hosts_and_clusters: Path to a HostSystem or ClusterComputeResource
                    --help, -h: Show this message
>
```

- Display the network information from LLDP. This is extremely useful for determining which hosts are attached to which switches when the Virtual SAN Cluster is spanning multiple switches. It may help to isolate a problem to a particular switch when only a subset of the hosts in the cluster is impacted.

```
> vsan.lldpnetmap 0
2013-08-15 19:34:18 -0700: This operation will take 30-60 seconds ...
+-----+-----+
| Host           | LLDP info           |
+-----+-----+
| 10.143.188.54  | w2r13-vsah-x650-2:  |
|                | w2r13-vsah-x650-1:  |
+-----+-----+
>
```

This is only available with non-Cisco switches that support LLDP. For Cisco switches, which do not support LLDP but which use their own CDP (Cisco Discovery Protocol), there is no RVC command

vsan.object_status_report

This command verifies the health of the Virtual SAN cluster. When all objects are in a known good state, it is expected that this command return no issues. With absent components however, this command provides details on the missing components.

Usage:

```
vsan.object_status_report {host|cluster} {-h, --help}
```

Examples:

- Display help:

```
> vsan.obj_status_report -h
usage: obj_status_report [opts] cluster_or_host...
Print component status for objects in the cluster.
  cluster_or_host: Path to a ClusterComputeResource or HostSystem
  --print-table, -t: Print a table of object and their status,
                    default all objects
  --filter-table, -f <s>: Filter the obj table based on status displayed
                        in histogram, e.g. 2/3
  --print-uuids, -u: In the table, print object UUIDs instead of
                    vmdk and vm paths
  --ignore-node-uuid, -i <s>: Estimate the status of objects if all comps on
                              a given host were healthy.
  --help, -h: Show this message
```

- Display a report on the state of all the objects in a cluster:

```
> vsan.obj_status_report 0
2015-02-27 16:00:37 +0000: Querying all VMs on VSAN ...
2015-02-27 16:00:38 +0000: Querying all objects in the system from cs-ie-
h01.ie.local ...
2015-02-27 16:00:38 +0000: Querying all disks in the system from cs-ie-
h01.ie.local ...
2015-02-27 16:00:39 +0000: Querying all components in the system from cs-ie-
h01.ie.local ...
2015-02-27 16:00:39 +0000: Querying all object versions in the system ...
2015-02-27 16:00:40 +0000: Got all the info, computing table ...
```

Histogram of component health for non-orphaned objects

```
+-----+-----+
| Num Healthy Comps / Total Num Comps | Num objects with such status |
+-----+-----+
| 10/10 (OK) | 2 |
| 3/3 (OK) | 97 |
| 8/8 (OK) | 2 |
| 4/4 (OK) | 2 |
| 7/7 (OK) | 1 |
```

```

| 5/5 (OK) | 1 |
| 6/6 (OK) | 1 |
+-----+-----+
Total non-orphans: 106

```

Histogram of component health for possibly orphaned objects

```

+-----+-----+
| Num Healthy Comps / Total Num Comps | Num objects with such status |
+-----+-----+
+-----+-----+
Total orphans: 0

```

```

Total v1 objects: 0
Total v2 objects: 106

```

The output should be read as follows:

- There are 106 objects in this Virtual SAN cluster.
- There are no orphaned objects, which is good.
- There are 2 objects that are made up of 10 components, and all 10 are healthy.
- There are 97 objects that are made up of 3 components, and all 3 are healthy.
- There are 2 objects that are made up of 8 components, and all 8 are healthy.
- There are 2 objects that are made up of 4 components, and all 4 are healthy.
- There is 1 objects that are made up of 7 components, and all 7 are healthy.
- There is 1 objects that are made up of 5 components, and all 5 are healthy.
- There is 1 objects that are made up of 6 components, and all 6 are healthy.
- All 106 objects are v2 objects, meaning that they have been updated for the new v2 on-disk format.

vsan.object_info

DOM, the Distributed Object Manager, is a core component of Virtual SAN that implements the RAID configuration. Using this “DOM object” uuid, one can ask Virtual SAN to display detailed information about the object via this command. For every component the physical location (host, SSD, HDD) is shown, along with operational states. The output also displays information about the VM Storage Policy in use by the object. For example:

- `forceProvisioning` – if set to 1, Force Provisioning is in use
- `hostFailuresToTolerate` – represents *NumberOfFailuresToTolerate*
- `proportionalCapacity` – represents *ObjectSpaceReservation*
- `cacheReservation` – represents *FlashReadCacheReservation*
- `StripeWidth` – represents *NumberOfDiskObjectsToStripe*

Usage:

```
vsan.object_info {cluster} {object_uuid} {-s, --skip-ext-attr}
                {-i, --include-detailed-usage} {-h, --help}
```

Examples:

- Display help:

```
> vsan.object_info -h
usage: object_info [opts] cluster obj_uuid...
Fetch information about a VSAN object
  cluster: Cluster on which to fetch the object info
  obj_uuid:
    --skip-ext-attr, -s:  Don't fetch extended attributes
    --include-detailed-usage, -i:  Include detailed usage info
    --help, -h:  Show this message
>
```

- Display object information. The object uuid can be found from the output of a `vsan.vm_object_info` command output. Note the DOM Object line contains a v2, stating that this is a v2 object.

```
> vsan.object_info 0 b4eb8a54-d47c-cd2d-4dae-001b21168828
  DOM Object: b4eb8a54-d47c-cd2d-4dae-001b21168828 (v2, owner: cs-ie-h01.ie.local, policy:
forceProvisioning = 0, hostFailuresToTolerate = 1, spbmProfileId = aa6d5a82-1c88-45da-
85d3-3d74b91a5bad, proportionalCapacity = [0, 100], spbmProfileGenerationNumber = 0,
cacheReservation = 0, stripeWidth = 1)
  RAID_1
    Component: 1986db54-5299-0ec1-1e0f-0010185def78 (state: ACTIVE (5), host: cs-ie-
h01.ie.local, md: naa.600508b1001c388c92e817e43fcd5237, ssd:
naa.600508b1001c61cedd42b0c3fbf55132,
                                usage: 0.4 GB)
    Component: f188db54-f210-2706-80eb-0010185def78 (state: ACTIVE (5), host: cs-ie-
h04.ie.local, md: naa.600508b1001cadff5d80ba7665b8f09a, ssd:
naa.600508b1001c29d8145d6cc1925e9fb9,
                                usage: 0.4 GB)
    Witness: 2e89db54-7647-1e74-2da2-0010185def78 (state: ACTIVE (5), host: cs-ie-
h03.ie.local, md: naa.600508b1001c9b93053e6dc3ea9bf3ef, ssd:
naa.600508b1001c9c8b5f6f0d7a2be44433,
                                usage: 0.0 GB)
  Extended attributes:
    Address space: 273804165120B (255.00 GB)
    Object class: vmnamespace
    Object path: /vmfs/volumes/vsan:529ccbe481d289bc-7a70a9c69bd23a19/
```


vsan.object_reconfigure

Configure an object with a new policy. The policy settings can be one or more of the following:

```

("hostFailuresToTolerate")
("forceProvisioning")
("stripeWidth")
("proportionalCapacity")
("cacheReservation")

```

These policy settings take an integer argument that is specified as i0, i1, i2 and so on. The syntax is rather complex. Please refer to the examples below for guidance. Note that this leads to VM Storage Policy being “out of date” with regards to the state of this object. The command completes when the reconfiguration has been acknowledged by Virtual SAN, but it doesn't wait for the object to be compliant with the policy. Use `vsan.resync_dashboard` or `vsan.object_info` to monitor the reconfiguration happening in the background.

Usage:

```

vsan.object_reconfigure {cluster} {object uuid} {policy }
                        {-h, --help}

```

Examples:

- Display help:

```

> vsan.object_reconfigure -h
usage: object_reconfigure [opts] cluster obj_uuid...
Reconfigure a VSAN object
      cluster: Cluster on which to execute the reconfig
      obj_uuid: Object UUID
--policy, -p <s>:   New policy
--help, -h:       Show this message

```

- Reconfigure the policy of an object to FTT=1:

```

> vsan.object_reconfigure 0 b4eb8a54-d47c-cd2d-4dae-001b21168828 --policy "(\"
hostFailuresToTolerate \" i1)\"
Reconfiguring 'b4eb8a54-d47c-cd2d-4dae-001b21168828' to ("
hostFailuresToTolerate " i1)

All reconfigs initiated. Synching operation may be happening in the background
>

```

vsan.observer

The VMware Virtual SAN Observer is a monitoring and troubleshooting tool for Virtual SAN. The tool is launched from RVC and can be utilized for monitoring performance statistics for Virtual SAN live mode or offline. When running in live mode, a web browser can be pointed at vCenter Server to see live graphs related to the performance of Virtual SAN.

The utility can be used to understand Virtual SAN performance characteristics. The utility is intended to provide deeper insights of Virtual SAN performance characteristics and analytics.

VSAN Observer needs a number of arguments supplied at the command line, and can be run in both live monitoring mode or offline/log gathering mode. Here is the list of options available in version 6.0:

```
> vsan.observer -h
usage: observer [opts] cluster_or_host...
Run observer
  cluster_or_host: Path to a ClusterComputeResource or HostSystem
                   --filename, -f <s>:   Output file path
                   --port, -p <i>:       Port on which to run webserver (default:
8010)
                   --run-webserver, -r:   Run a webserver to view live stats
                   --force, -o:           Apply force
  --keep-observation-in-memory, -k:       Keep observed stats in memory even when
commands ends. Allows to resume later
  --generate-html-bundle, -g <s>:        Generates an HTML bundle after
completion. Pass a location
                   --interval, -i <i>:    Interval (in sec) in which to collect
stats (default: 60)
  --max-runtime, -m <i>:                 Maximum number of hours to collect stats.
Caps memory usage. (Default: 2)
  --forever, -e <s>:                     Runs until stopped. Every --max-runtime
intervals retires snapshot to disk. Pass
a location
                   --no-https, -n:        Don't use HTTPS and don't require login.
Warning: Insecure
  --max-diskspace-gb, -a <i>:            Maximum disk space (in GB) to use in
forever mode. Deletes old data
periodically (default: 5)
                   --help, -h:           Show this message
```

Further discussions on the use of `vsan.observer` are outside the scope of this document. For details on how to get started with `vsan.observer` and how it can be used for troubleshooting performance in Virtual SAN environments, please refer to the *Virtual SAN 6.0 Troubleshooting Reference Manual*.

vsan.observer_process_statsfile

This command converts a JSON dump captured from VSAN observer command with “`--generate-html-bundle`” option to HTML. This HTML can then be used for troubleshooting performance issues offline.

```
> vsan.observer_process_statsfile -h
usage: observer_process_statsfile [opts] statsfile outputpath
Analyze an offline observer stats file and produce static HTML
    statsfile:
    outputpath:
--max-traces, -m <i>:    Only process this many traces
--help, -h:             Show this message
```

Further discussions on the use of `vsan.observer_process_statsfile` are outside the scope of this document. For details on how to use `vsan.observer_process_statsfile` and how it can be used for troubleshooting performance in Virtual SAN environments, please refer to the *Virtual SAN 6.0 Troubleshooting Reference Manual*.

vsan.proactive_rebalance

This is a manual rebalance command that looks at the distribution of components around the cluster, and will proactively begin to balance the distribution of components around the cluster. Otherwise rebalancing only begins to occur when a physical disk reached 80% capacity.

Proactive rebalance is not running by default. An administrator will have to initiate the proactive balancing of components with the `--start` option.

Usage:

```
vsan.proactive_rebalance {cluster} {-s, --start} {-t, --time-span}
                          {-v, --variance-threshold} {-i, --time-threshold}
                          {-r, --rate-threshold} {-o, --stop}
                          {-h, --help}
```

Examples:

- Display help:

```
> vsan.proactive_rebalance -h
usage: proactive_rebalance [opts] cluster
Configure proactive rebalance for Virtual SAN
  cluster: Path to ClusterComputeResource
        --start, -s:      Start proactive rebalance
        --time-span, -t <i>: Determine how long this proactive rebalance
                           lasts in seconds, only be valid when option
                           'start' is specified
        --variance-threshold, -v <f>: Configure the threshold, that only if disk's
                           used_capacity/disk_capacity exceeds this
                           threshold, disk is qualified for proactive
                           rebalance, only be valid when
                           option 'start' is specified
        --time-threshold, -i <i>: Threshold in seconds, that only when
                           variance threshold continuously exceeds this
                           threshold, corresponding disk will be
                           involved to proactive rebalance, only be
                           valid when option 'start' is specified
        --rate-threshold, -r <i>: Determine how many data in MB could be moved
                           per hour for each node, only be valid when
                           option 'start' is specified
        --stop, -o:      Stop proactive rebalance
        --help, -h:     Show this message
```

Some clarity might be needed for the start parameter “--variance-threshold”. The description in the --help output states "Configure the threshold, that only if disk's used capacity divided by disk capacity exceeds this threshold... "

In fact, the trigger condition is only when the following calculation is greater than the <variance_threshold>:

$$\frac{\langle \text{used_capacity_of_this_disk} \rangle}{\langle \text{this_disk_capacity} \rangle} - \frac{\langle \text{used_capacity_of_least_full_disk_in_cluster} \rangle}{\langle \text{least_full_disk_capacity} \rangle}$$

In other words, a disk is qualified for proactive rebalancing only if its fullness (used_capacity/disk_capacity) exceeds the fullness of the "least-full" disk in the vsan cluster by the threshold. The rebalancing process also needs to wait until the <time_threshold> is met under this situation, and then start to try rebalancing.

- Start proactive component balancing:

```
> vsan.proactive_rebalance -s 0
2014-12-11 14:15:05 +0000: Processing Virtual SAN proactive rebalance on host
cs-ie-h02.ie.local ...
2014-12-11 14:15:05 +0000: Processing Virtual SAN proactive rebalance on host
cs-ie-h04.ie.local ...
2014-12-11 14:15:05 +0000: Processing Virtual SAN proactive rebalance on host
cs-ie-h01.ie.local ...
2014-12-11 14:15:05 +0000: Processing Virtual SAN proactive rebalance on host
cs-ie-h03.ie.local ...
```

Proactive rebalance has been started!

- Stop proactive component balancing:

```
> vsan.proactive_rebalance -o 0
2014-12-11 14:15:45 +0000: Processing Virtual SAN proactive rebalance on host
cs-ie-h01.ie.local ...
2014-12-11 14:15:45 +0000: Processing Virtual SAN proactive rebalance on host
cs-ie-h02.ie.local ...
2014-12-11 14:15:45 +0000: Processing Virtual SAN proactive rebalance on host
cs-ie-h04.ie.local ...
2014-12-11 14:15:45 +0000: Processing Virtual SAN proactive rebalance on host
cs-ie-h03.ie.local ...
```

Proactive rebalance has been stopped!

vsan.proactive_rebalance_info

This command, which takes a cluster as an argument displays information about proactive rebalancing activities, including whether or not proactive rebalance is running or not.

Usage:

```
vsan.proactive_rebalance_info {cluster} {-h, --help}
```

Examples:

- Display help:

```
> vsan.proactive_rebalance_info -h
usage: proactive_rebalance_info cluster
Retrieve proactive rebalance status for Virtual SAN
cluster: Path to ClusterComputeResource
--help, -h: Show this message
```

- Get information about proactive rebalancing when it is not running:

```
> vsan.proactive_rebalance_info 0
2014-12-11 14:14:27 +0000: Retrieving proactive rebalance information from
host cs-ie-h02.ie.local ...
2014-12-11 14:14:27 +0000: Retrieving proactive rebalance information from
host cs-ie-h04.ie.local ...
2014-12-11 14:14:27 +0000: Retrieving proactive rebalance information from
host cs-ie-h01.ie.local ...
2014-12-11 14:14:27 +0000: Retrieving proactive rebalance information from
host cs-ie-h03.ie.local ...
```

Proactive rebalance is not running!

```
Max usage difference triggering rebalancing: 30.00%
Average disk usage: 5.00%
Maximum disk usage: 26.00% (21.00% above mean)
Imbalance index: 5.00%
No disk detected to be rebalanced
```

- Get information about proactive rebalancing when it is running:

```
> vsan.proactive_rebalance_info 0
2014-12-11 14:15:11 +0000: Retrieving proactive rebalance information from
host cs-ie-h02 ...
2014-12-11 14:15:11 +0000: Retrieving proactive rebalance information from
host cs-ie-h01 ...
2014-12-11 14:15:11 +0000: Retrieving proactive rebalance information from
host cs-ie-h04 ...
```

```
2014-12-11 14:15:11 +0000: Retrieving proactive rebalance information from
host cs-ie-h03 ...
```

```
Proactive rebalance start: 2014-12-11 14:13:10 UTC
Proactive rebalance stop: 2014-12-12 14:16:17 UTC
Max usage difference triggering rebalancing: 30.00%
Average disk usage: 5.00%
Maximum disk usage: 26.00% (21.00% above mean)
Imbalance index: 5.00%
No disk detected to be rebalanced
```

vsan.purge_inaccessible_vswp_objects

This command should only be run as part of an on-disk format upgrade if the upgrade command, `vsan.v2_ondisk_upgrade`, detects inaccessible VM swap objects. If inaccessible swap object exists, the administrator may use the following command to clean them up.

Usage:

```
vsan.purge_inaccessible_vswp_objects {cluster} {-f, --force} {-h, --help}
```

Examples:

- Display help:

```
> vsan.purge_inaccessible_vswp_objects -h
```

```
usage: purge_inaccessible_vswp_objects [opts] cluster_or_host
Search and delete inaccessible vswp objects on a virtual SAN cluster.
```

VM vswp file is used for memory swapping for running VMs by ESX. In VMware virtual SAN a vswp file is stored as a separate virtual SAN object. When a vswp object goes inaccessible, memory swapping will not be possible and the VM may crash when next time ESX tries to swap the memory for the VM. Deleting the inaccessible vswp object will not make thing worse, but it will eliminate the possibility for the object to regain accessibility in future time if this is just a temporary issue (e.g. due to network failure or planned maintenance).

Due to a known issue in vSphere 5.5, it is possible for Virtual SAN to have done incomplete deletions of vswp objects. In such cases, the majority of components of such objects were deleted while a minority of components were left unavailable (e.g. due to one host being temporarily down at the time of deletion). It is then possible for the minority to resurface and present itself as an inaccessible object because a minority can never gain quorum. Such objects waste space and cause issues for any operations involving data evacuations from hosts or disks. This command employs heuristics to detect this kind of left-over vswp objects in order to delete them.

It will not cause data loss by deleting the vswp object. The vswp object will be regenerated when the VM is powered on next time.

```
cluster_or_host: Path to a ClusterComputeResource or HostSystem
--force, -f: Force to delete the inaccessible vswp objects quietly (no
interactive confirmations)
--help, -h: Show this message
```

Caution: *Extreme caution needs to exercise here because this command will also allow you to force delete non-vswp objects (which may cause a real data loss). If you are not completely sure that this is indeed a vswp object, please contact GSS for support with the upgrade.*

vsan.reapply_vsan_vmknlic_config

There may be instances where network issues were encountered, and then resolved, but Virtual SAN does not learn about the updated network changes. In this situation, the RVC command `vsan.reapply_vsan_vmknlic_config` can help by unbinding Virtual SAN from the VMkernel port, rebinding the Virtual SAN VMkernel port and reapply the Virtual SAN networking configuration.

In rare cases VMware has seen this resolve a situation where a Virtual SAN node lost contact to the rest of the cluster and other troubleshooting had not revealed any underlying network issue.

Use this command after doing the regular Virtual SAN network troubleshooting steps as outlined in the *Virtual SAN 6.0 Troubleshooting Reference Manual*.

Usage:

```
vsan.reapply_vsan_vmknlic_config {host} {-v, --vmknlic} {-d, --dry-run}
                                {-h, --help}
```

Examples:

- Display help:

```
> vsan.reapply_vsan_vmknlic_config -h
usage: reapply_vsan_vmknlic_config [opts] host...
Unbinds and rebinds VSAN to its vmknics
  host: Path to a HostSystem
  --vmknlic, -v <s>: Refresh a specific vmknlic. default is all vmknics
  --dry-run, -d: Do a dry run: Show what changes would be made
  --help, -h: Show this message
```

- There is also an option to do a dry-run of the command to show the changes that would be made:

```
> vsan.reapply_vsan_vmknlic_config -d 2
Host: cs-ie-h03.ie.local
Would reapply config of vmknlic vmk2:
  AgentGroupMulticastAddress: 224.2.3.4
  AgentGroupMulticastPort: 23451
  IPProtocol: IPv4
  InterfaceUUID: a3836354-af89-3093-dc4f-0010185def78
  MasterGroupMulticastAddress: 224.1.2.3
  MasterGroupMulticastPort: 12345
  MulticastTTL: 5
```

- Do an actual run. Without the `--dry-run`, `-d` option, the VSAN VMkernel interface is unbound and rebound.

```
> vsan.reapply_vsan_vmknics_config 1
Host: cs-ie-h02.ie.local
Reapplying config of vmk2:
  AgentGroupMulticastAddress: 224.2.3.4
  AgentGroupMulticastPort: 23451
  IPProtocol: IPv4
  InterfaceUUID: 6a836354-bf24-f157-dda7-001517a69c72
  MasterGroupMulticastAddress: 224.1.2.3
  MasterGroupMulticastPort: 12345
  MulticastTTL: 5
Unbinding VSAN from vmknics vmk2 ...
Rebinding VSAN to vmknics vmk2 ...
>
```

vsan.recover_spbm

This command is used in situations where the vCenter server needs to be reinstalled, and the VM storage policies are lost. While the VMs will continue to run with their policies, the new vCenter server will not know about them. This command will recreate these policies on the new vCenter Server.

Not only will it detect VMs that are missing policy settings, but it will also provide the option to recreate policies on the new vCenter server, as shown here. It takes either a host or a cluster as an argument.

Usage:

```
vsan.recover_spbm {cluster|host} {-d, --dry-run} {-f, --force}
                  {-h, --help}
```

Examples:

- Display help:

```
> vsan.recover_spbm -h
usage: recover_spbm [opts] cluster_or_host
SPBM Recovery
  cluster_or_host: Path to a ClusterComputeResource or HostSystem
  --dry-run, -d:   Don't take any automated actions
  --force, -f:     Answer all question with 'yes'
  --help, -h:     Show this message
```

- Recover policies from VMs and apply to the current vCenter server:

```
> vsan.recover_spbm 0
2014-12-02 14:54:02 +0000: Fetching Host info
2014-12-02 14:54:02 +0000: Fetching Datastore info
2014-12-02 14:54:02 +0000: Fetching VM properties
2014-12-02 14:54:02 +0000: Fetching policies used on VSAN from CMMDS
2014-12-02 14:54:03 +0000: Fetching SPBM profiles
2014-12-02 14:54:04 +0000: Fetching VM <-> SPBM profile association
2014-12-02 14:54:04 +0000: Computing which VMs do not have a SPBM Profile ...
2014-12-02 14:54:04 +0000: Fetching additional info about some VMs
2014-12-02 14:54:04 +0000: Got all info, computing after 1.92 sec
2014-12-02 14:54:04 +0000: Done computing
SPBM Profiles used by VSAN:
+-----+-----+
| SPBM ID | policy |
+-----+-----+
| Existing SPBM Profile: | stripeWidth: 1 | |
| Virtual SAN Default Storage Policy | cacheReservation: 0 |
| | | proportionalCapacity: 0 |
| | | hostFailuresToTolerate: 1 |
| | | forceProvisioning: 0 |
+-----+-----+
| Existing SPBM Profile: | stripeWidth: 1 |
```

RVC Command Reference Guide for Virtual SAN

```
| Virtual SAN Default Storage Policy      | cacheReservation: 0      |
|                                          | proportionalCapacity: 0 |
|                                          | hostFailuresToTolerate: 1 |
|                                          | forceProvisioning: 0    |
+-----+-----+-----+-----+
| Unknown SPBM Profile. UUID:            | hostFailuresToTolerate: 1 |
| 5810fe86-6f0f-4718-835d-ce30ff4e0975-gen0 |                          |
+-----+-----+-----+-----+
```

Recreate missing SPBM Profiles using following RVC commands:
spbm.profile_create --rule VSAN.hostFailuresToTolerate=1 5810fe86-6f0f-4718-835d-
ce30ff4e0975-gen0

Do you want to create SPBM Profiles now? [Y/N]

Y

Running: spbm.profile_create --rule VSAN.hostFailuresToTolerate=1 5810fe86-6f0f-4718-
835d-ce30ff4e0975-gen0

Please rerun the command to fix up any missing VM <-> SPBM Profile associations
>

vsan.resync_dashboard

The command `vsan.resync_dashboard` will display the re-syncing of the components that are being rebuilt elsewhere in the cluster. Using this command, it is possible to tell how many bytes are left to sync for that particular VM/Object. The command displays an overview of the resync/rebuild for a snapshot in time. To get a sense of resync/rebuild progress, either run the command multiple times, or use the `--refresh-rate` parameter to display an updated table at a fixed time interval.

Usage:

```
vsan.resync_dashboard {cluster|host} {-r, --refresh-rate} {-h, --help}
```

Examples:

- Display help:

```
> vsan.resync_dashboard -h
usage: resync_dashboard [opts] cluster_or_host
Resyncing dashboard
cluster_or_host: Path to a ClusterComputeResource or HostSystem
--refresh-rate, -r <i>: Refresh interval (in sec). Default is no refresh
--help, -h: Show this message
```

- Display current synchronization information (in this example, nothing is synchronizing):

```
> vsan.resync_dashboard 0
2014-11-06 12:07:45 +0000: Querying all VMs on VSAN ...
2014-11-06 12:07:45 +0000: Querying all objects in the system from cs-ie-
h01.ie.local ...
2014-11-06 12:07:45 +0000: Got all the info, computing table ...
+-----+-----+-----+
| VM/Object | Syncing objects | Bytes to sync |
+-----+-----+-----+
+-----+-----+-----+
| Total     | 0                | 0.00 GB       |
+-----+-----+-----+
```

vsan.scrubber_info

For every host, the command will list each VM and its disks, and display several metrics related to the Virtual SAN background task known as "scrubbing".

Scrubbing is responsible for periodically reading through the entire address space of every object stored on Virtual SAN, for the purpose of finding latent sector errors on the physical disks backing VSAN.

Note that this is a background task performed by Virtual SAN automatically. It is running quite slowly in order to not impact production workloads. This command is intended to give some insight into this background task.

This command is reserved for *future use*. There is no scrubber task in Virtual SAN 6.0.

Usage:

```
vsan.scrubber {cluster|hosts} {-h, --help}
```

Examples:

- Display help:

```
> vsan.scrubber_info -h
usage: scrubber_info cluster_or_hosts...
Print scrubber info about objects on this host or cluster
cluster_or_hosts: Path to a HostSystem or ClusterComputeResource
--help, -h: Show this message
```

vsan.support_information

This command generates a support bundles that includes the output of many RVC commands. This is extremely useful to the technical support personnel at VMware. Typically the Virtual SAN cluster will be provided as an argument to the command, but a vCenter or a datacenter may also be provided. You should only run this command when requested by VMware technical support. The goal is to generate a comprehensive output that can be sent to VMware Support so that a lot of information provided in RVC is readily available to engineers at VMware as part of a support request.

Usage:

```
vsan.support_information {cluster|DC|vCenter} {-h, --help}
```

Examples:

- Display help:

```
> vsan.support_information -h
usage: support_information dc_or_clust_conn
Command to collect vsan support information
  dc_or_clust_conn: Path to a RbVmomi::VIM or Datacenter or
ClusterComputeResource
  --help, -h: Show this message
```

vsan.v2_ondisk_upgrade

This command will rotate through each of the hosts in the Virtual SAN cluster (rolling upgrade), doing a number of verification checks on the state of the host and cluster before evacuating components from each of the disk groups and rebuilding them elsewhere in the cluster. It then upgrades the on-disk format from v1 to v2.

Usage:

```
vsan.v2_ondisk_upgrade {cluster|host} {-i, --ignore-objects}
                        {-d, --downgrade-format}
                        {-a, --allow-reduced-redundancy} {-f, --force}
                        {-h, --help}
```

Examples:

- Display help:

```
> vsan.v2_ondisk_upgrade -h
usage: v2_ondisk_upgrade [opts] hosts_and_clusters...
Upgrade a cluster to VSAN 2.0
  hosts_and_clusters: Path to all HostSystems of cluster or
ClusterComputeResource
    --ignore-objects, -i:   Ignore objects upgrade
    --downgrade-format, -d: Downgrade disk format and file system, be
                           available only if there is no v2 object in
                           VSAN cluster; Virsto will be disabled on
                           given nodes, so no v2 diskgroups
                           can be created.
    --allow-reduced-redundancy, -a: Removes the need for one disk group worth
                           of free space, by allowing reduced
                           redundancy during disk upgrade
    --force, -f:           Automatically answer all confirmation
                           questions with 'proceed'
    --help, -h:           Show this message
```

- Upgrading from v1 on-disk format to v2 on-disk format:

```
/ie-vcsa-03.ie.local/vsan-dc/computers> vsan.v2_ondisk_upgrade 0
+-----+-----+-----+-----+-----+
| Host           | State   | ESX version | v1 Disk-Groups | v2 Disk-Groups |
+-----+-----+-----+-----+-----+
| cs-ie-h02.ie.local | connected | 6.0.0      | 1               | 0               |
| cs-ie-h03.ie.local | connected | 6.0.0      | 1               | 0               |
| cs-ie-h04.ie.local | connected | 6.0.0      | 1               | 0               |
| cs-ie-h01.ie.local | connected | 6.0.0      | 1               | 0               |
+-----+-----+-----+-----+-----+

2014-12-10 14:49:16 +0000: Running precondition checks ...
2014-12-10 14:49:19 +0000: Passed precondition checks
2014-12-10 14:49:19 +0000:
```


RVC Command Reference Guide for Virtual SAN

```
2014-12-10 14:49:19 +0000: Target file system version: v2
2014-12-10 14:49:19 +0000: Disk mapping decommission mode: evacuateAllData
2014-12-10 14:49:28 +0000: Cluster is still in good state, proceeding ...
2014-12-10 14:49:28 +0000: Enabled v2 filesystem as default on host cs-ie-h02.ie.local
2014-12-10 14:49:28 +0000: Removing VSAN disk group on cs-ie-h02.ie.local:
2014-12-10 14:49:28 +0000:   SSD: HP Serial Attached SCSI Disk (naa.600508b1001c64b76c8ceb56e816a89d)
2014-12-10 14:49:28 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c19335174d82278dee603)
2014-12-10 14:49:28 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001ca36381622ca880f3aacd)
2014-12-10 14:49:28 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001cb2234d6ff4f7b1144f59)
2014-12-10 14:49:28 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c0cc0ba2a3866cf8e28be)
2014-12-10 14:49:28 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c07d525259e83da9541bf)
2014-12-10 14:49:28 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c10548f5105fc60246b4a)
RemoveDiskMapping cs-ie-h02.ie.local: success
2014-12-10 15:20:40 +0000: Re-adding disks to VSAN on cs-ie-h02.ie.local:
2014-12-10 15:20:40 +0000:   SSD: HP Serial Attached SCSI Disk (naa.600508b1001c64b76c8ceb56e816a89d)
2014-12-10 15:20:40 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c19335174d82278dee603)
2014-12-10 15:20:40 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001ca36381622ca880f3aacd)
2014-12-10 15:20:40 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001cb2234d6ff4f7b1144f59)
2014-12-10 15:20:40 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c0cc0ba2a3866cf8e28be)
2014-12-10 15:20:40 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c07d525259e83da9541bf)
2014-12-10 15:20:40 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c10548f5105fc60246b4a)
AddDisks cs-ie-h02.ie.local: success
2014-12-10 15:21:13 +0000: Done upgrade host cs-ie-h02.ie.local
2014-12-10 15:21:16 +0000:
2014-12-10 15:21:16 +0000: Cluster is still in good state, proceeding ...
2014-12-10 15:21:16 +0000: Enabled v2 filesystem as default on host cs-ie-h03.ie.local
2014-12-10 15:21:16 +0000: Removing VSAN disk group on cs-ie-h03.ie.local:
2014-12-10 15:21:16 +0000:   SSD: HP Serial Attached SCSI Disk (naa.600508b1001c9c8b5f6f0d7a2be44433)
2014-12-10 15:21:16 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001cceeefc4213ceb9b51c4be4)
2014-12-10 15:21:16 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001cd259ab7ef213c87eaad7)
2014-12-10 15:21:16 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c2b7a3d39534ac6be92d)
2014-12-10 15:21:16 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001cb11f3292fe743a0fd2e7)
2014-12-10 15:21:16 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c1a7f310269ccd51a4e83)
2014-12-10 15:21:16 +0000:   HDD: HP Serial Attached SCSI Disk
(naa.600508b1001c9b93053e6dc3ea9bf3ef)

RemoveDiskMapping cs-ie-h03.ie.local: running
[=====
<<truncated>>
```

- The overall progress of the command can be monitored via RVC, as shown here. Notice that RVC upgrades one disk group at a time. For each disk group upgrade, a disk is first removed from Virtual SAN cluster by evacuating all data from the disk. The format is updated and the disk is then added back to Virtual SAN with the new v2 on-disk format. Once the upgrade is completed successfully, the following message appears:

```
<<<truncated>>>
2014-12-10 16:27:26 +0000: Cluster is still in good state, proceeding ...
2014-12-10 16:27:29 +0000: Enabled v2 filesystem as default on host cs-ie-h01.ie.local
2014-12-10 16:27:29 +0000: Removing VSAN disk group on cs-ie-h01.ie.local:
2014-12-10 16:27:29 +0000:   SSD: HP Serial Attached SCSI Disk (naa.600508b1001c61cedd42b0c3fbf55132)
2014-12-10 16:27:29 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c16be6e256767284eaf88)
2014-12-10 16:27:29 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c64816271482a56a48c3c)
2014-12-10 16:27:29 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c388c92e817e43fcd5237)
2014-12-10 16:27:29 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001ccd5d506e7ed19c40a64c)
2014-12-10 16:27:29 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c79748e8465571b6f4a46)
2014-12-10 16:27:29 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c2ee9a6446e708105054b)
2014-12-10 16:27:29 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c3ea7838c0436dbe6d7a2)
RemoveDiskMapping cs-ie-h01.ie.local: success
2014-12-10 16:52:17 +0000: Re-adding disks to VSAN on cs-ie-h01.ie.local:
2014-12-10 16:52:17 +0000:   SSD: HP Serial Attached SCSI Disk (naa.600508b1001c61cedd42b0c3fbf55132)
2014-12-10 16:52:17 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c16be6e256767284eaf88)
2014-12-10 16:52:17 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c64816271482a56a48c3c)
2014-12-10 16:52:17 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c388c92e817e43fcd5237)
2014-12-10 16:52:17 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001ccd5d506e7ed19c40a64c)
2014-12-10 16:52:17 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c79748e8465571b6f4a46)
2014-12-10 16:52:17 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c2ee9a6446e708105054b)
2014-12-10 16:52:17 +0000:   HDD: HP Serial Attached SCSI Disk (naa.600508b1001c3ea7838c0436dbe6d7a2)
AddDisks cs-ie-h01.ie.local: success
2014-12-10 16:52:58 +0000: Done upgrade host cs-ie-h01.ie.local
2014-12-10 16:52:58 +0000:
2014-12-10 16:52:58 +0000: Done with disk format upgrade phase
2014-12-10 16:52:58 +0000: There are 97 vl objects that require upgrade
```

```
2014-12-10 16:53:04 +0000: Object upgrade progress: 97 upgraded, 0 left
2014-12-10 16:53:04 +0000: Object upgrade completed: 97 upgraded
2014-12-10 16:53:04 +0000: Done VSAN upgrade
/ie-vcsa-03.ie.local/vsan-dc>
```

The `vsan.v2_ondisk_upgrade` has an option called *allow-reduced-redundancy*. To facilitate upgrades when there are not enough resources in the cluster to accommodate disk evacuations. It should be noted that there are risks associated with this approach but unfortunately there is no other way to do the upgrade. For a portion of the upgrade, virtual machines will be running without replica copies of the data, so any failure during the upgrade can lead to virtual machine downtime.

When this option is used, the upgrade deletes and creating disk groups one at a time, on each host, and then allows the components rebuild once the on-disk format is at v2. When the operation has completed on the first host, it is repeat for the next host and so on until all hosts in the cluster are running on-disk format v2. However administrators need to be aware that their virtual machines could be running unprotected for a period during this upgrade.

vsan.vm_object_info

By using the `vsan.vm_object_info` command, the objects and component layout (RAID-0, RAID-1) of objects can now be examined in detail. It takes one or more virtual machines as an argument. VM Home Namespace and VMDK objects are currently shown by this command. Other objects, such as VM Swap and Snapshot Deltas are not currently shown by this command.

Usage:

```
vsan.vm_object_info {vm} {-c, --cluster} {-p, --perspective-from-host}
                    {-i, --include-detailed-usage} {-h, --help}
```

Examples:

- Display help:

```
> vsan.vm_object_info -h
usage: vm_object_info [opts] vms...
Fetch VSAN object information about a VM
vms: Path to a VirtualMachine
      --cluster, -c <s>: Cluster on which to fetch the object info
--perspective-from-host, -p <s>: Host to query object info from
      --include-detailed-usage, -i: Include detailed usage info
      --help, -h: Show this message
```

- Create a report on the objects and components that make up a virtual machine:
-

```
> vsan.vm_object_info 1
VM ch-vsan-desktop2:
  Namespace directory
    DOM Object: 82e38e54-6383-3870-e701-001f29595f9f (v2, owner: cs-ie-h01.ie.local,
policy: forceProvisioning = 0, hostFailuresToTolerate = 1, spbmProfileId = aa6d5a82-1c88-
45da-85d3-3d74b91a5bad, proportionalCapacity = [0, 100], spbmProfileGenerationNumber = 0,
cacheReservation = 0, stripeWidth = 1)
    RAID_1
      Component: cb87db54-503e-a7f0-1bfe-001b21168828 (state: ACTIVE (5), host: cs-ie-
h01.ie.local, md: naa.600508b1001c388c92e817e43fcd5237, ssd:
naa.600508b1001c61cedd42b0c3fbf55132, usage: 0.4 GB)
      Component: 6889db54-5c5f-5790-a4dd-001b21168828 (state: ACTIVE (5), host: cs-ie-
h02.ie.local, md: naa.600508b1001c07d525259e83da9541bf, ssd:
naa.600508b1001c64b76c8ceb56e816a89d, usage: 0.4 GB)
      Witness: d189db54-c440-c267-0279-001b21168828 (state: ACTIVE (5), host: cs-ie-
h04.ie.local, md: naa.600508b1001c6a664d5d576299cec941, ssd:
naa.600508b1001c29d8145d6cc1925e9fb9, usage: 0.0 GB)
  Disk backing: [vsanDatastore] 82e38e54-6383-3870-e701-001f29595f9f/ch-vsan-desktop2-
000002.vmdk
```

```

DOM Object: 377eae54-cc9a-23f4-03ec-001f29595f9f (v2, owner: cs-ie-h01.ie.local,
policy: spbmProfileGenerationNumber = 0, forceProvisioning = 0, cacheReservation = 0,
hostFailuresToTolerate = 1, stripeWidth = 1, spbmProfileId = aa6d5a82-1c88-45da-85d3-
3d74b91a5bad, proportionalCapacity = [0, 100], objectVersion = 2)
  RAID_1
    Component: 8cfc054-c86e-4602-c0e5-001b21168828 (state: ACTIVE (5), host: cs-ie-
h04.ie.local, md: naa.600508b1001c846c000c3d9114ed71b3, ssd:
naa.600508b1001c29d8145d6cc1925e9fb9,
usage: 10.4 GB)
    Component: 787adb54-9832-f5f3-4e70-001b21168828 (state: ACTIVE (5), host: cs-ie-
h01.ie.local, md: naa.600508b1001c2ee9a6446e708105054b, ssd:
naa.600508b1001c61cedd42b0c3fbf55132,
usage: 10.4 GB)
    Witness: 0b7fdb54-605a-8367-c90f-001b21168828 (state: ACTIVE (5), host: cs-ie-
h03.ie.local, md: naa.600508b1001cb11f3292fe743a0fd2e7, ssd:
naa.600508b1001c9c8b5f6f0d7a2be44433,
usage: 0.0 GB)
>

```

The interesting parts are highlighted in bold. There are two objects visible, the VM Home Namespace and the VMDK (referred to as Disk backing in the above output). Again, the VM Home is using a ***StripeWidth=1***, and the VMDK is also using a ***StripeWidth=1***.

There is lots of useful information displayed here. Another important point is that all components are **ACTIVE**. There are no components in an **STALE**, **ABSENT** or **DEGRADED** state. For more information about component states, please refer to the *Virtual SAN 6.0 Troubleshooting Reference Manual*.

vsan.vm_perf_stats

This command displays IOPS, throughput and latency for a virtual machine for a specified period of time. The command gives a quick, simple, command line based insight into current storage performance of VMs stored on VSAN. The metrics shown are IOPS, Throughput (in KB/s) and Latency (in ms). The command does so by first fetching statistic counters, then waiting a user specified time period (20 seconds by default, use `--interval` to adapt) and collecting counters a second time. The command will then compute the average over the time period.

Usage:

```
vsan.vm_perf_stats {vm} {-i, --interval} {-s, --show-objects} {-h, --help}
```

Examples:

- Display help:

```
> vsan.vm_perf_stats -h
usage: vm_perf_stats [opts] vms...
VM perf stats
vms: Path to a VirtualMachine
--interval, -i <i>:   Time interval to compute average over (default: 20)
--show-objects, -s:  Show objects that are part of VM
--help, -h:         Show this message
```

- Display virtual machine performance statistics. The `--show-objects, -s` option will display each of the different objects that are part of the VM, for example, if it has a number of different VMDK objects associated with it:

```
> vsan.vm_perf_stats ~/vms/W2k12-SQL2k12 --interval 10 --show-objects
output:
2014-10-31 15:19:33 +0000: Got all data, computing table
+-----+-----+-----+-----+
| VM/Object          | IOPS          | Tput (KB/s)  | Latency (ms) |
+-----+-----+-----+-----+
| W2k12-SQL2k12     |               |               |               |
|   /W2k12-SQL2k12.vmx | 0.3r/0.3w    | 0.2r/0.2w    | 0.5r/1.2w    |
|   /W2k12-SQL2k12.vmdk | 1.2r/6.1w    | 7.7r/46.5w   | 0.4r/1.8w    |
|   /W2k12-SQL2k12_1.vmdk | 0.0r/7.7w   | 0.4r/1236.7w | 0.8r/1.8w    |
|   /W2k12-SQL2k12_2.vmdk | 0.4r/647.6w | 1.6r/4603.3w | 1.3r/1.8w    |
+-----+-----+-----+-----+
```

The following calculations may help in understanding the metrics:

- $IOPS = (MBps \text{ Throughput} / KB \text{ per IO}) * 1024$
- $MBps = (IOPS * KB \text{ per IO}) / 1024$

vsan.vmdk_stats

Display read cache and capacity stats for VMs and VMDKs.

Usage:

```
vsan.vmdk_stats {cluster|host} {vm} {-h, --help}
```

Examples:

- Display help:

```
> vsan.vmdk_stats -h
usage: vmdk_stats cluster_or_host vms...
Print read cache and capacity stats for vmdks.
Disk Capacity (GB):
Disk Size: Size of the vmdk
Used Capacity: MD capacity used by this vmdk
Data Size: Size of data on this vmdk
Read Cache (GB):
Used: RC used by this vmdk
Reserved: RC reserved by this vmdk
  cluster_or_host: Path to a ClusterComputeResource or HostSystem
  vms: Path to a VirtualMachine
--help, -h: Show this message
```

Field information:

- **Disk Size (GB):** The size the VM was configured for, i.e. the size the guest OS sees.
- **Used Capacity (GB):** Capacity used on VSAN, taking into account thin provisioning, but also the replication overhead and any temporary overhead during data movements or failure handling. So this number may be lower than Disk Size (due to thin provisioning) or higher (due to replication overhead). Used capacity includes both actual allocated space as well as reserved space (thick provisioning).
- **Data Size (GB):** Same as "used capacity", but only counts actual allocated space, and not reserved space.
- **Read Cache Used (GB):** The portion of the Read Cache that this VMDK is currently using. This may be due to a reservation or due to its fair share use of the Read Cache.
- **Read Cache Reserved:** The amount of Read Cache reserved for this VMDK.

vsan.whatif_host_failures

This is a very useful RVC command for determining if there are enough resources remaining in the cluster to rebuild the missing components in the event of a failure. The HDD capacity reported below refers to the capacity layer, both for all-flash and hybrid. RC reservations refers to read cache reservations, an option that allows an administrator to dedicate a certain amount of read cache to a virtual machine through VM storage policy settings, but it is only relevant to hybrid configurations as there is no read cache reservation setting in all-flash configurations.

There are no 'read cache reservations' in this example. This command once again takes a single argument, which is the cluster.

Usage:

```
vsan.whatif_host_failures {host|cluster} {-n, --num-host-failures-to-simulate}
                        {-s, --show-current-usage-per-host} {-h, --help}
```

Examples:

- Display help:

```
> vsan.whatif_host_failures -h
```

```
usage: whatif_host_failures [opts] hosts_and_clusters...
Simulates how host failures impact VSAN resource usage
```

The command shows current VSAN disk usage, but also simulates how disk usage would evolve under a host failure. Concretely the simulation assumes that all objects would be brought back to full policy compliance by bringing up new mirrors of existing data. The command makes some simplifying assumptions about disk space balance in the cluster. It is mostly intended to do a rough estimate if a host failure would drive the cluster to being close to full.

```
hosts_and_clusters: Path to a HostSystem or ClusterComputeResource
--num-host-failures-to-simulate, -n <i>: Number of host failures to
simulate (default: 1)
--show-current-usage-per-host, -s: Show current resources used per
host
--help, -h: Show this message
```

- Display resources after a single host failure:

```
> vsan.whatif_host_failures 0
Simulating 1 host failures:
```

Resource	Usage right now	Usage after failure/re-protection
HDD capacity	64% used (1190.97 GB free)	90% used (235.88 GB free)
Components	1% used (35647 available)	1% used (26647 available)
RC reservations	0% used (521.57 GB free)	0% used (391.17 GB free)

- Display current resource usage and resources after a single host failure:

```
> vsan.whatif_host_failures 0 -s
Current utilization of hosts:
```

Host	NumHDDs	HDD Capacity		Components		SSD Capacity	
		Total	Used	Reserved	Used	Reserved	
cs-ie-h04	6	818.65 GB	64 %	59 %	86/9000 (1 %)	0 %	
cs-ie-h01	7	955.09 GB	65 %	65 %	111/9000 (1 %)	0 %	
cs-ie-h03	6	818.65 GB	64 %	59 %	81/9000 (1 %)	0 %	
cs-ie-h02	5	682.21 GB	61 %	47 %	75/9000 (1 %)	0 %	

```
Simulating 1 host failures:
```

Resource	Usage right now	Usage after failure/re-protection
HDD capacity	64% used (1190.97 GB free)	90% used (235.88 GB free)
Components	1% used (35647 available)	1% used (26647 available)
RC reservations	0% used (521.57 GB free)	0% used (391.17 GB free)

Reference

Ruby vSphere Console Help Output

- [RVC CLI Help Output](#)
- [RVC v1.8.0 Command List](#)

VMware Blogs

RVC series: blogs.vmware.com/vsphere/2014/07/managing-vsan-ruby-vsphere-console.html

VSAN blog: blogs.vmware.com/vsphere/2014/07/official-vmware-virtual-vsan-blog-index.html

Documentation

[Virtual SAN 6.0 Troubleshooting Reference Manual](#)



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Copyright © 2012 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdiction. All other marks and names mentioned herein may be trademarks of their respective companies.