
ESXi 4.1

vim-cmd

examples

*Entries marked with an * are unfortunately just empty placeholders*

hostsvc/	1
hostsvc/advopt/	1
hostsvc/advopt/options	1
hostsvc/advopt/settings.....	2
hostsvc/advopt/update	2
hostsvc/advopt/view	3
hostsvc/enable_local_tsm.....	3
hostsvc/queryconnectioninfo	4
hostsvc/autostartmanager/	5
hostsvc/autostartmanager/autostart	5
hostsvc/autostartmanager/get_autostartseq.....	5
hostsvc/autostartmanager/update_defaults.....	6
hostsvc/autostartmanager/autostop.....	6
hostsvc/autostartmanager/get_defaults	6
hostsvc/autostartmanager/enable_autostart	6
hostsvc/autostartmanager/update_autostartentry	7
* hostsvc/enable_remote_tsm	8
hostsvc/querydisabledmethods.....	9
hostsvc/datastore/	9
hostsvc/datastore/capabilities	9
hostsvc/datastore/remove.....	10
hostsvc/datastore/destroy.....	10
hostsvc/datastore/rename.....	10
hostsvc/datastore/info.....	11
hostsvc/datastore/summary	12
hostsvc/datastore/listsummary	12
* hostsvc/datastore/vmfs_create	14
hostsvc/datastore/listvm	15
* hostsvc/datastore/vmfs_extend	15
* hostsvc/datastore/localds_create.....	15
hostsvc/datastore/vmfs_queryAvailableDisks	16
hostsvc/datastore/nas_create	17
* hostsvc/datastore/vmfs_query_create_options.....	17

* hostsvc/datastore/refresh.....	17
* hostsvc/datastore/vmfs_query_extend_options.....	17
* hostsvc/firewall_disable_ruleset	17
* hostsvc/refresh_firewall	17
* hostsvc/datastorebrowser/.....	17
* hostsvc/datastorebrowser/deleteFile.....	17
* hostsvc/datastorebrowser/list	17
* hostsvc/datastorebrowser/searchsubfolders	17
* hostsvc/datastorebrowser/disksearch.....	17
* hostsvc/datastorebrowser/search	17
* hostsvc/firewall_enable_ruleset.....	17
* hostsvc/refresh_services.....	17
hostsvc/firmware/.....	17
hostsvc/firmware/backup_config	18
* hostsvc/firmware/reset_config.....	18
* hostsvc/firmware/restore_config	18
* hostsvc/firmware/sync_config	18
hostsvc/hostconfig	19
hostsvc/runtimeinfo.....	20
hostsvc/net/.....	21
* hostsvc/net/mockup/	21
* hostsvc/net/portgroup_remove.....	21
hostsvc/net/provider/	21
hostsvc/net/portgroup_set.....	21
hostsvc/net/capabilities	28
hostsvc/net/portgroup_unsetpolicy	28
hostsvc/net/config	29
hostsvc/net/query_networkhint	30
hostsvc/net/config_append	34
hostsvc/net/refresh.....	35
* hostsvc/net/config_incremental_update.....	35
hostsvc/net/route_info	35
hostsvc/net/config_load	35
* hostsvc/net/route_set.....	35

hostsvc/net/config_save	36
hostsvc/net/set_v6stack	37
* hostsvc/net/console_route_set	37
* hostsvc/net/toggle_raw	37
* hostsvc/net/consolevnic_add	38
* hostsvc/net/vnic_add	38
hostsvc/net/consolevnic_info	39
hostsvc/net/vnic_info	39
* hostsvc/net/consolevnic_remove	41
hostsvc/net/vnic_remove	41
* hostsvc/net/consolevnic_set.....	41
* hostsvc/net/vnic_set	41
* hostsvc/net/consolevnic_setv6.....	42
hostsvc/net/vnic_setv6	42
hostsvc/net/dns_info	42
hostsvc/net/vswitch_add	43
* hostsvc/net/dns_set.....	43
hostsvc/net/vswitch_info.....	43
* hostsvc/net/dns_unset.....	44
hostsvc/net/vswitch_pnicinfo	44
* hostsvc/net/dvs_info.....	46
hostsvc/net/vswitch_remove.....	47
hostsvc/net/info	47
* hostsvc/net/vswitch_setautobridge.....	48
* hostsvc/net/offload_capabilities.....	48
hostsvc/net/ vswitch_setbondbridge.....	48
hostsvc/net/pnic_info	48
hostsvc/net/vswitch_setnumports	50
* hostsvc/net/pnic_setip.....	50
hostsvc/net/vswitch_setpolicy.....	50
hostsvc/net/pnic_setlinkspeed	50
* hostsvc/net/vswitch_setsimplebridge	51
hostsvc/net/portgroup_add.....	51
hostsvc/net/vswitch_unsetbridge.....	51

hostsvc/hosthardware	51
* hostsvc/set_hostid	52
hostsvc/rsrc/.....	53
hostsvc/rsrc/create	53
hostsvc/rsrc/pool_config_get	55
hostsvc/rsrc/pools_get.....	56
* hostsvc/rsrc/vm_config_set	56
hostsvc/rsrc/destroy	57
hostsvc/rsrc/pool_config_set.....	57
hostsvc/rsrc/rename	59
hostsvc/rsrc/vms_get.....	59
hostsvc/rsrc/destroy_children	59
hostsvc/rsrc/pool_runtime_get	60
* hostsvc/rsrc/vm_config_get.....	60
hostsvc/hostsummary.....	60
hostsvc/standby_mode_enter	61
hostsvc/storage/	61
hostsvc/storage/diagnostic/.....	61
* hostsvc/storage/multipath_lun_setpolicy	62
hostsvc/storage/mockup/	63
* hostsvc/storage/multipath_path_disable	63
hostsvc/storage/provider/	63
* hostsvc/storage/multipath_path_enable	64
* hostsvc/storage/datastore_browser_search	64
hostsvc/storage/partition_get	64
hostsvc/storage/fs_info	66
* hostsvc/storage/partition_layout_add	69
hostsvc/storage/hba_info	70
hostsvc/storage/partition_layout_get.....	71
hostsvc/storage/hba_rescan.....	71
* hostsvc/storage/partition_layout_set.....	71
hostsvc/storage/info	72
* hostsvc/storage/partition_remove	72
hostsvc/storage/iscsi_add_send_target	73

* hostsvc/storage/partition_spec_add	73
* hostsvc/storage/iscsi_add_static_target	73
hostsvc/storage/partition_spec_get	73
hostsvc/storage/iscsi_disable_chap.....	75
hostsvc/storage/refresh	76
* hostsvc/storage/iscsi_discovery.....	76
hostsvc/storage/scsilun_info	76
hostsvc/storage/iscsi_enable_chap	77
hostsvc/storage/software_iscsi_enabled.....	78
* hostsvc/storage/iscsi_ip_config	79
hostsvc/storage/topology_info	79
* hostsvc/storage/iscsi_ip_use_dhcp	80
hostsvc/storage/upgrade_vm_layout	80
hostsvc/storage/iscsi_remove_send_target.....	80
* hostsvc/storage/vmfs_attach_extent	81
hostsvc/storage/iscsi_remove_static_target.....	81
* hostsvc/storage/vmfs_format.....	81
hostsvc/storage/iscsi_set_alias.....	82
hostsvc/storage/vmfs_rescan	82
* hostsvc/storage/iscsi_set_name	85
hostsvc/storage/vmfs_upgrade	85
hostsvc/storage/multipath_info	86
* hostsvc/login	93
* hostsvc/standby_mode_exit.....	93
hostsvc/summary/	93
hostsvc/summary/fsvolume.....	93
hostsvc/summary/hba	93
hostsvc/summary/scsilun.....	94
* hostsvc/logout.....	94
hostsvc/start_local_tsm	94
hostsvc/vmotion/	95
* hostsvc/vmotion/ipconfig_get	95
* hostsvc/vmotion/ipconfig_set.....	95
hostsvc/vmotion/netconfig_get.....	95

hostsvc/vmotion/vnic_set.....	97
hostsvc/vmotion/vnic_unset.....	97
hostsvc/maintenance_mode_enter.....	98
* hostsvc/start_remote_tsm.....	98
* hostsvc/connect	98
hostsvc/maintenance_mode_exit.....	98
* hostsvc/stop_local_tsm.....	98
hostsvc/cpuinfo.....	99
* hostsvc/memoryinfo	99
* hostsvc/stop_remote_tsm	99
* hostsvc/disable_local_tsm	99
* hostsvc/pci_add	99
hostsvc/task_list.....	99
* hostsvc/disable_remote_tsm.....	100
* hostsvc/pci_remove	100
proxysvc/	101
* proxysvc/add_np_service.....	101
* proxysvc/add_tcp_service login.....	101
* proxysvc/connect	101
* proxysvc/logout.....	101
proxysvc/port_info.....	101
* proxysvc/remove_service	101
proxysvc/service_list.....	101
supportsvc_cmds/	103
* supportsvc_cmds/connect	103
* supportsvc_cmds/login	103
* supportsvc_cmds/logout.....	103
* supportsvc_cmds/printstatstofile	103
vmsvc/	104
* vmsvc/acquiremksticket.....	104
* vmsvc/acquireticket	104
* vmsvc/connect	104
* vmsvc/convert.toTemplate	104

* vmsvc/convert.toVm	104
* vmsvc/createdummyvm.....	104
* vmsvc/destroy	104
* vmsvc/device.connection.....	104
* vmsvc/device.connusbdev	104
* vmsvc/device.disconnusbdev.....	104
* vmsvc/device.diskadd	104
* vmsvc/device.diskaddexisting.....	104
* vmsvc/device.diskremove	104
vmsvc/device.getdevices	104
* vmsvc/device.toolsSyncSet	110
* vmsvc/device.vmiadd.....	110
* vmsvc/device.vmiremove	110
* vmsvc/devices.createnic	110
vmsvc/get.capability	111
vmsvc/get.config	111
vmsvc/get.config.cpuidmask.....	119
vmsvc/get.configoption	121
vmsvc/get.datastores.....	135
vmsvc/get.disabledmethods	136
vmsvc/get.environment	137
vmsvc/get.filelayout.....	139
vmsvc/get.guest	140
vmsvc/get.guestheartbeatStatus.....	141
vmsvc/get.managedentitystatus.....	141
vmsvc/get.networks.....	142
vmsvc/get.runtime	142
vmsvc/get.snapshotinfo	142
vmsvc/get.summary.....	143
* vmsvc/get.tasklist.....	145
* vmsvc/gethostconstraints	145
* vmsvc/login	145
* vmsvc/logout.....	145
* vmsvc/message	145

vmsvc/power.getstate	145
vmsvc/power.hibernate	145
vmsvc/power.off	145
vmsvc/power.reboot.....	145
vmsvc/power.reset	146
vmsvc/power.shutdown.....	146
vmsvc/power.suspend	146
* vmsvc/power.suspendResume	146
* vmsvc/queryftcompat	146
* vmsvc/reload	146
* vmsvc/setscreenres.....	146
vmsvc/snapshot.create	147
* vmsvc/snapshot.dumpoption	147
vmsvc/snapshot.get	147
* vmsvc/snapshot.remove	147
vmsvc/snapshot.removeall	147
vmsvc/snapshot.revert.....	148
* vmsvc/snapshot.setoption	149
vmsvc/tools.cancelinstall	149
vmsvc/tools.install.....	149
* vmsvc/tools.upgrade	149
* vmsvc/unregister.....	149
* vmsvc/upgrade	149
vmsvc/getallvms.....	150
vmsvc/power.on.....	150

internalsvc/ 151

internalsvc/perfcount/	151
* internalsvc/perfcount/enumerate	151
* internalsvc/perfcount/query_execute	151
* internalsvc/perfcount/query_list	151
* internalsvc/perfcount/query_regex.....	151
* internalsvc/perfcount/query_destroy.....	151
* internalsvc/perfcount /query_info.....	151
* internalsvc/perfcount/query_names	151

internalsvc/perfcount /scoreboard	151
* internalsvc/logout	152
* internalsvc/refresh_network	152
* internalsvc/access_address.....	152
* internalsvc/redirect_stderr	152
* internalsvc/refresh_pnic	152
* internalsvc/cold_quit	152
* internalsvc/redirect_stdout	152
* internalsvc/set_exception_variable	152
* internalsvc/connect.....	152
* internalsvc/refresh	152
* internalsvc/set_log_level	152
* internalsvc/host_mode_lock.....	152
* internalsvc/refresh_consolenic.....	152
* internalsvc/shutdown	152
* internalsvc/login.....	152
* internalsvc/refresh_datastores.....	152
* internalsvc/throw_exception	152
internalsvc/loglist.....	153
* internalsvc/refresh_gateway	154
* internalsvc/use_fds	154
solo/	155
* solo/connect.....	155
* solo/login.....	155
* solo/querycfgopt	155
solo/registervm	155
solo/environment.....	155
* solo/logout	156
solo/querycfgoptdesc.....	156
vimsvc/	157
vimsvc/auth/	157
* vimsvc/auth/entity_permission_add	157
* vimsvc/auth/lockdown_mode_enter.....	157

vimsvc/auth/role_permissions.....	157
* vimsvc/auth/entity_permission_remove.....	157
* vimsvc/auth/lockdown_mode_exit.....	157
* vimsvc/auth/role_remove.....	157
* vimsvc/auth/entity_permissions.....	157
vimsvc/auth/permissions.....	157
vimsvc/auth/lockdown_is_enabled.....	164
vimsvc/auth/privileges.....	164
vimsvc/auth/lockdown_is_possible.....	193
vimsvc/auth/role_add.....	193
* vimsvc/login.....	194
* vimsvc/task_cancel.....	194
vimsvc/task_list.....	194
* vimsvc/connect.....	194
* vimsvc/logout.....	194
vimsvc/task_description.....	195
vimsvc/license.....	196
vimsvc/property_dump.....	197
* vimsvc/task_info.....	198

vim-cmd

hostsvc/

hostsvc/advopt/

hostsvc/advopt/options

~ # vim-cmd /hostsvc/advopt/options

```
(vim.option.OptionDef) [
  (vim.option.OptionDef) {
    dynamicType = <unset>,
    label = "Annotations.WelcomeMessage",
    summary = "",
    key = "Annotations.WelcomeMessage",
    optionType = (vim.option.StringOption) {
      dynamicType = <unset>,
      valuesReadOnly = false,
      defaultValue = "",
      validCharacters = <unset>,
    },
  },
  (vim.option.OptionDef) {
    dynamicType = <unset>,
    label = "SoftMaxDirty",
    summary = "Flush immediately if this many buffers are dirty (percent)",
    key = "BufferCache.SoftMaxDirty",
    optionType = (vim.option.LongOption) {
      dynamicType = <unset>,
      valuesReadOnly = false,
      min = 0,
      max = 100,
      defaultValue = 15,
    },
  },
  (vim.option.OptionDef) {
    dynamicType = <unset>,
    label = "HardMaxDirty",
    summary = "Block writers if this many buffers are dirty (percent)",
    key = "BufferCache.HardMaxDirty",
    optionType = (vim.option.LongOption) {
      dynamicType = <unset>,
      valuesReadOnly = false,
      min = 0,
      max = 100,
      defaultValue = 95,
    },
  },
  ...
]
```

hostsvc/advopt/settings

~ # vim-cmd /hostsvc/advopt/settings | less

```
(vim.option.OptionValue) [
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "Annotations.WelcomeMessage",
  value = "",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "BufferCache.SoftMaxDirty",
  value = 15,
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "BufferCache.HardMaxDirty",
  value = 95,
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "BufferCache.FlushInterval",
  value = 30000,
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "COW.COWMaxHeapSizeMB",
  value = 192,
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "Config.Defaults.cpuidMask.mode.0.eax",
  value = "disable",
},
...

```

hostsvc/advopt/update

~ # vim-cmd /hostsvc/advopt/update Annotations.WelcomeMessage string "Welcome"

~ # vim-cmd /hostsvc/advopt/view Annotations.WelcomeMessage

```
(vim.option.OptionValue) [
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "Annotations.WelcomeMessage",
  value = "Welcome",
}

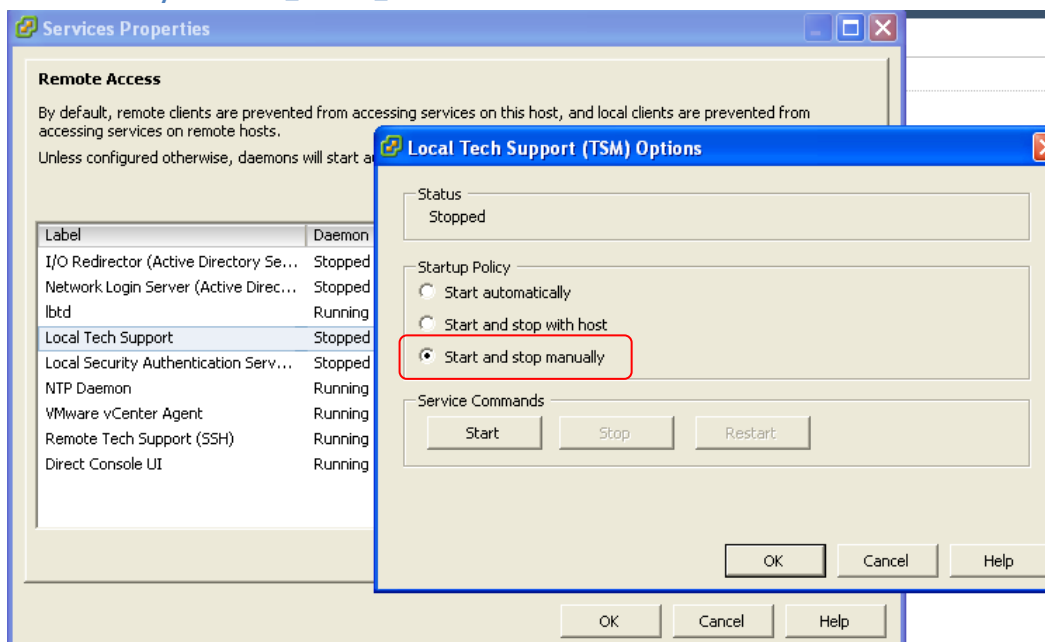
```

hostsvc/advopt/view

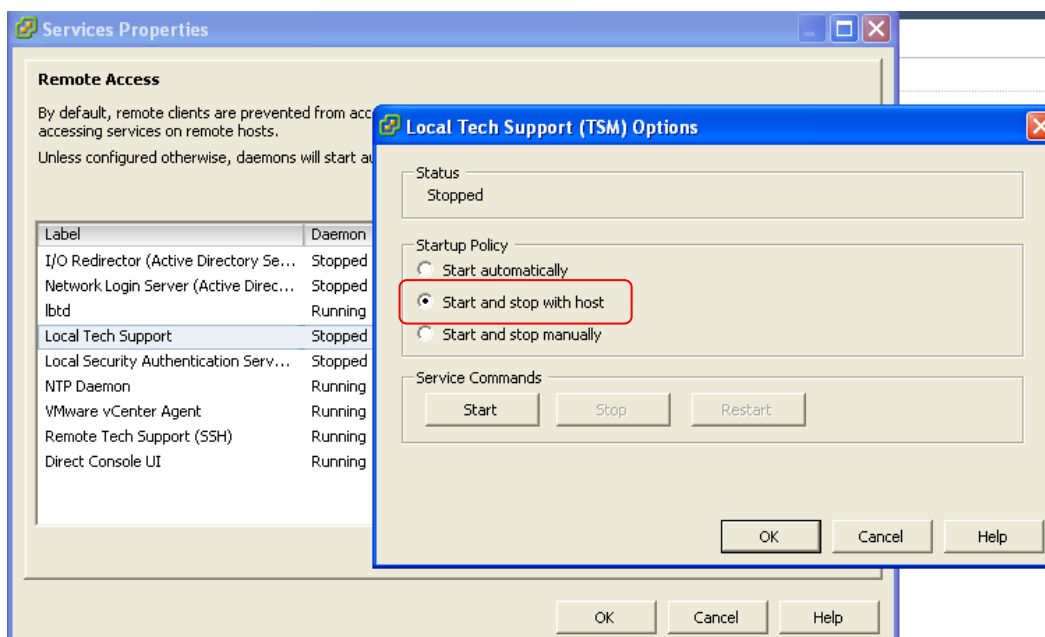
~ # vim-cmd /hostsvc/advopt/view Annotations.WelcomeMessage

```
(vim.option.OptionValue) [
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "Annotations.WelcomeMessage",
  value = "Welcome",
}
```

hostsvc/enable_local_tsm



~ # vim-cmd /hostsvc/enable_local_tsm



hostsvc/queryconnectioninfo










~ # vim-cmd hostsvc/queryconnectioninfo

```
(vim.host.ConnectInfo) {
  dynamicType = <unset>,
  serverIp = "192.168.9.231",
  host = (vim.host.Summary) {
    dynamicType = <unset>,
    host = 'vim.HostSystem:ha-host',
    hardware = (vim.host.Summary.HardwareSummary) {
      dynamicType = <unset>,
      vendor = "VMware, Inc.",
      model = "VMware Virtual Platform",
      uuid = "564d985c-68ac-a9b8-0486-1d8d95829b75",
      otherIdentifyingInfo = (vim.host.SystemIdentificationInfo) [
        (vim.host.SystemIdentificationInfo) {
          dynamicType = <unset>,
          identifierValue = " No Asset Tag",
          identifierType = (vim.ElementDescription) {
            dynamicType = <unset>,
            label = "AssetTag",
            summary = "AssetTag",
            key = "AssetTag",
          },
        },
      ],
    },
    memorySize = 2356490240,
    cpuModel = "Intel(R) Core(TM)2 Duo CPU   T7500 @ 2.20GHz",
    cpuMhz = 2186,
    numCpuPkgs = 1,
    numCpuCores = 1,
    numCpuThreads = 1,
    numNics = 6,
    numHBAs = 4,
  },
  runtime = (vim.host.RuntimeInfo) {
    dynamicType = <unset>,
    connectionState = "connected",
    powerState = "poweredOn",
    standbyMode = <unset>,
    inMaintenanceMode = false,
    bootTime = "2010-08-13T09:20:18.58907Z",
    healthSystemRuntime = (vim.host.HealthStatusSystem.Runtime) {
    ...
```

hostsvc/autostartmanager/

hostsvc/autostartmanager/autostart

~ # vim-cmd hostsvc/autostartmanager/autostart

 Power On virtual machine	 VMS	 Completed
 Auto power On		 In Progress
 Reconfigure AutoStart Manager		 Completed
 Reconfigure AutoStart Manager		 Completed

hostsvc/autostartmanager/get_autostartseq

~ # vim-cmd hostsvc/autostartmanager/get_autostartseq

```
(vim.host.AutoStartManager.AutoPowerInfo) [
  (vim.host.AutoStartManager.AutoPowerInfo) {
    dynamicType = <unset>,
    key = 'vim.VirtualMachine:208',
    startOrder = 1,
    startDelay = 60,
    waitForHeartbeat = "yes",
    startAction = "PowerOn",
    stopDelay = 60,
    stopAction = "Suspend",
  },
  (vim.host.AutoStartManager.AutoPowerInfo) {
    dynamicType = <unset>,
    key = 'vim.VirtualMachine:240',
    startOrder = 2,
    startDelay = 78,
    waitForHeartbeat = "no",
    startAction = "PowerOn",
    stopDelay = 999,
    stopAction = "Suspend",
  },
  (vim.host.AutoStartManager.AutoPowerInfo) {
    dynamicType = <unset>,
    key = 'vim.VirtualMachine:224',
    startOrder = -1,
    startDelay = -1,
    waitForHeartbeat = "systemDefault",
    startAction = "PowerOn",
    stopDelay = -1,
    stopAction = "SystemDefault",
  }
]
```

hostsvc/autostartmanager/update_defaults

```
~ # vim-cmd hostsvc/autostartmanager/update_defaults
```

Insufficient arguments.

Usage: update_defaults StartDelay [StopDelay] [StopAction] [WaitForHeartbeat]

```
~ # vim-cmd hostsvc/autostartmanager/update_defaults 240 240 "Suspend" true
```

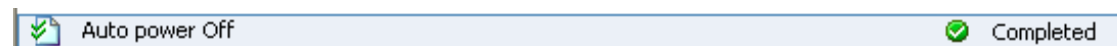
Updated System defaults.

```
~ # vim-cmd hostsvc/autostartmanager/get_defaults
```

```
(vim.host.AutoStartManager.SystemDefaults) {
  dynamicType = <unset>,
  enabled = false,
  startDelay = 240,
  stopDelay = 240,
  waitForHeartbeat = true,
  stopAction = "Suspend",
}
```

hostsvc/autostartmanager/autostop

```
~ # vim-cmd hostsvc/autostartmanager/autostop
```



hostsvc/autostartmanager/get_defaults

```
~ # vim-cmd hostsvc/autostartmanager/get_defaults
```

```
(vim.host.AutoStartManager.SystemDefaults) {
  dynamicType = <unset>,
  enabled = false,
  startDelay = 120,
  stopDelay = 120,
  waitForHeartbeat = false,
  stopAction = "PowerOff",
}
```

hostsvc/autostartmanager/enable_autostart

```
~ # vim-cmd hostsvc/autostartmanager/enable_autostart true
```

Enabled AutoStart

```
~ # vim-cmd hostsvc/autostartmanager/get_defaults
```

```
(vim.host.AutoStartManager.SystemDefaults) {
  dynamicType = <unset>,
  enabled = true,
  startDelay = 240,
  stopDelay = 240,
  waitForHeartbeat = true,
  stopAction = "Suspend",
}
```

hostsvc/autostartmanager/update_autostartentry

```
~ # vim-cmd hostsvc/autostartmanager/update_autostartentry
```

Insufficient arguments.

Usage: update_autostartentry VMId StartAction StartDelay StartOrder StopAction StopDelay WaitForHeartbeat

Updates a single VM's AutoStart entry.

```
~ # vim-cmd vmsvc/getallvms
```

Vmid	Name	File	Guest OS	Version	Annotation
176	ubuntuServer8.04-2	[NFS] ubuntuServer8.04-2/ubunt	ubuntuServer8.04-2.vmx	ubuntuGuest	vmx-07 ubuntu Server 8.04 LTS x86
192	VM5	[NFS] VM5/VM5.vmx	winXPProGuest	vmx-07	
208	VM1	[NFS] VM1/VM1.vmx	winXPProGuest	vmx-07	
224	ubuntuServer8.04-1	[NFS] ubuntuServer8.04-1/ubunt	ubuntuServer8.04-1.vmx	ubuntuGuest	vmx-07 ubuntu Server 8.04 LTS x86
240	New Virtual Machine	[of-3] New Virtual Machine/New	Virtual Machine.vmx	windows7Server64Guest	vmx-07

```
~ # vim-cmd hostsvc/autostartmanager/update_autostartentry 208 stop 90 1 stop 90 no
```

1Updated AutoStart order.

```
~ # vim-cmd hostsvc/autostartmanager/update_autostartentry 192 poweron 90 1 stop 90 no
```

1Updated AutoStart order.

```
~ # vim-cmd hostsvc/autostartmanager/update_autostartentry 192 PowerOn 90 2 stop 90 no
```

2Updated AutoStart order.

```
~ # vim-cmd hostsvc/autostartmanager/update_autostartentry 240 PowerOn 90 3 Suspend 90 Yes
```

3Invalid argument

```
~ # vim-cmd hostsvc/autostartmanager/update_autostartentry 240 PowerOn 90 3 Suspend 90 yes
```

3Updated AutoStart order.

```
~ # vim-cmd hostsvc/autostartmanager/get_autostartseq
```

```
(vim.host.AutoStartManager.AutoPowerInfo) [
  (vim.host.AutoStartManager.AutoPowerInfo) {
    dynamicType = <unset>,
    key = 'vim.VirtualMachine:208',
    startOrder = 2,
    startDelay = 90,
    waitForHeartbeat = "no",
    startAction = "stop",
    stopDelay = 90,
    stopAction = "stop",
  },
  (vim.host.AutoStartManager.AutoPowerInfo) {
    dynamicType = <unset>,
    key = 'vim.VirtualMachine:192',
    startOrder = 2,
    startDelay = 90,
    waitForHeartbeat = "no",
    startAction = "PowerOn",
    stopDelay = 90,
    stopAction = "stop",
  },
]
```

```
(vim.host.AutoStartManager.AutoPowerInfo) {
  dynamicType = <unset>,
  key = 'vim.VirtualMachine:240',
  startOrder = 3,
  startDelay = 90,
  waitForHeartbeat = "yes",
  startAction = "PowerOn",
  stopDelay = 90,
  stopAction = "Suspend",
}
]
```

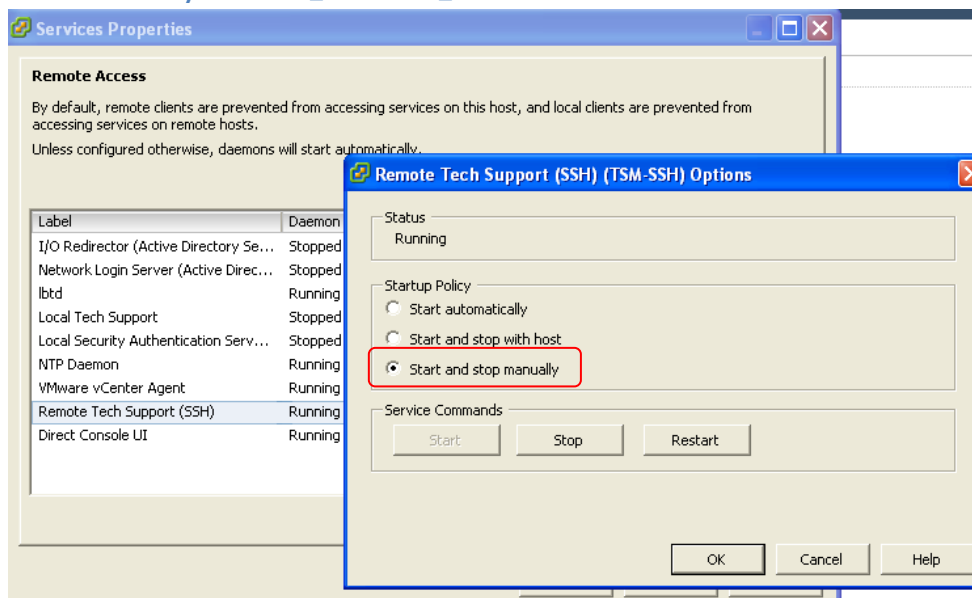
Virtual Machine Startup and Shutdown

Start and Stop Virtual Machines with the system	Enabled
Default Startup Delay	4 minutes
Default Shutdown Delay	4 minutes

Startup Order

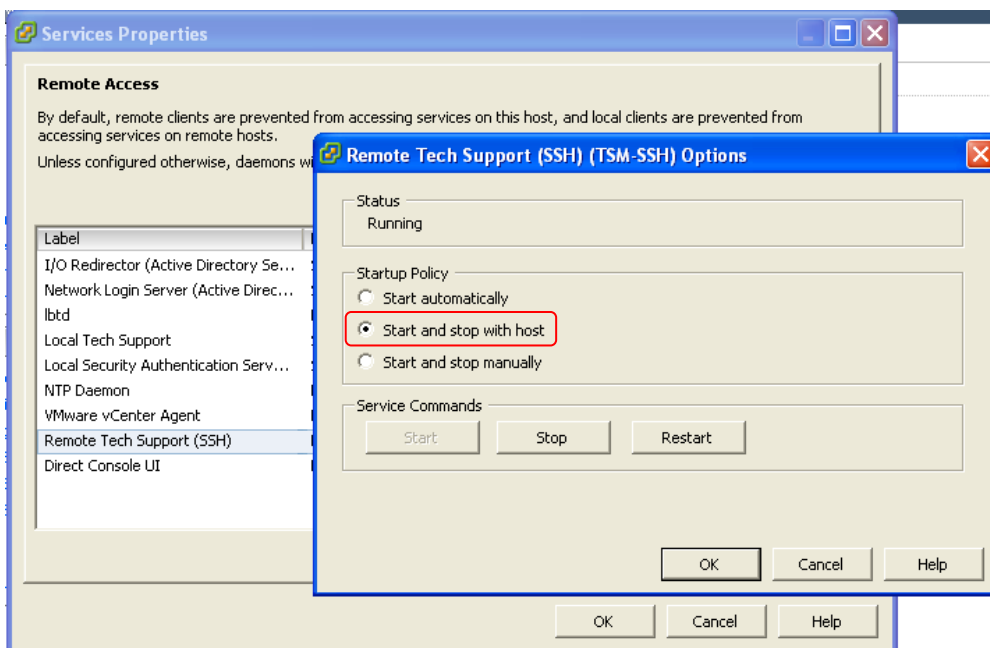
Order	Virtual Machine	Startup	Startup Delay	Shutdown	Shutdown Delay
Automatic Startup					
2	VM5	Enabled	90 seconds	stop	90 seconds
2	VM1	Enabled	90 seconds	stop	90 seconds
3	New Virtual Machi...	Enabled	90 seconds	Suspend	90 seconds
Manual Startup					
	ubuntuServer8.04...	Disabled	240 seconds	Suspend	240 seconds
	ubuntuServer8.04...	Disabled	240 seconds	Suspend	240 seconds

* hostsvc/enable_remote_tsm



```
~ # vim-cmd hostsvc/enable_remote_tsm
```

```
~ #
```



hostsvc/querydisabledmethods

~ # vim-cmd hostsvc/querydisabledmethods

Disabled methods:

```
(vmodl.MethodName) [
  "vim.HostSystem.disconnect",
  "vim.HostSystem.reconnect",
  "vim.HostSystem.reconfigureDAS",
  "vim.HostSystem.exitStandbyMode",
  "vim.HostSystem.exitMaintenanceMode"
```

]

hostsvc/datastore/

hostsvc/datastore/capabilities

~ # vim-cmd hostsvc/datastore/capabilities

```
(vim.host.DatastoreSystem.Capabilities) {
  dynamicType = <unset>,
  nfsMountCreationRequired = true,
  nfsMountCreationSupported = true,
  localDatastoreSupported = false,
  vmfsExtentExpansionSupported = true,
}
```

hostsvc/datastore/remove

~ # vim-cmd hostsvc/datastore/remove of-2

~ #

hostsvc/datastore/destroy

~ # vim-cmd hostsvc/datastore/destroy of-2

~ #

hostsvc/datastore/rename

~ # vim-cmd hostsvc/datastore/rename

Insufficient arguments.

Usage: rename name newname

Rename a datastore.

~ # vim-cmd hostsvc/datastore/summary OF-1

```
(vim.Datastore.Summary) {
  dynamicType = <unset>,
  datastore = 'vim.Datastore:4c654f65-75b87896-ef8e-000c29829b75',
  name = "OF-1",
  url = "/vmfs/volumes/4c654f65-75b87896-ef8e-000c29829b75",
  capacity = 3489660928,
  freeSpace = 3195011072,
  uncommitted = 0,
  accessible = true,
  multipleHostAccess = <unset>,
  type = "VMFS",
}
```

~ # vim-cmd hostsvc/datastore/rename OF-1 of-1

~ # vim-cmd hostsvc/datastore/rename OF-1 of-1

Datastore not found.

~ # vim-cmd hostsvc/datastore/summary of-1

```
(vim.Datastore.Summary) {
  dynamicType = <unset>,
  datastore = 'vim.Datastore:4c654f65-75b87896-ef8e-000c29829b75',
  name = "of-1",
  url = "/vmfs/volumes/4c654f65-75b87896-ef8e-000c29829b75",
  capacity = 3489660928,
  freeSpace = 3195011072,
  uncommitted = 0,
  accessible = true,
  multipleHostAccess = <unset>,
  type = "VMFS",
}
```

hostsvc/datastore/info

~ # vim-cmd hostsvc/datastore/info of-2

```
(vim.host.VmfsDatastoreInfo) {
  dynamicType = <unset>,
  name = "of-2",
  url = "/vmfs/volumes/4c6557df-f4157457-9ac2-000c29829b75",
  freeSpace = 1047527424,
  maxFileSize = 274877906944,
  timestamp = "2010-08-13T15:44:21.890433Z",
  vmfs = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "of-2",
    capacity = 1342177280,
    blockSizeMb = 1,
    maxBlocks = 262144,
    majorVersion = 3,
    version = "3.46",
    uuid = "4c6557df-f4157457-9ac2-000c29829b75",
    extent = (vim.host.ScsiDisk.Partition) [
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
        partition = 1,
      }
    ],
    vmfsUpgradable = false,
    taggedForVmfsUpgrade = <unset>,
    forceMountedInfo = (vim.host.ForceMountedInfo) null,
  },
}
(vim.Datastore.HostMount) [
  (vim.Datastore.HostMount) {
    dynamicType = <unset>,
    key = 'vim.HostSystem:ha-host',
    mountInfo = (vim.host.MountInfo) {
      dynamicType = <unset>,
      path = "/vmfs/volumes/4c6557df-f4157457-9ac2-000c29829b75",
      accessMode = "readWrite",
      accessible = true,
    },
  }
]
```

hostsvc/datastore/summary

~ # vim-cmd hostsvc/datastore/summary of-2

```
(vim.Datastore.Summary) {
  dynamicType = <unset>,
  datastore = 'vim.Datastore:4c6557df-f4157457-9ac2-000c29829b75',
  name = "of-2",
  url = "/vmfs/volumes/4c6557df-f4157457-9ac2-000c29829b75",
  capacity = 1342177280,
  freeSpace = 1047527424,
  uncommitted = 0,
  accessible = true,
  multipleHostAccess = <unset>,
  type = "VMFS",
}
```

hostsvc/datastore/listssummary

~ # vim-cmd hostsvc/datastore/listssummary

```
(vim.Datastore.Summary) [
  (vim.Datastore.Summary) {
    dynamicType = <unset>,
    datastore = 'vim.Datastore:4c5cc8e0-80fe1968-98ec-000c29829b7f',
    name = "datastore1",
    url = "/vmfs/volumes/4c5cc8e0-80fe1968-98ec-000c29829b7f",
    capacity = 4294967296,
    freeSpace = 3967811584,
    uncommitted = 0,
    accessible = true,
    multipleHostAccess = <unset>,
    type = "VMFS",
  },
  (vim.Datastore.Summary) {
    dynamicType = <unset>,
    datastore = 'vim.Datastore:4c62c8c2-ecb7f710-88ac-000c29829b75',
    name = "FUNB",
    url = "/vmfs/volumes/4c62c8c2-ecb7f710-88ac-000c29829b75",
    capacity = 4026531840,
    freeSpace = 3710910464,
    uncommitted = 0,
    accessible = true,
    multipleHostAccess = <unset>,
    type = "VMFS",
  },
  (vim.Datastore.Summary) {
    dynamicType = <unset>,
    datastore = 'vim.Datastore:192.168.9.222:/media/dwl/public/ISO_Library',
    name = "ISO",
    url = "/vmfs/volumes/be455b13-e1a6c8e3",
    capacity = 250056011776,
    freeSpace = 60769251328,
    uncommitted = 0,
    accessible = true,
  }
]
```

```

multipleHostAccess = <unset>,
type = "NFS",
},
(vim.Datastore.Summary) {
dynamicType = <unset>,
datastore = 'vim.Datastore:192.168.9.222:/media/dwl/public/NFS',
name = "NFS",
url = "/vmfs/volumes/1c445c7d-fd56af7a",
capacity = 250056011776,
freeSpace = 60769251328,
uncommitted = 32886079488,
accessible = true,
multipleHostAccess = <unset>,
type = "NFS",
},
(vim.Datastore.Summary) {
dynamicType = <unset>,
datastore = 'vim.Datastore:4c654f65-75b87896-ef8e-000c29829b75',
name = "OF-1",
url = "/vmfs/volumes/4c654f65-75b87896-ef8e-000c29829b75",
capacity = 3489660928,
freeSpace = 3195011072,
uncommitted = 0,
accessible = true,
multipleHostAccess = <unset>,
type = "VMFS",
},
(vim.Datastore.Summary) {
dynamicType = <unset>,
datastore = 'vim.Datastore:4c6557df-f4157457-9ac2-000c29829b75',
name = "of-2",
url = "/vmfs/volumes/4c6557df-f4157457-9ac2-000c29829b75",
capacity = 1342177280,
freeSpace = 1047527424,
uncommitted = 0,
accessible = true,
multipleHostAccess = <unset>,
type = "VMFS",
},
(vim.Datastore.Summary) {
dynamicType = <unset>,
datastore = 'vim.Datastore:4c65661f-1a2e18b5-783e-000c29829b75',
name = "snap-0d853437-of-2",
url = "/vmfs/volumes/4c65661f-1a2e18b5-783e-000c29829b75",
capacity = 1342177280,
freeSpace = 1047527424,
uncommitted = 0,
accessible = true,
multipleHostAccess = <unset>,
type = "VMFS",
},
(vim.Datastore.Summary) {
dynamicType = <unset>,
datastore = 'vim.Datastore:4c656728-065ce6bc-aea3-000c29829b75',

```

```

name = "of-3",
url = "/vmfs/volumes/4c656728-065ce6bc-aea3-000c29829b75",
capacity = 1342177280,
freeSpace = 942669824,
uncommitted = 0,
accessible = true,
multipleHostAccess = <unset>,
type = "VMFS",
}
]

```

*** `hostsvc/datastore/vmfs_create`**

`--partition=int`

Partition number.

`--startsector=int`

Start sector.

`--endsector=int`

End sector.

`--typeid=fs_type`

Type of data in the file system.

`--logical=bool`

If it is logical partition.

`--attributes=byte`

Partition attributes.

`--partition2=int`

Partition number.

`--startsector2=int`

Start sector.

`--endsector2=int`

End sector.

`--typeid2=byte`

Type of data in the file system.

`--logical2=bool`

If it is logical partition.

--attributes2=byte

Partition attributes.

--partition3=int

Partition number.

--startsector3=int

Start sector.

--endsector3=int

End sector.

--typeid3=byte

Type of data in the file system.

--logical3=bool

If it is logical partition.

--attributes3=byte

Partition attributes.

--extentIds=partition_list

List of partitions to extent the VMFS.

hostsvc/datastore/listvm

~ # vim-cmd hostsvc/datastore/listvm of-3

(ManagedObjectReference) [

'vim.VirtualMachine:240'

]

* **hostsvc/datastore/vmfs_extend**

* **hostsvc/datastore/localds_create**

hostsvc/datastore/vmfs_queryAvailableDisks

~ # vim-cmd hostsvc/datastore/vmfs_queryAvailableDisk

```
(vim.host.ScsiDisk) [
(vim.host.ScsiDisk) {
  dynamicType = <unset>,
  deviceName = "/vmfs/devices/disks/t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
  deviceType = "disk",
  key = "key-vim.host.ScsiDisk-010002000034626f4c31372d774536692d4d52706d564952545541",
  uuid = "010002000034626f4c31372d774536692d4d52706d564952545541",
  descriptor = (vim.host.ScsiLun.Descriptor) [
    (vim.host.ScsiLun.Descriptor) {
      dynamicType = <unset>,
      quality = "highQuality",
      id = "t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
    },
    (vim.host.ScsiLun.Descriptor) {
      dynamicType = <unset>,
      quality = "mediumQuality",
      id = "vml.010002000034626f4c31372d774536692d4d52706d564952545541",
    },
    (vim.host.ScsiLun.Descriptor) {
      dynamicType = <unset>,
      quality = "mediumQuality",
      id = "010002000034626f4c31372d774536692d4d52706d564952545541",
    }
  ],
  canonicalName = "t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
  displayName = "OPNFILER iSCSI Disk (t10.F405E46494C454004326F6C41373D277546396D2D42507D6)",
  lunType = "disk",
  vendor = "OPNFILER",
  model = "VIRTUAL-DISK ",
  revision = "0 ",
  scsiLevel = 4,
  serialNumber = "unavailable",
  durableName = (vim.host.ScsiLun.DurableName) {
    dynamicType = <unset>,
    namespace = "SERIALNUM",
    namespaceId = 4,
    data = (byte) [
      52,
      98,
      111,
      76,
      49,
      55,
      ...
    ]
  }
}
```

hostsvc/datastore/nas_create

~ # vim-cmd hostsvc/datastore/nas_create

Insufficient arguments.

Usage: nas_create name remoteHost remotePath readonly

Add a NAS datastore.

readonly is a boolean value, 1 for readonly and 0 for rw access.

~ # vim-cmd hostsvc/datastore/nas_create testnfs 192.168.9.222 /media/dwl/public/TESTNFS 0

~ #

* **hostsvc/datastore/vmfs_query_create_options**

* **hostsvc/datastore/refresh**

* **hostsvc/datastore/vmfs_query_extend_options**

* **hostsvc/firewall_disable_ruleset**

* **hostsvc/refresh_firewall**

* **hostsvc/datastorebrowser/**

* **hostsvc/datastorebrowser/deleteFile**

* **hostsvc/datastorebrowser/list**

* **hostsvc/datastorebrowser/searchsubfolders**

* **hostsvc/datastorebrowser/disksearch**

* **hostsvc/datastorebrowser/search**

* **hostsvc/firewall_enable_ruleset**

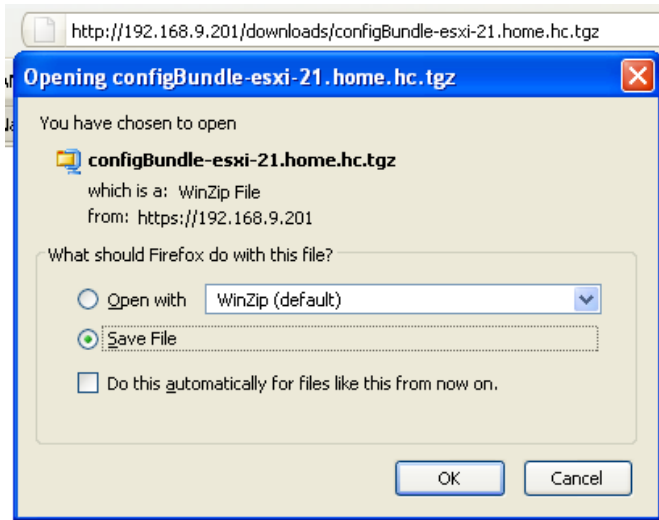
* **hostsvc/refresh_services**

hostsvc/firmware/

hostsvc/firmware/backup_config

~ # vim-cmd /hostsvc/firmware/backup_config

Bundle can be downloaded at : http://*/downloads/configBundle-esxi-21.home.hc.tgz



* **hostsvc/firmware/reset_config**

* **hostsvc/firmware/restore_config**

* **hostsvc/firmware/sync_config**

hostsvc/hostconfig

~ # vim-cmd /hostsvc/hostconfig

```
(vim.host.ConfigInfo) {
  dynamicType = <unset>,
  host = 'vim.HostSystem:ha-host',
  product = (vim.AboutInfo) {
    dynamicType = <unset>,
    name = "VMware ESXi",
    fullName = "VMware ESXi 4.1.0 build-260247",
    vendor = "VMware, Inc.",
    version = "4.1.0",
    build = "260247",
    localeVersion = "INTL",
    localeBuild = "000",
    osType = "vmnix-x86",
    productLineId = "embeddedEsx",
    apiType = "HostAgent",
    apiVersion = "4.1",
    instanceUuid = <unset>,
    licenseProductName = "VMware ESX Server",
    licenseProductVersion = "4.0",
  },
  hyperThread = (vim.host.CpuSchedulerSystem.HyperThreadScheduleInfo) {
    dynamicType = <unset>,
    available = false,
    active = false,
    config = true,
  },
  consoleReservation = (vim.host.MemoryManagerSystem.ServiceConsoleReservationInfo) null,
  virtualMachineReservation = (vim.host.MemoryManagerSystem.VirtualMachineReservationInfo) null,
  storageDevice = (vim.host.StorageDeviceInfo) {
    dynamicType = <unset>,
    hostBusAdapter = (vim.host.HostBusAdapter) [
      (vim.host.BlockHba) {
        dynamicType = <unset>,
        key = "key-vim.host.BlockHba-vmhba0",
        device = "vmhba0",
        bus = 0,
        status = "unknown",
        model = "Virtual Machine Chipset",
        driver = "ata_piix",
        pci = "00:07.1",
      },
      (vim.host.ParallelScsiHba) {
        dynamicType = <unset>,
        key = "key-vim.host.ParallelScsiHba-vmhba1",
        device = "vmhba1",
      },
    ],
  },
  ...
}
```

hostsvc/runtimeinfo

~ # vim-cmd /hostsvc/runtimeinfo

```
(vim.host.RuntimeInfo) {
  dynamicType = <unset>,
  connectionState = "connected",
  powerState = "poweredOn",
  standbyMode = <unset>,
  inMaintenanceMode = false,
  bootTime = "2010-08-13T09:20:18.58907Z",
  healthSystemRuntime = (vim.host.HealthStatusSystem.Runtime) {
    dynamicType = <unset>,
    systemHealthInfo = (vim.host.SystemHealthInfo) {
      dynamicType = <unset>,
      numericSensorInfo = (vim.host.NumericSensorInfo) [
        (vim.host.NumericSensorInfo) {
          dynamicType = <unset>,
          name = "VMware Rollup Health State",
          healthState = (vim.ElementDescription) {
            dynamicType = <unset>,
            label = "Green",
            summary = "Sensor is operating under normal conditions",
            key = "green",
          },
          currentReading = 0,
          unitModifier = 0,
          baseUnits = "",
          rateUnits = <unset>,
          sensorType = "system",
        },
        (vim.host.NumericSensorInfo) {
          dynamicType = <unset>,
          name = "Phoenix Technologies LTD System BIOS 6.00 2009-12-31 00:00:00.000",
          healthState = (vim.ElementDescription) {
            dynamicType = <unset>,
            label = "Green",
            summary = "Sensor is operating under normal conditions",
            key = "green",
          },
        },
      ],
    },
  },
  ...
}
```

hostsvc/net/

* hostsvc/net/mockup/

** hostsvc/net/mockup/clear*

** hostsvc/net/mockup/pnic_add*

** hostsvc/net/mockup/pnic_remove*

** hostsvc/net/mockup/pnic_setactuallinkspeed*

* hostsvc/net/portgroup_remove

hostsvc/net/provider/

/hostsvc/net/provider/test_abort

~ # vim-cmd /hostsvc/net/provider/test_abort

Test Succeeded

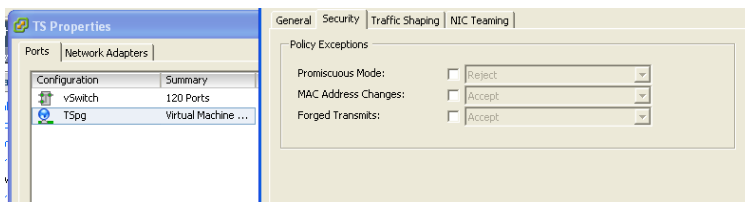
Logger output:

```
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] ActionDummy: Do() called on 0
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] ActionDummy: Do() called on 1
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] ActionDummy: Do() called on 2
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] Aborting network system provider pseudo-transaction
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] ActionDummy: Undo() called on 2
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] ActionDummy: Undo() called on 1
[2010-08-13 20:35:44.693 FFA85E90 error 'NetworkProvider'] NetworkSystemProviderTrans::Abort failed to undo an action
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] Aborting network system provider pseudo-transaction
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] ActionDummy: Undo() called on 1
[2010-08-13 20:35:44.693 FFA85E90 verbose 'NetworkProvider'] ActionDummy: Undo() called on 0
[2010-08-13 20:35:44.693 FFA85E90 info 'NetworkProvider'] TestAbort succeeded
```

hostsvc/net/portgroup_set

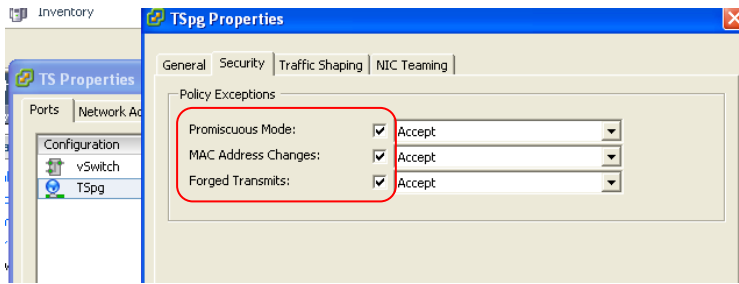
--securepolicy-promisc=bool

Allow promiscuous mode.



~ # vim-cmd hostsvc/net/portgroup_set --securepolicy-promisc=true TS TSpg

~ #

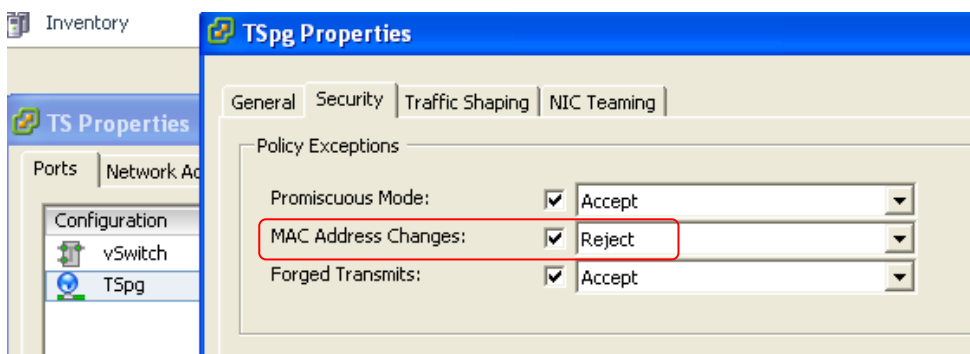


--securepolicy-macchange=bool

Allow MAC address to be changed.

```
~ # vim-cmd hostsvc/net/portgroup_set --securepolicy-macchange=false TS TSpg
```

```
~ #
```

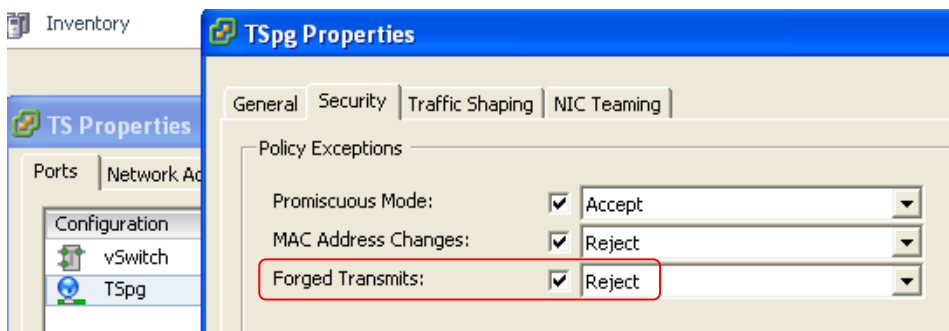


--securepolicy-forgedxmit=bool

Allow forged transmits.

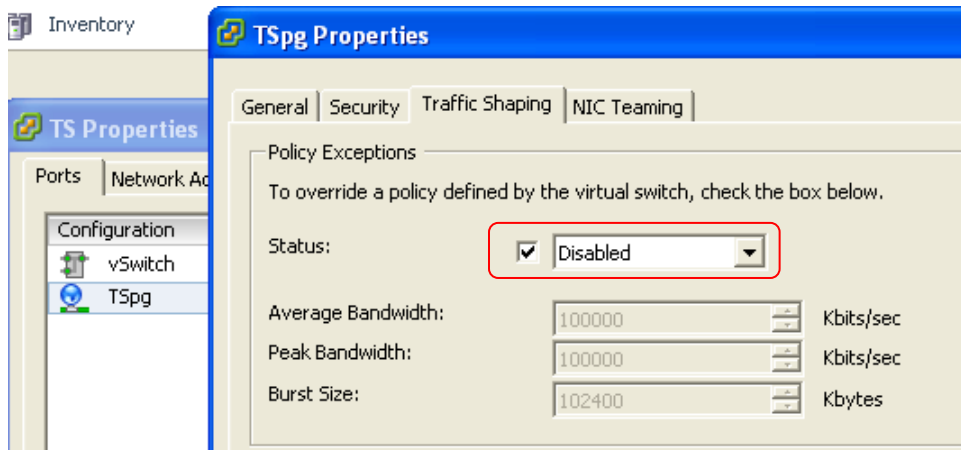
```
~ # vim-cmd hostsvc/net/portgroup_set --securepolicy-forgedxmit=false TS TSpg
```

```
~ #
```



--shapingpolicy-enabled=bool

Enabled shaper.



```
~ # vim-cmd hostsvc/net/portgroup_set --shapingpolicy-enabled=true TS TSpg
```

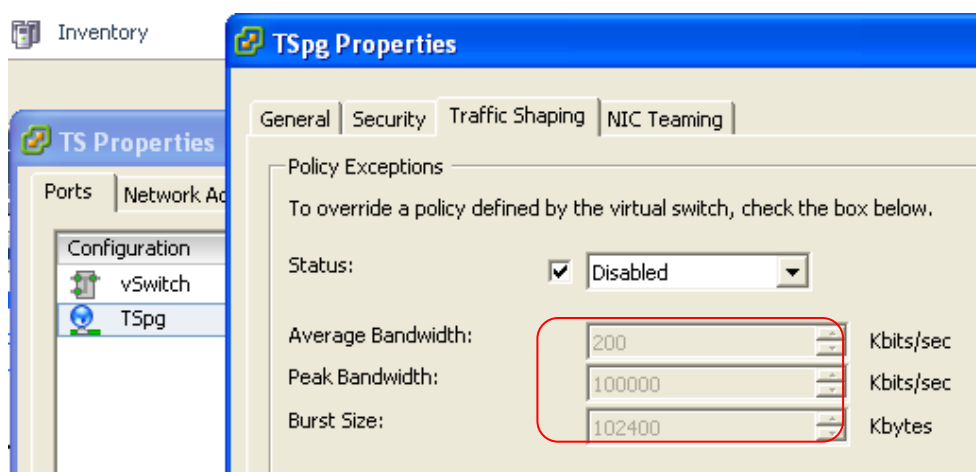
```
(vmodl.fault.InvalidArgument) {
  dynamicType = <unset>,
  faultCause = (vmodl.MethodFault) null,
  invalidProperty = <unset>,
  msg = "A specified parameter was not correct."
}
```

--shapingpolicy-average-bandwidth=int

Average bandwidth.

```
~ # vim-cmd hostsvc/net/portgroup_set --shapingpolicy-average-bandwidth=200000 TS TSpg
```

```
~ #
```

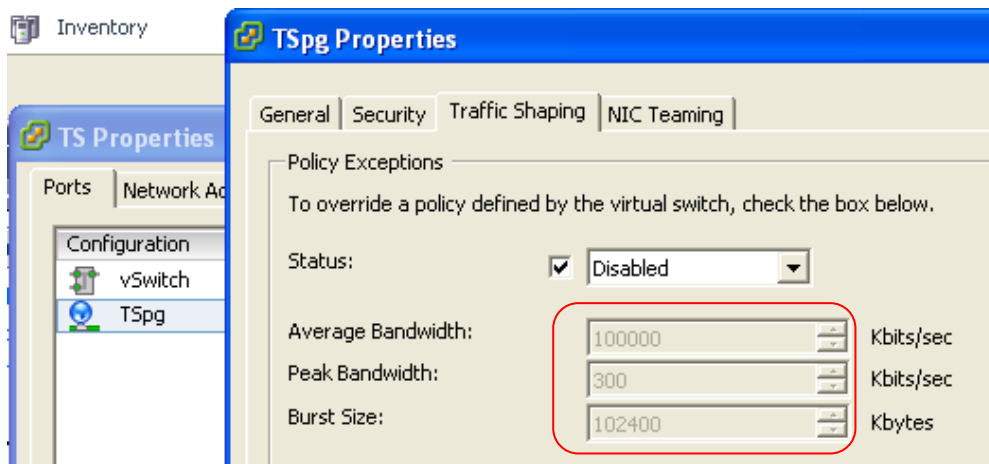


--shapingpolicy-peak-bandwidth=int

Peak bandwidth.

```
~ # vim-cmd hostsvc/net/portgroup_set --shapingpolicy-peak-bandwidth=300000 TS TSpg
```

~ #

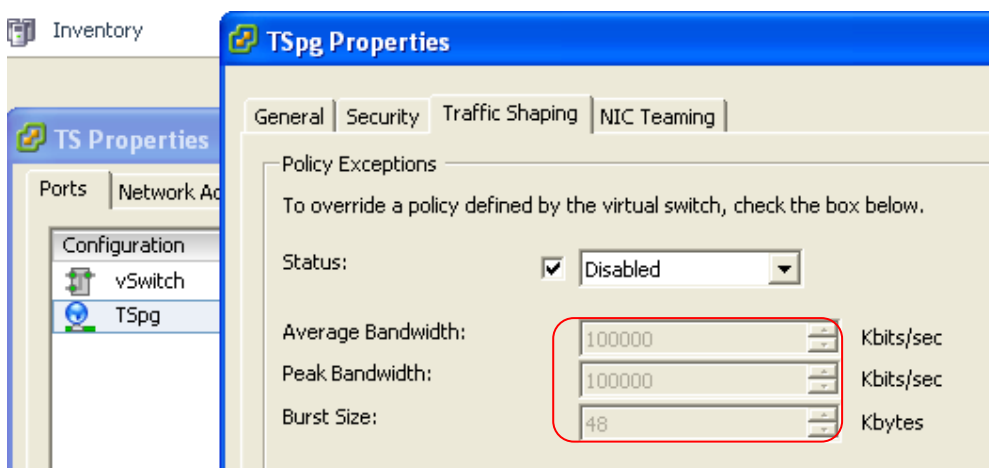


--shapingpolicy-burst-size=int

Burst size.

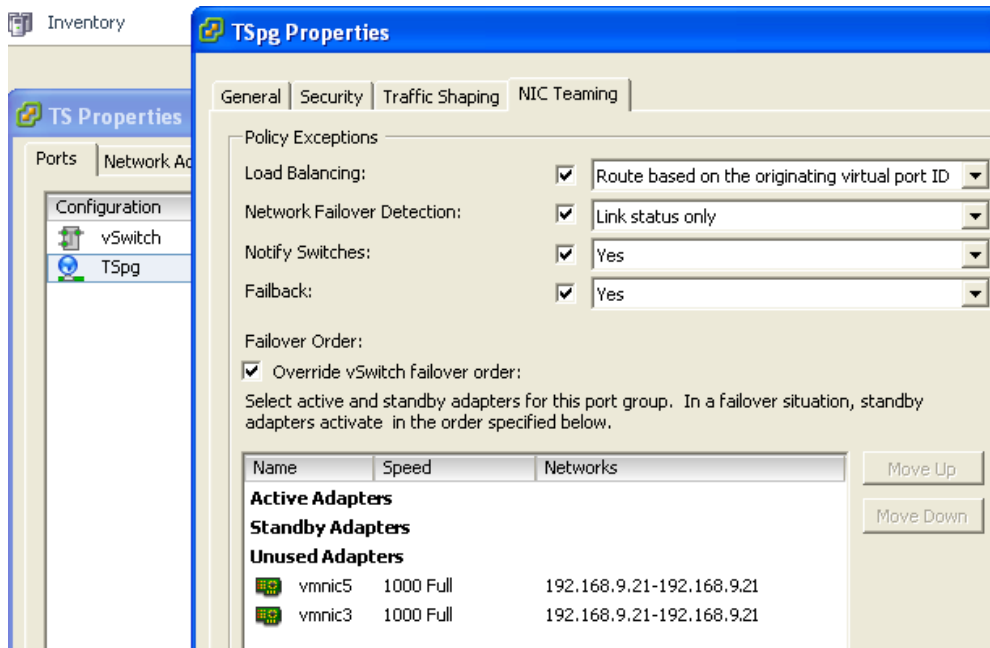
~ # vim-cmd hostsvc/net/portgroup_set --shapingpolicy-burst-size=50000 TS TSpg

~ #



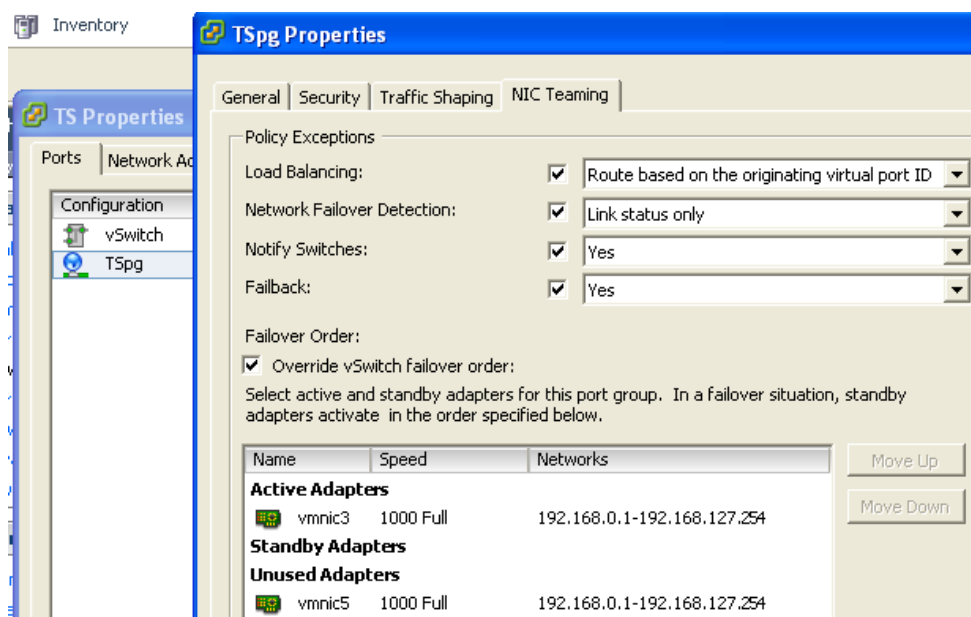
--nicorderpolicy-active=vmnic_list

List of active NICs.



```
~ # vim-cmd hostsvc/net/portgroup_set --nicorderpolicy-active=vmnic3 TS TSpg
```

```
~ #
```

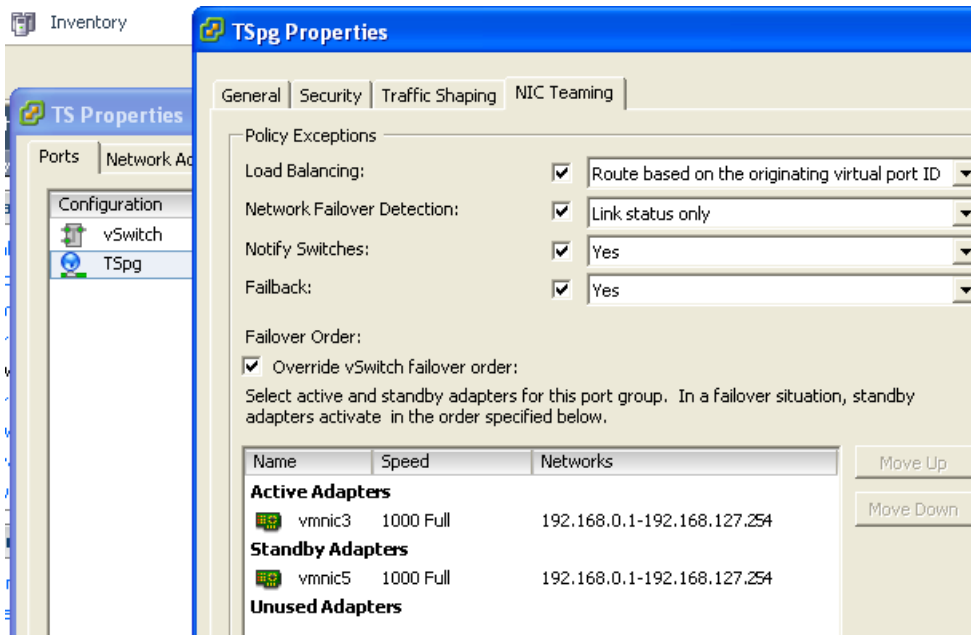


```
--nicorderpolicy-standby=vmnic_list
```

Order in which failover should occur.

```
~ # vim-cmd hostsvc/net/portgroup_set --nicorderpolicy-standby=vmnic5 TS TSpg
```

```
~ #
```



--failurecriteria-check-speed=failurecriteria-check-speed

Detecting failover using link speed check

--failurecriteria-speed=int

The speed for link speed check method

--failurecriteria-check-duplex=bool

Detecting failover using link duplex check

--failurecriteria-duplex=bool

The duplexity for link duplex check method

--failurecriteria-check-error=bool

Detecting failover using link error percentage

--failurecriteria-error=int

The error percentage for link error percentage check method

```
~ # vim-cmd hostsvc/net/portgroup_set --failurecriteria-check-error=true --failurecriteria-error=50 TS TSpg
```

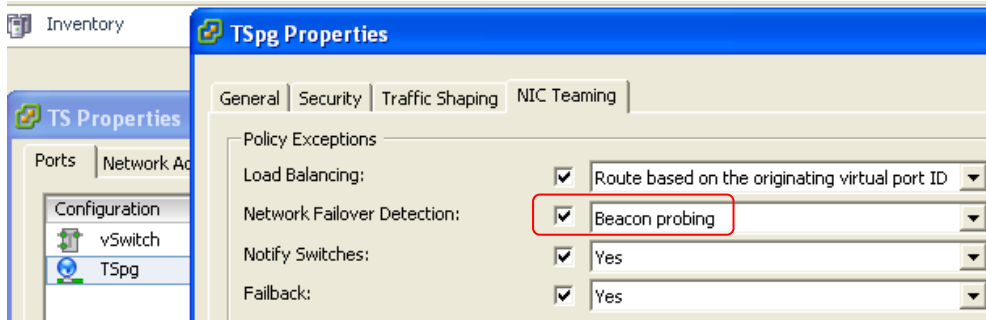
```
~ #
```

--failurecriteria-check-beacon=bool

Detecting failover using the beacon

```
~ # vim-cmd hostsvc/net/portgroup_set --failurecriteria-check-beacon=true TS TSpG
```

```
~ #
```

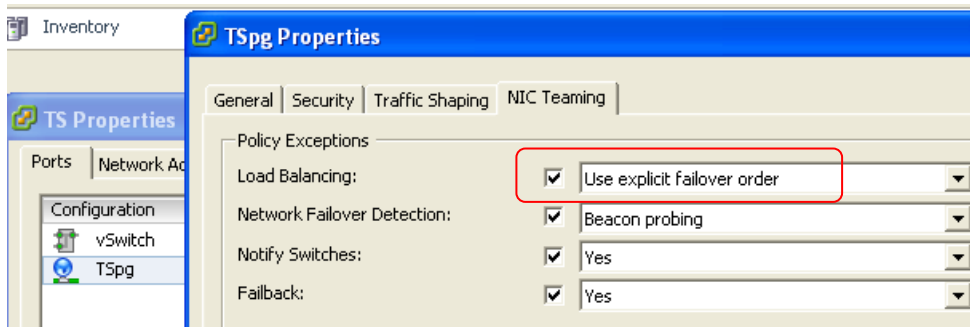


--nicteaming-policy=nicteaming-policy

Nic-Teaming policy.

```
~ # vim-cmd hostsvc/net/portgroup_set --nicteaming-policy=failover_explicit TS TSpG
```

```
~ #
```



--nicteaming-reverse-policy=bool

Apply the teaming policy to inbound frames as well.

--nicteaming-notify-switch=bool

Notify switches when detects a link failure.

--nicteaming-rollingorder=bool

Whether or not to use rolling failover.

--offloadcaps-csum-offload=bool

Checksum offload

--offloadcaps-tcp-segment=bool

TCP segment offload

--offloadcaps-zero-copy-xmit=bool

Scatter gather

hostsvc/net/capabilities

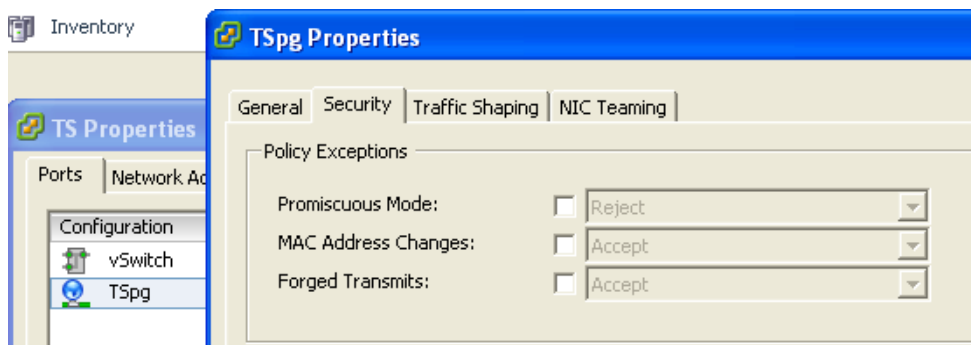
~ # vim-cmd /hostsvc/net/capabilities

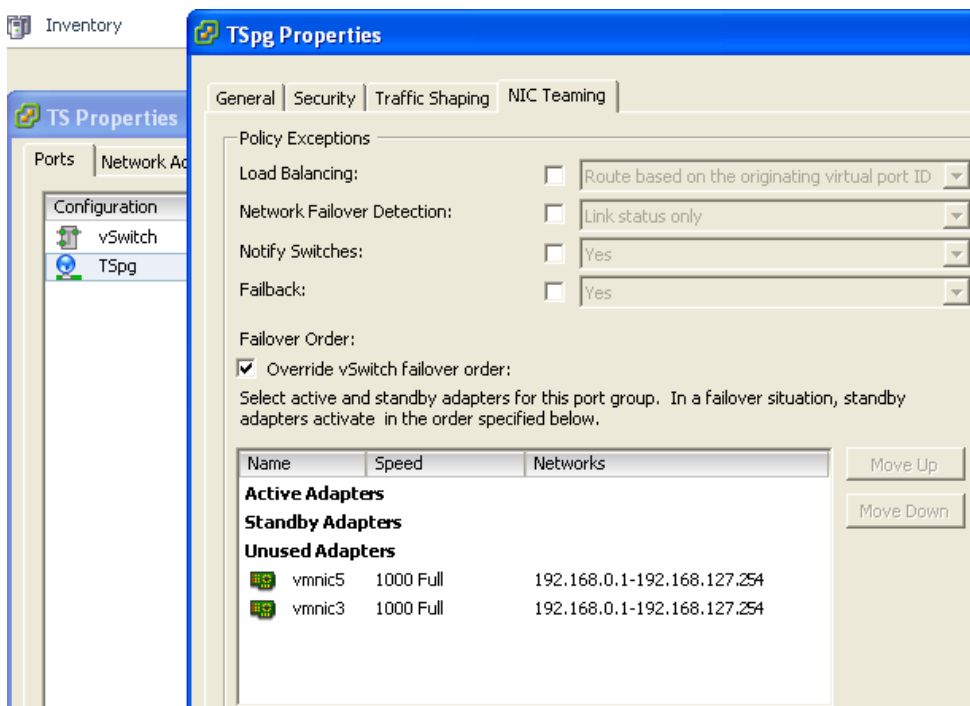
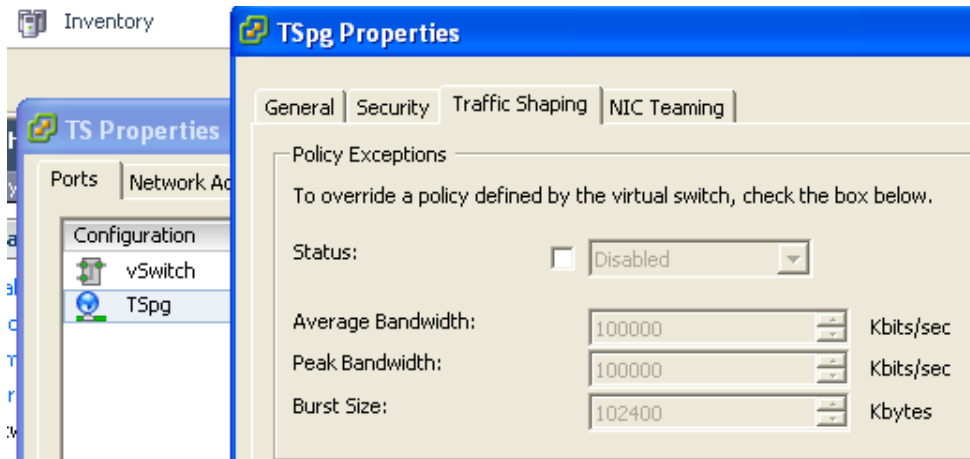
```
(vim.host.NetCapabilities) {
  dynamicType = <unset>,
  canSetPhysicalNicLinkSpeed = true,
  supportsNicTeaming = true,
  nicTeamingPolicy = (string) [
    "loadbalance_ip",
    "loadbalance_srcmac",
    "loadbalance_srcid",
    "failover_explicit"
  ],
  supportsVlan = true,
  usesServiceConsoleNic = false,
  supportsNetworkHints = true,
  maxPortGroupsPerVswitch = <unset>,
  vswitchConfigSupported = true,
  vnicConfigSupported = true,
  ipRouteConfigSupported = true,
  dnsConfigSupported = true,
  dhcpOnVnicSupported = true,
  ipV6Supported = true,
}
```

hostsvc/net/portgroup_unsetpolicy

~ # vim-cmd hostsvc/net/portgroup_unsetpolicy TS TSpg

~ #





hostsvc/net/config

~ # vim-cmd /hostsvc/net/config

```
(vim.host.NetworkConfig) {
  dynamicType = <unset>,
  vswitch = (vim.host.VirtualSwitch.Config) [
    (vim.host.VirtualSwitch.Config) {
      dynamicType = <unset>,
      changeOperation = <unset>,
      name = "vSS-Management",
      spec = (vim.host.VirtualSwitch.Specification) {
        dynamicType = <unset>,
        numPorts = 128,
        bridge = (vim.host.VirtualSwitch.BondBridge) {
          dynamicType = <unset>,
          nicDevice = (string) [
            "vmnic0"
          ],
        },
      },
    ],
  ],
}
```

```

beacon = (vim.host.VirtualSwitch.BeaconConfig) {
  dynamicType = <unset>,
  interval = 1,
},
linkDiscoveryProtocolConfig = (vim.host.LinkDiscoveryProtocolConfig) {
  dynamicType = <unset>,
  protocol = "cdp",
  operation = "both",
},
},
policy = (vim.host.NetworkPolicy) {
  dynamicType = <unset>,
  security = (vim.host.NetworkPolicy.SecurityPolicy) {
    dynamicType = <unset>,
    allowPromiscuous = false,
    macChanges = true,
    forgedTransmits = true,
  },
  nicTeaming = (vim.host.NetworkPolicy.NicTeamingPolicy) {
    dynamicType = <unset>,
    policy = "loadbalance_srcid",
    reversePolicy = true,
    notifySwitches = true,
    rollingOrder = false,
    failureCriteria = (vim.host.NetworkPolicy.NicFailureCriteria) {
      dynamicType = <unset>,
      checkSpeed = "minimum",
      speed = 10,
      checkDuplex = false,
      fullDuplex = false,
    }
  }
}

```

...

hostsvc/net/query_networkhint

~ # vim-cmd hostsvc/net/query_networkhint

```

(vim.host.PhysicalNic.NetworkHint) [
  (vim.host.PhysicalNic.NetworkHint) {
    dynamicType = <unset>,
    device = "vmmnic0",
    subnet = (vim.host.PhysicalNic.NetworkHint.IpNetwork) [
      (vim.host.PhysicalNic.NetworkHint.IpNetwork) {
        dynamicType = <unset>,
        vlanId = 0,
        ipSubnet = "192.168.0.1-192.168.127.254",
      }
    ],
    connectedSwitchPort = (vim.host.PhysicalNic.CdpInfo) null,
  },
  (vim.host.PhysicalNic.NetworkHint) {
    dynamicType = <unset>,
    device = "vmmnic1",
    subnet = (vim.host.PhysicalNic.NetworkHint.IpNetwork) [
      (vim.host.PhysicalNic.NetworkHint.IpNetwork) {
        dynamicType = <unset>,

```

```

    vlanId = 0,
    ipSubnet = "192.168.0.1-192.168.127.254",
  }
],
connectedSwitchPort = (vim.host.PhysicalNic.CdpInfo) {
  dynamicType = <unset>,
  cdpVersion = 2,
  timeout = 0,
  ttl = 171,
  samples = 425,
  devId = "esxi-21.home.hc",
  address = "0.0.0.0",
  portId = "vmnic0",
  deviceCapability = (vim.host.PhysicalNic.CdpDeviceCapability) {
    dynamicType = <unset>,
    router = false,
    transparentBridge = false,
    sourceRouteBridge = false,
    networkSwitch = true,
    host = false,
    igmpEnabled = false,
    repeater = false,
  },
  softwareVersion = "Releasebuild-260247",
  hardwarePlatform = "VMware ESX",
  ipPrefix = "0.0.0.0",
  ipPrefixLen = 0,
  vlan = 0,
  fullDuplex = true,
  mtu = 1500,
  systemName = "esxi-21.home.hc",
  systemOID = "",
  mgmtAddr = "0.0.0.0",
  location = "",
},
},
(vim.host.PhysicalNic.NetworkHint) {
  dynamicType = <unset>,
  device = "vmnic2",
  subnet = (vim.host.PhysicalNic.NetworkHint.IpNetwork) [
    (vim.host.PhysicalNic.NetworkHint.IpNetwork) {
      dynamicType = <unset>,
      vlanId = 0,
      ipSubnet = "192.168.0.1-192.168.127.254",
    }
  ],
connectedSwitchPort = (vim.host.PhysicalNic.CdpInfo) {
  dynamicType = <unset>,
  cdpVersion = 2,
  timeout = 0,
  ttl = 171,
  samples = 425,
  devId = "esxi-21.home.hc",
  address = "0.0.0.0",

```

```

portId = "vmnic0",
deviceCapability = (vim.host.PhysicalNic.CdpDeviceCapability) {
  dynamicType = <unset>,
  router = false,
  transparentBridge = false,
  sourceRouteBridge = false,
  networkSwitch = true,
  host = false,
  igmpEnabled = false,
  repeater = false,
},
softwareVersion = "Releasebuild-260247",
hardwarePlatform = "VMware ESX",
ipPrefix = "0.0.0.0",
ipPrefixLen = 0,
vlan = 0,
fullDuplex = true,
mtu = 1500,
systemName = "esxi-21.home.hc",
systemOID = "",
mgmtAddr = "0.0.0.0",
location = "",
},
},
(vim.host.PhysicalNic.NetworkHint) {
  dynamicType = <unset>,
  device = "vmnic3",
  subnet = (vim.host.PhysicalNic.NetworkHint.IpNetwork) [
    (vim.host.PhysicalNic.NetworkHint.IpNetwork) {
      dynamicType = <unset>,
      vlanId = 0,
      ipSubnet = "192.168.0.1-192.168.127.254",
    }
  ],
connectedSwitchPort = (vim.host.PhysicalNic.CdpInfo) {
  dynamicType = <unset>,
  cdpVersion = 2,
  timeout = 0,
  ttl = 171,
  samples = 425,
  devId = "esxi-21.home.hc",
  address = "0.0.0.0",
  portId = "vmnic0",
  deviceCapability = (vim.host.PhysicalNic.CdpDeviceCapability) {
    dynamicType = <unset>,
    router = false,
    transparentBridge = false,
    sourceRouteBridge = false,
    networkSwitch = true,
    host = false,
    igmpEnabled = false,
    repeater = false,
  },
softwareVersion = "Releasebuild-260247",

```

```

hardwarePlatform = "VMware ESX",
ipPrefix = "0.0.0.0",
ipPrefixLen = 0,
vlan = 0,
fullDuplex = true,
mtu = 1500,
systemName = "esxi-21.home.hc",
systemOID = "",
mgmtAddr = "0.0.0.0",
location = "",
},
},
(vim.host.PhysicalNic.NetworkHint) {
dynamicType = <unset>,
device = "vmnic4",
subnet = (vim.host.PhysicalNic.NetworkHint.IpNetwork) [
(vim.host.PhysicalNic.NetworkHint.IpNetwork) {
dynamicType = <unset>,
vlanId = 0,
ipSubnet = "192.168.0.1-192.168.127.254",
}
],
connectedSwitchPort = (vim.host.PhysicalNic.CdpInfo) {
dynamicType = <unset>,
cdpVersion = 2,
timeout = 0,
ttl = 171,
samples = 425,
devId = "esxi-21.home.hc",
address = "0.0.0.0",
portId = "vmnic0",
deviceCapability = (vim.host.PhysicalNic.CdpDeviceCapability) {
dynamicType = <unset>,
router = false,
transparentBridge = false,
sourceRouteBridge = false,
networkSwitch = true,
host = false,
igmpEnabled = false,
repeater = false,
},
softwareVersion = "Releasebuild-260247",
hardwarePlatform = "VMware ESX",
ipPrefix = "0.0.0.0",
ipPrefixLen = 0,
vlan = 0,
fullDuplex = true,
mtu = 1500,
systemName = "esxi-21.home.hc",
systemOID = "",
mgmtAddr = "0.0.0.0",
location = "",
},
},
},

```

```

(vim.host.PhysicalNic.NetworkHint) {
  dynamicType = <unset>,
  device = "vmnic5",
  subnet = (vim.host.PhysicalNic.NetworkHint.IpNetwork) [
    (vim.host.PhysicalNic.NetworkHint.IpNetwork) {
      dynamicType = <unset>,
      vlanId = 0,
      ipSubnet = "192.168.0.1-192.168.127.254",
    }
  ],
  connectedSwitchPort = (vim.host.PhysicalNic.CdpInfo) {
    dynamicType = <unset>,
    cdpVersion = 2,
    timeout = 0,
    ttl = 171,
    samples = 425,
    devId = "esxi-21.home.hc",
    address = "0.0.0.0",
    portId = "vmnic0",
    deviceCapability = (vim.host.PhysicalNic.CdpDeviceCapability) {
      dynamicType = <unset>,
      router = false,
      transparentBridge = false,
      sourceRouteBridge = false,
      networkSwitch = true,
      host = false,
      igmpEnabled = false,
      repeater = false,
    },
    softwareVersion = "Releasebuild-260247",
    hardwarePlatform = "VMware ESX",
    ipPrefix = "0.0.0.0",
    ipPrefixLen = 0,
    vlan = 0,
    fullDuplex = true,
    mtu = 1500,
    systemName = "esxi-21.home.hc",
    systemOID = "",
    mgmtAddr = "0.0.0.0",
    location = "",
  },
}
]

```

hostsvc/net/config_append

~ # vim-cmd hostsvc/net/config_append

Insufficient arguments.

Usage: config_append filename

Appends a network configuration from a file and updates it.

hostsvc/net/refresh

```
~ # vim-cmd /hostsvc/net/vswitch_unsetbridge TS
```

```
(vmodl.fault.InvalidArgument) {
  dynamicType = <unset>,
  faultCause = (vmodl.MethodFault) null,
  invalidProperty = <unset>,
  msg = "A specified parameter was not correct.
",
}
```

```
~ # vim-cmd /hostsvc/net/refresh
```

```
~ # vim-cmd /hostsvc/net/vswitch_unsetbridge TS
```

```
~ #
```

*** hostsvc/net/config_incremental_update****hostsvc/net/route_info**

```
~ # vim-cmd hostsvc/net/route_info
```

VMKernel Forwarding Table

```
(vim.host.IpRouteConfig) {
  dynamicType = <unset>,
  defaultGateway = "192.168.9.9",
  gatewayDevice = <unset>,
  ipv6DefaultGateway = <unset>,
  ipv6GatewayDevice = <unset>,
}
```

Console OS Forwarding Table

```
(vim.host.IpRouteConfig) {
  dynamicType = <unset>,
  defaultGateway = <unset>,
  gatewayDevice = <unset>,
  ipv6DefaultGateway = <unset>,
  ipv6GatewayDevice = <unset>,
}
```

hostsvc/net/config_load

```
~ # vim-cmd hostsvc/net/config_load
```

Insufficient arguments.

Usage: config_load filename

Loads a network configuration from a file and updates it.

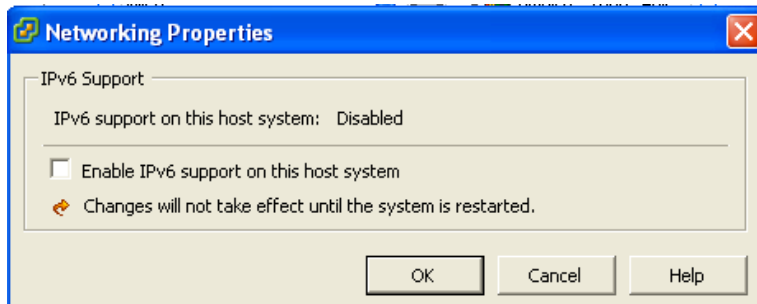
*** hostsvc/net/route_set**

hostsvc/net/config_save

```
~ # vim-cmd /hostsvc/net/config_save /vmfs/volumes/datastore1/netconf.xml
```

```
~ #
```

```
<ConfigRoot>
  <_type>vim.host.NetworkConfig</_type>
  <dnsConfig>
    <_type>vim.host.DnsConfig</_type>
    <address>
      <_length>2</_length>
      <_type>string[]</_type>
      <e id="0">192.168.9.231</e>
      <e id="1">192.168.9.222</e>
    </address>
    <dhcp>>false</dhcp>
    <domainName>home.hc</domainName>
    <hostName>esxi-21</hostName>
  </dnsConfig>
  <ipV6Enabled>>false</ipV6Enabled>
  <pnic>
    <_length>6</_length>
    <_type>vim.host.PhysicalNic.Config[]</_type>
    <e id="0">
      <_type>vim.host.PhysicalNic.Config</_type>
      <device>vmnic0</device>
      <spec>
        <_type>vim.host.PhysicalNic.Specification</_type>
        <ip>
          <_type>vim.host.IpConfig</_type>
          <dhcp>>false</dhcp>
          <ipAddress/>
          <subnetMask/>
        </ip>
        <linkSpeed>
          <_type>vim.host.PhysicalNic.LinkSpeedDuplex</_type>
          <duplex>>true</duplex>
          <speedMb>1000</speedMb>
        </linkSpeed>
      </spec>
    </e>
  </pnic>
  ...
```

hostsvc/net/set_v6stack

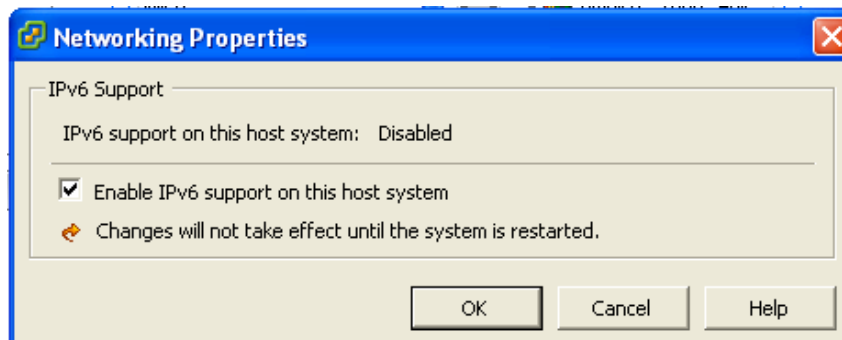
```
~ # vim-cmd hostsvc/net/set_v6stack true
```

Sending this over...

```
(vim.host.NetworkConfig) {
  dynamicType = <unset>,
  dnsConfig = (vim.host.DnsConfig) null,
  ipRouteConfig = (vim.host.IpRouteConfig) null,
  consoleIpRouteConfig = (vim.host.IpRouteConfig) null,
  routeTableConfig = (vim.host.IpRouteTableConfig) null,
  ipv6Enabled = true,
  ipSecConfig = (vim.host.IpSecConfig) null,
  consoleIpSecConfig = (vim.host.IpSecConfig) null,
}
```

Change status report

```
(vim.host.NetworkConfig.Result) {
  dynamicType = <unset>,
}
```



* **hostsvc/net/console_route_set**

* **hostsvc/net/toggle_raw**

* `hostsvc/net/consolevnic_add`

`--ip-dhcp=bool`

Enables or disables DHCP configuration.

`--ip-address=ip_address`

IP address.

`--ip-subnet=ip_address`

Subnet mask.

`--dynamic-addrs=ip_addr_origin`

Where to obtain address: dhcpv6, autoconf, both, none.

`--ipv6-addresses=string_list`

Provide IP address(es)[/prefix] separated by comma, prefix defaults to 64.

`--replace=bool`

replace existing manual v6 addresses, default is to merge

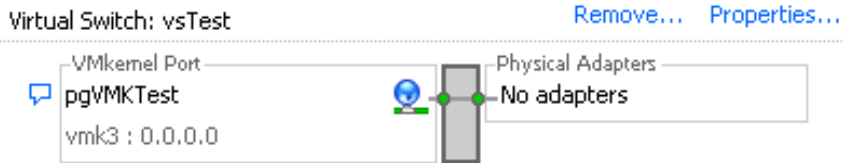
* `hostsvc/net/vnic_add`



```
~ # vim-cmd hostsvc/net/vnic_add --ip-dhcp=true pgVMKTest
```

Sending this over...

```
(vim.host.VirtualNic.Specification) {
  dynamicType = <unset>,
  ip = (vim.host.IpConfig) {
    dynamicType = <unset>,
    dhcp = true,
    ipAddress = <unset>,
    subnetMask = <unset>,
    ipv6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
  },
  mac = <unset>,
  distributedVirtualPort = (vim.dvs.PortConnection) null,
  portgroup = <unset>,
  mtu = <unset>,
  tsoEnabled = <unset>,
}
```



hostsvc/net/consolevnic_info

```
~ # vim-cmd hostsvc/net/consolevnic_info
```

```
(vim.host.VirtualNic) []
```

hostsvc/net/vnic_info

```
~ # vim-cmd hostsvc/net/vnic_info
```

```
(vim.host.VirtualNic) [
```

```
(vim.host.VirtualNic) {
```

```
  dynamicType = <unset>,
```

```
  device = "vmk0",
```

```
  key = "key-vim.host.VirtualNic-vmk0",
```

```
  portgroup = "Management",
```

```
  spec = (vim.host.VirtualNic.Specification) {
```

```
    dynamicType = <unset>,
```

```
    ip = (vim.host.IpConfig) {
```

```
      dynamicType = <unset>,
```

```
      dhcp = false,
```

```
      ipAddress = "192.168.9.201",
```

```
      subnetMask = "255.255.255.0",
```

```
      ipV6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
```

```
    },
```

```
    mac = "00:50:56:71:bd:7a",
```

```
    distributedVirtualPort = (vim.dvs.PortConnection) null,
```

```
    portgroup = "Management",
```

```
    mtu = 1500,
```

```
    tsoEnabled = true,
```

```
  },
```

```
  port = <vim.host.PortGroup.Port:key-vim.host.PortGroup.Port-16777219>,
```

```
},
```

```
(vim.host.VirtualNic) {
```

```
  dynamicType = <unset>,
```

```
  device = "vmk1",
```

```
  key = "key-vim.host.VirtualNic-vmk1",
```

```
  portgroup = "VMk-SWiSCSI-PG-1",
```

```
  spec = (vim.host.VirtualNic.Specification) {
```

```
    dynamicType = <unset>,
```

```
    ip = (vim.host.IpConfig) {
```

```
      dynamicType = <unset>,
```

```
      dhcp = false,
```

```
      ipAddress = "192.168.9.202",
```

```
      subnetMask = "255.255.255.0",
```

```
      ipV6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
```

```
    },
```

```
    mac = "00:50:56:77:00:94",
```

```
    distributedVirtualPort = (vim.dvs.PortConnection) null,
```

```

    portgroup = "VMk-SWiSCSI-PG-1",
    mtu = 1500,
    tsoEnabled = true,
  },
  port = <vim.host.PortGroup.Port:key-vim.host.PortGroup.Port-33554435>,
},
(vim.host.VirtualNic) {
  dynamicType = <unset>,
  device = "vmk2",
  key = "key-vim.host.VirtualNic-vmk2",
  portgroup = "VMk-SWiSCSI-PG-2",
  spec = (vim.host.VirtualNic.Specification) {
    dynamicType = <unset>,
    ip = (vim.host.IpConfig) {
      dynamicType = <unset>,
      dhcp = false,
      ipAddress = "192.168.9.203",
      subnetMask = "255.255.255.0",
      ipV6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
    },
    mac = "00:50:56:7f:d2:ea",
    distributedVirtualPort = (vim.dvs.PortConnection) null,
    portgroup = "VMk-SWiSCSI-PG-2",
    mtu = 1500,
    tsoEnabled = true,
  },
  port = <vim.host.PortGroup.Port:key-vim.host.PortGroup.Port-50331651>,
},
(vim.host.VirtualNic) {
  dynamicType = <unset>,
  device = "vmk3",
  key = "key-vim.host.VirtualNic-vmk3",
  portgroup = "pgVMKTest",
  spec = (vim.host.VirtualNic.Specification) {
    dynamicType = <unset>,
    ip = (vim.host.IpConfig) {
      dynamicType = <unset>,
      dhcp = true,
      ipAddress = "169.254.0.1",
      subnetMask = "255.255.0.0",
      ipV6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
    },
    mac = "00:50:56:7b:6b:a2",
    distributedVirtualPort = (vim.dvs.PortConnection) null,
    portgroup = "pgVMKTest",
    mtu = 1500,
    tsoEnabled = true,
  },
  port = <vim.host.PortGroup.Port:key-vim.host.PortGroup.Port-67108866>,
}
]

```

* `hostsvc/net/consolevnic_remove`

`hostsvc/net/vnic_remove`



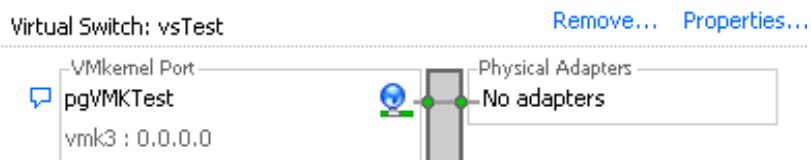
~ # vim-cmd hostsvc/net/vnic_remove vmk3

~ #



* `hostsvc/net/consolevnic_set`

* `hostsvc/net/vnic_set`



`--vnic-mac=string`

Sets the mac address of this vnic.

`--ip-dhcp=bool`

Enables or disables DHCP configuration.

`--ip-address=ip_address`

IP address.

`--ip-subnet=ip_address`

Subnet mask.

~ # vim-cmd hostsvc/net/vnic_set --ip-address 192.168.100.100 --ip-subnet 255.255.255.0 vmk3

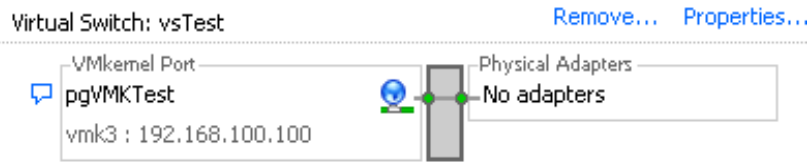
Sending this over...

```
(vim.host.VirtualNic.Specification) {
  dynamicType = <unset>,
  ip = (vim.host.IpConfig) {
    dynamicType = <unset>,
    dhcp = false,
    ipAddress = "192.168.100.100",
    subnetMask = "255.255.255.0",
    ipv6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
```

```

},
mac = "00:50:56:7b:6b:a2",
distributedVirtualPort = (vim.dvs.PortConnection) null,
portgroup = "pgVMKTest",
mtu = 1500,
tsoEnabled = true,
}

```



* [hostsvc/net/consolevnic_setv6](#)

[hostsvc/net/vnic_setv6](#)

Usage: `vnic_setv6 [OPTIONS] vnic`

Change IPv6 addressing on a vmkernel nic.

Options:

`--vnic-mac=string`

Sets the mac address of this vnic.

`--dynamic-addr=ip_addr_origin`

Where to obtain address: dhcpv6, autoconf, both, none.

`--ipv6-addresses=string_list`

Provide IP address(es)[/prefix] separated by comma, prefix defaults to 64.

`--replace=bool`

replace existing manual v6 addresses, default is to merge

[hostsvc/net/dns_info](#)

~ # `vim-cmd /hostsvc/net/dns_info`

```

(vim.host.DnsConfig) {
  dynamicType = <unset>,
  dhcp = false,
  virtualNicDevice = <unset>,
  hostName = "esxi-21",
  domainName = "home.hc",
  address = (string) [
    "192.168.9.231",
    "192.168.9.222"
  ],
}

```

hostsvc/net/vswitch_add

~ # vim-cmd /hostsvc/net/vswitch_add TS

~ #



* hostsvc/net/dns_set

hostsvc/net/vswitch_info

~ # vim-cmd /hostsvc/net/vswitch_info TS

```
(vim.host.VirtualSwitch) {
  dynamicType = <unset>,
  name = "TS",
  key = "key-vim.host.VirtualSwitch-TS",
  numPorts = 128,
  numPortsAvailable = 127,
  mtu = 1500,
  portgroup = (vim.host.PortGroup) [
    <vim.host.PortGroup:key-vim.host.PortGroup-TSpg>
  ],
  spec = (vim.host.VirtualSwitch.Specification) {
    dynamicType = <unset>,
    numPorts = 128,
    bridge = (vim.host.VirtualSwitch.Bridge) null,
    policy = (vim.host.NetworkPolicy) {
      dynamicType = <unset>,
      security = (vim.host.NetworkPolicy.SecurityPolicy) {
        dynamicType = <unset>,
        allowPromiscuous = false,
        macChanges = true,
        forgedTransmits = true,
      },
      nicTeaming = (vim.host.NetworkPolicy.NicTeamingPolicy) {
        dynamicType = <unset>,
        policy = "loadbalance_srcid",
        reversePolicy = true,
        notifySwitches = true,
        rollingOrder = false,
        failureCriteria = (vim.host.NetworkPolicy.NicFailureCriteria) {
          dynamicType = <unset>,
          checkSpeed = "minimum",
          speed = 10,
          checkDuplex = false,
          fullDuplex = false,
          checkErrorPercent = false,
          percentage = 0,
        },
      },
    },
  },
}
```

```

        checkBeacon = false,
    },
    nicOrder = (vim.host.NetworkPolicy.NicOrderPolicy) {
        dynamicType = <unset>,
    },
},
offloadPolicy = (vim.host.NetOffloadCapabilities) {
    dynamicType = <unset>,
    csumOffload = true,
    tcpSegmentation = true,
    zeroCopyXmit = true,
},
shapingPolicy = (vim.host.NetworkPolicy.TrafficShapingPolicy) {
    dynamicType = <unset>,
    enabled = false,
    averageBandwidth = <unset>,
    peakBandwidth = <unset>,
    burstSize = <unset>,
},
},
mtu = <unset>,
},
}

```

* **hostsvc/net/dns_unset**

hostsvc/net/vswitch_pnicinfo

~ # vim-cmd hostsvc/net/vswitch_pnicinfo TS

```

(vim.host.PhysicalNic) [
(vim.host.PhysicalNic) {
    dynamicType = <unset>,
    key = "key-vim.host.PhysicalNic-vmnic5",
    device = "vmnic5",
    pci = "02:08.0",
    driver = "e1000",
    linkSpeed = (vim.host.PhysicalNic.LinkSpeedDuplex) {
        dynamicType = <unset>,
        speedMb = 1000,
        duplex = true,
    },
    validLinkSpecification = (vim.host.PhysicalNic.LinkSpeedDuplex) [
    (vim.host.PhysicalNic.LinkSpeedDuplex) {
        dynamicType = <unset>,
        speedMb = 10,
        duplex = false,
    },
    (vim.host.PhysicalNic.LinkSpeedDuplex) {
        dynamicType = <unset>,
        speedMb = 10,
        duplex = true,
    },
    (vim.host.PhysicalNic.LinkSpeedDuplex) {

```

```

    dynamicType = <unset>,
    speedMb = 100,
    duplex = false,
  },
  (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 100,
    duplex = true,
  },
  (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 1000,
    duplex = true,
  }
},
spec = (vim.host.PhysicalNic.Specification) {
  dynamicType = <unset>,
  ip = (vim.host.IpConfig) {
    dynamicType = <unset>,
    dhcp = false,
    ipAddress = "",
    subnetMask = "",
    ipv6Config = (vim.host.IpConfig.Ipv6AddressConfiguration) null,
  },
  linkSpeed = (vim.host.PhysicalNic.LinkSpeedDuplex) null,
},
wakeOnLanSupported = false,
mac = "00:0c:29:82:9b:a7",
fcoeConfiguration = (vim.host.FcoeConfig) null,
vmDirectPathGen2Supported = false,
vmDirectPathGen2SupportedMode = <unset>,
resourcePoolSchedulerAllowed = true,
autoNegotiateSupported = true,
},
(vim.host.PhysicalNic) {
  dynamicType = <unset>,
  key = "key-vim.host.PhysicalNic-vmnic3",
  device = "vmnic3",
  pci = "02:04.0",
  driver = "e1000",
  linkSpeed = (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 1000,
    duplex = true,
  },
  validLinkSpecification = (vim.host.PhysicalNic.LinkSpeedDuplex) [
    (vim.host.PhysicalNic.LinkSpeedDuplex) {
      dynamicType = <unset>,
      speedMb = 10,
      duplex = false,
    },
    (vim.host.PhysicalNic.LinkSpeedDuplex) {
      dynamicType = <unset>,
      speedMb = 10,
    }
  ]
}

```

```

        duplex = true,
    },
    (vim.host.PhysicalNic.LinkSpeedDuplex) {
        dynamicType = <unset>,
        speedMb = 100,
        duplex = false,
    },
    (vim.host.PhysicalNic.LinkSpeedDuplex) {
        dynamicType = <unset>,
        speedMb = 100,
        duplex = true,
    },
    (vim.host.PhysicalNic.LinkSpeedDuplex) {
        dynamicType = <unset>,
        speedMb = 1000,
        duplex = true,
    }
},
spec = (vim.host.PhysicalNic.Specification) {
    dynamicType = <unset>,
    ip = (vim.host.IpConfig) {
        dynamicType = <unset>,
        dhcp = false,
        ipAddress = "",
        subnetMask = "",
        ipv6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
    },
    linkSpeed = (vim.host.PhysicalNic.LinkSpeedDuplex) {
        dynamicType = <unset>,
        speedMb = 1000,
        duplex = true,
    },
},
wakeOnLanSupported = false,
mac = "00:0c:29:82:9b:93",
fcoeConfiguration = (vim.host.FcoeConfig) null,
vmDirectPathGen2Supported = false,
vmDirectPathGen2SupportedMode = <unset>,
resourcePoolSchedulerAllowed = true,
autoNegotiateSupported = true,
}
]

```

*** [hostsvc/net/dvs_info](#)**

hostsvc/net/vswitch_remove

Virtual Switch: TS [Remove...](#) [Properties...](#)



Virtual Switch: TS2 [Remove...](#) [Properties...](#)



~ # vim-cmd hostsvc/net/vswitch_remove TS2

~ #

Virtual Switch: TS [Remove...](#) [Properties...](#)



hostsvc/net/info

~ # vim-cmd /hostsvc/net/info

```
(vim.host.NetworkInfo) {
  dynamicType = <unset>,
  vswitch = (vim.host.VirtualSwitch) [
    (vim.host.VirtualSwitch) {
      dynamicType = <unset>,
      name = "vSS-Management",
      key = "key-vim.host.VirtualSwitch-vSS-Management",
      numPorts = 128,
      numPortsAvailable = 125,
      mtu = 1500,
      portgroup = (vim.host.PortGroup) [
        <vim.host.PortGroup:key-vim.host.PortGroup-VM-0>,
        <vim.host.PortGroup:key-vim.host.PortGroup-Management>
      ],
      pnic = (vim.host.PhysicalNic) [
        <vim.host.PhysicalNic:key-vim.host.PhysicalNic-vmnic0>
      ],
      spec = (vim.host.VirtualSwitch.Specification) {
        dynamicType = <unset>,
        numPorts = 128,
        bridge = (vim.host.VirtualSwitch.BondBridge) {
          dynamicType = <unset>,
          nicDevice = (string) [
            "vmnic0"
          ],
        },
      },
    },
  ],
}
```

```

beacon = (vim.host.VirtualSwitch.BeaconConfig) {
  dynamicType = <unset>,
  interval = 1,
},
linkDiscoveryProtocolConfig = (vim.host.LinkDiscoveryProtocolConfig) {
  dynamicType = <unset>,
  protocol = "cdp",
  operation = "both",
},
},
policy = (vim.host.NetworkPolicy) {
  dynamicType = <unset>,
  security = (vim.host.NetworkPolicy.SecurityPolicy) {
    dynamicType = <unset>,
    allowPromiscuous = false,
    macChanges = true,
    forgedTransmits = true,
  },
  nicTeaming = (vim.host.NetworkPolicy.NicTeamingPolicy) {
    dynamicType = <unset>,

```

...

* `hostsvc/net/vswitch_setautobridge`

* `hostsvc/net/offload_capabilities`

`hostsvc/net/vswitch_setbondbridge`

`--vsbridge-bond-pnics=physical_nic_list`

List of physical NIC to which bond bridge is configured.

`--vsbridge-bond-beacon-interval=int`

Probe interval for beacon. Zero to disable.

~ # `vim-cmd /hostsvc/net/vswitch_setbondbridge --vsbridge-bond-pnics=vmnic5 TS`

~ #



`hostsvc/net/pnic_info`

~ # `vim-cmd hostsvc/net/pnic_info | more`

```

(vim.host.PhysicalNic) [
  (vim.host.PhysicalNic) {
    dynamicType = <unset>,
    key = "key-vim.host.PhysicalNic-vmnic0",
    device = "vmnic0",
    pci = "02:01.0",
    driver = "e1000",

```

```

linkSpeed = (vim.host.PhysicalNic.LinkSpeedDuplex) {
  dynamicType = <unset>,
  speedMb = 1000,
  duplex = true,
},
validLinkSpecification = (vim.host.PhysicalNic.LinkSpeedDuplex) [
  (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 10,
    duplex = false,
  },
  (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 10,
    duplex = true,
  },
  (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 100,
    duplex = false,
  },
  (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 100,
    duplex = true,
  },
  (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 1000,
    duplex = true,
  }
],
spec = (vim.host.PhysicalNic.Specification) {
  dynamicType = <unset>,
  ip = (vim.host.IpConfig) {
    dynamicType = <unset>,
    dhcp = false,
    ipAddress = "",
    subnetMask = "",
    ipv6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
  },
  linkSpeed = (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 1000,
    duplex = true,
  },
},
wakeOnLanSupported = false,
mac = "00:0c:29:82:9b:75",
fcoeConfiguration = (vim.host.FcoeConfig) null,
vmDirectPathGen2Supported = false,
vmDirectPathGen2SupportedMode = <unset>,
resourcePoolSchedulerAllowed = true,
autoNegotiateSupported = true,

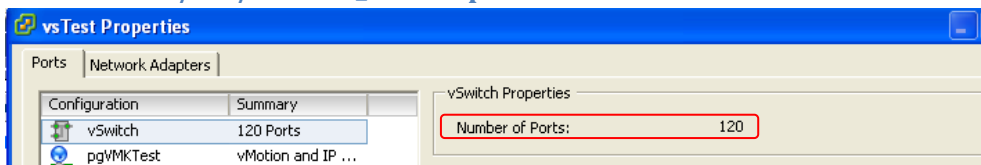
```

```

},
(vim.host.PhysicalNic) {
  dynamicType = <unset>,
  key = "key-vim.host.PhysicalNic-vmnic1",
  device = "vmnic1",
  pci = "02:02.0",
  driver = "e1000",
  linkSpeed = (vim.host.PhysicalNic.LinkSpeedDuplex) {
    dynamicType = <unset>,
    speedMb = 1000,
    duplex = true,
  },
},
...

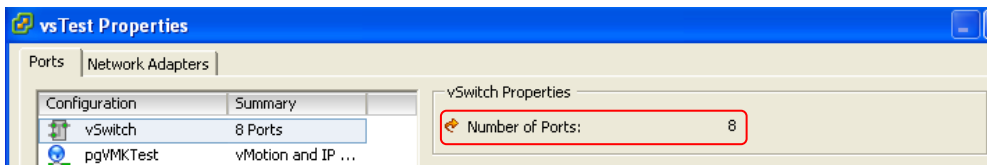
```

hostsvc/net/vswitch_setnumports



```
~ # vim-cmd hostsvc/net/vswitch_setnumports vsTest 16
```

```
~ #
```



* hostsvc/net/pnic_setip

hostsvc/net/vswitch_setpolicy

See [hostsvc/net/portgroup_set](#) – same options but on vSwitch level

hostsvc/net/pnic_setlinkspeed

```
~ # vim-cmd hostsvc/net/pnic_setlinkspeed --linkspeed-mb=100 --linkspeed-duplex=false vmnic5
```

```
~ #
```







Network Adapters

Device	Speed	Configured
Intel Corporation PRO/1000 MT Single Port Adapter		
vmnic5	100 Half	100 Half
vmnic4	1000 Full	1000 Full
vmnic3	1000 Full	1000 Full
vmnic2	1000 Full	1000 Full
vmnic1	1000 Full	1000 Full
vmnic0	1000 Full	1000 Full

```
~ # vim-cmd hostsvc/net/pnic_setlinkspeed vmnic5
```

```
~ #
```

Network Adapters

Device	Speed	Configured
Intel Corporation PRO/1000 MT Single Port Adapter		
 vmnic5	1000 Full	Negotiate
 vmnic4	1000 Full	1000 Full
 vmnic3	1000 Full	1000 Full
 vmnic2	1000 Full	1000 Full
 vmnic1	1000 Full	1000 Full
 vmnic0	1000 Full	1000 Full

```
* hostsvc/net/vswitch_setsimplebridge
```

```
hostsvc/net/portgroup_add
```

```
~ # vim-cmd /hostsvc/net/portgroup_add TS TSpg
```

```
~ #
```



```
hostsvc/net/vswitch_unsetbridge
```

```
~ # vim-cmd /hostsvc/net/vswitch_unsetbridge TS
```

```
~ #
```



hostsvc/hosthardware

```
~ # vim-cmd /hostsvc/hosthardware
```

```
(vim.host.HardwareInfo) {
  dynamicType = <unset>,
  systemInfo = (vim.host.SystemInfo) {
    dynamicType = <unset>,
    vendor = "VMware, Inc.",
    model = "VMware Virtual Platform",
    uuid = "564d985c-68ac-a9b8-0486-1d8d95829b75",
    otherIdentifyingInfo = (vim.host.SystemIdentificationInfo) [
```

```

(vim.host.SystemIdentificationInfo) {
  dynamicType = <unset>,
  identifierValue = "No Asset Tag",
  identifierType = (vim.ElementDescription) {
    dynamicType = <unset>,
    label = "AssetTag",
    summary = "AssetTag",
    key = "AssetTag",
  },
}
],
},
cpuPowerManagementInfo = (vim.host.CpuPowerManagementInfo) {
  dynamicType = <unset>,
  currentPolicy = "Not supported",
  hardwareSupport = "Not available",
},
cpuInfo = (vim.host.CpuInfo) {
  dynamicType = <unset>,
  numCpuPackages = 1,
  numCpuCores = 1,
  numCpuThreads = 1,
  hz = 2186185980,
},
cpuPkg = (vim.host.CpuPackage) [
  (vim.host.CpuPackage) {
    dynamicType = <unset>,
    index = 0,
    vendor = "intel",
    hz = 2186185980,
    busHz = 66248060,
    description = "Intel(R) Core(TM)2 Duo CPU  T7500 @ 2.20GHz",
    threadId = (short) [
      0
    ],
  ],
...

```

* [hostsvc/set_hostid](#)

hostsvc/rsrc/

~ # cat /etc/vmware/hostd/pools.xml

```
<ConfigRoot>
<resourcePool id="0000">
  <name>Resources</name>
  <objID>ha-root-pool</objID>
  <path>host/user</path>
</resourcePool>
<resourcePool id="0001">
  <lastModified>2010-08-13T22:24:14.909074Z</lastModified>
  <name>r1</name>
  <objID>pool1</objID>
  <path>host/user/pool1</path>
</resourcePool>
<vm id="0002">
  <lastModified>2010-08-11T11:37:37.976104Z</lastModified>
  <objID>176</objID>
  <resourcePool>ha-root-pool</resourcePool>
</vm>
<vm id="0003">
  <lastModified>2010-08-12T06:17:27.866378Z</lastModified>
  <objID>192</objID>
  <resourcePool>ha-root-pool</resourcePool>
</vm>
<vm id="0004">
  <lastModified>2010-08-13T10:44:09.276609Z</lastModified>
  <objID>208</objID>
  <resourcePool>ha-root-pool</resourcePool>
</vm>
<vm id="0005">
  <lastModified>2010-08-13T13:55:18.115098Z</lastModified>
  <objID>224</objID>
  <resourcePool>ha-root-pool</resourcePool>
</vm>
<vm id="0006">
  <lastModified>2010-08-13T15:40:09.165016Z</lastModified>
  <objID>240</objID>
  <resourcePool>ha-root-pool</resourcePool>
</vm>
</ConfigRoot>
```

hostsvc/rsrc/create

--cpu-min=int

CPU minimum (in MHz).

--cpu-min-expandable=bool

Can pool borrow CPU resources from parent?

--cpu-max=int

CPU maximum (in MHz).

`--cpu-shares=string`

CPU shares (low/normal/high/<custom>).

`--mem-min=int`

Memory minimum (in MB).

`--mem-min-expandable=bool`

Can pool borrow memory resources from parent?

`--mem-max=int`

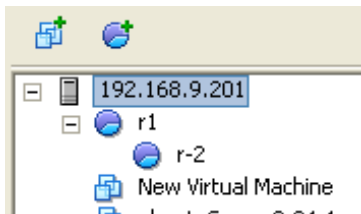
Memory maximum (in MB).

`--mem-shares=string`

Memory shares (low/normal/high/<custom>).

~ # vim-cmd /hostsvc/rsrc/create pool1 r-2

'vim.ResourcePool:pool2'



~ # vim-cmd /hostsvc/rsrc/create --cpu-min=1000 --cpu-min-expandable=true --cpu-max=2000 --cpu-shares=low --mem-min=1024 --mem-min-expandable=false --mem-max=2048 --mem-shares=500 pool1 r-3

```
(vim.fault.InsufficientMemoryResourcesFault) {
  dynamicType = <unset>,
  faultCause = (vmodl.MethodFault) null,
  faultMessage = (vmodl.LocalizableMessage) [
    (vmodl.LocalizableMessage) {
      dynamicType = <unset>,
      key = "com.vmware.esx.hostctl.default",
      arg = (vmodl.KeyAnyValue) [
        (vmodl.KeyAnyValue) {
          dynamicType = <unset>,
          key = "reason",
          value = "Sysinfo error on operation returned status : Admission check failed for memory resource. Please see the VMkernel log for detailed error information",
        }
      ],
    }
  ],
  message = "Operation failed, diagnostics report: Sysinfo error on operation returned status : Admission check failed for memory resource. Please see the VMkernel log for detailed error information",
}
```

```

},
unreserved = -1,
requested = -1,
msg = "The available Memory resources in the parent resource pool are insufficient for the operation."
}

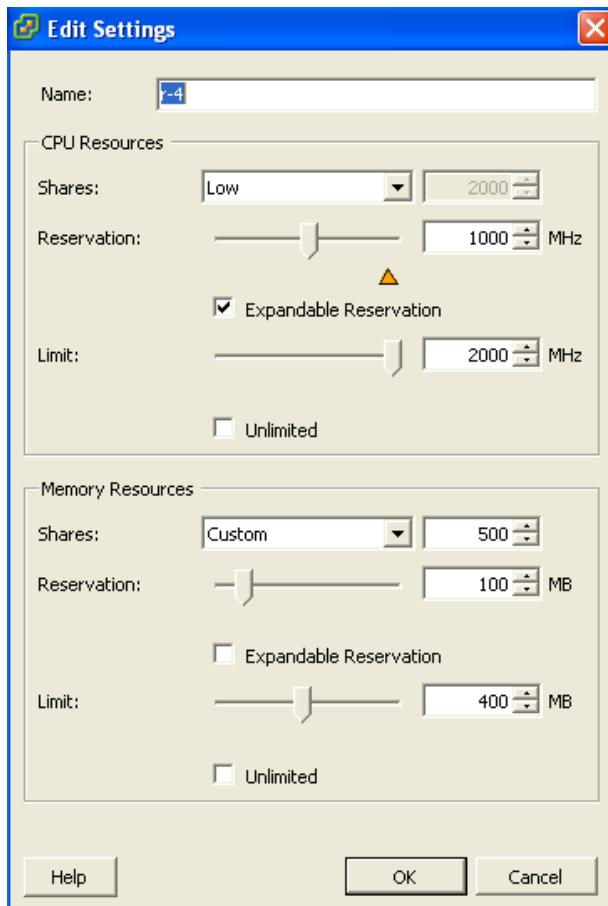
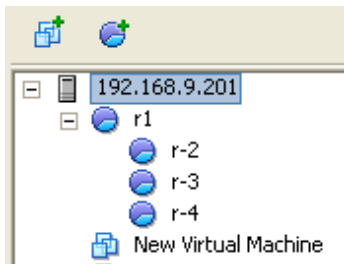
```

```

~ # vim-cmd /hostsvc/rsrc/create --cpu-min=1000 --cpu-min-expandable=true --cpu-max=2000 --cpu-
shares=low --mem-min=100 --mem-min=expandable=false --mem-max=400 --mem-shares=500 pool1
r-4

```

'vim.ResourcePool:pool4'



hostsvc/rsrc/pool_config_get

```

~ # vim-cmd /hostsvc/rsrc/pool_config_get ha-root-pool

```

```

(vim.ResourceConfigSpec) {
dynamicType = <unset>,
entity = 'vim.ResourcePool:ha-root-pool',
changeVersion = <unset>,

```

```

lastModified = <unset>,
cpuAllocation = (vim.ResourceAllocationInfo) {
  dynamicType = <unset>,
  reservation = 1996,
  expandableReservation = false,
  limit = 1996,
  shares = (vim.SharesInfo) {
    dynamicType = <unset>,
    shares = 9000,
    level = "custom",
  },
  overheadLimit = <unset>,
},
memoryAllocation = (vim.ResourceAllocationInfo) {
  dynamicType = <unset>,
  reservation = 845,
  expandableReservation = false,
  limit = 845,
  shares = (vim.SharesInfo) {
    dynamicType = <unset>,
    shares = 9000,
    level = "custom",
  },
  overheadLimit = <unset>,
},
}

```

hostsvc/rsrc/pools_get

```
~ # vim-cmd /hostsvc/rsrc/pools_get ha-root-pool
```

```

(ManagedObjectReference) [
  'vim.ResourcePool:pool1'
]

```

* hostsvc/rsrc/vm_config_set

--cpu-min=int

CPU minimum (in MHz).

--cpu-min-expandable=bool

Can pool borrow CPU resources from parent?

--cpu-max=int

CPU maximum (in MHz).

--cpu-shares=string

CPU shares (low/normal/high/<custom>).

--mem-min=int

Memory minimum (in MB).

`--mem-min-expandable=bool`

Can pool borrow memory resources from parent?

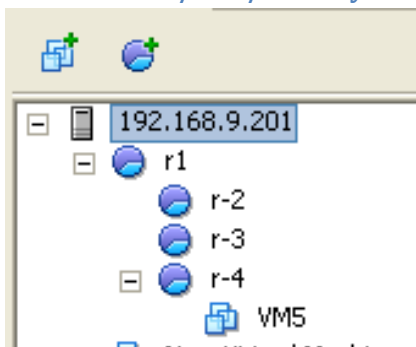
`--mem-max=int`

Memory maximum (in MB).

`--mem-shares=string`

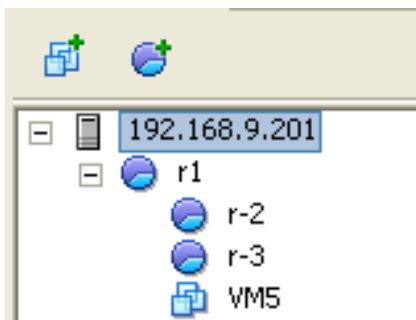
Memory shares (low/normal/high/<custom>).

hostsvc/rsrc/destroy



```
~ # vim-cmd /hostsvc/rsrc/destroy pool4
```

```
~ #
```



hostsvc/rsrc/pool_config_set

```
~ # vim-cmd /hostsvc/rsrc/pool_config_get pool2
```

```
(vim.ResourceConfigSpec) {
  dynamicType = <unset>,
  entity = 'vim.ResourcePool:pool2',
  changeVersion = "2010-08-13T22:32:12.175054Z",
  lastModified = "2010-08-13T22:32:12.175054Z",
  cpuAllocation = (vim.ResourceAllocationInfo) {
    dynamicType = <unset>,
    reservation = 0,
    expandableReservation = false,
```

```

    limit = -1,
    shares = (vim.SharesInfo) {
      dynamicType = <unset>,
      shares = 0,
      level = "custom",
    },
    overheadLimit = <unset>,
  },
  memoryAllocation = (vim.ResourceAllocationInfo) {
    dynamicType = <unset>,
    reservation = 0,
    expandableReservation = false,
    limit = -1,
    shares = (vim.SharesInfo) {
      dynamicType = <unset>,
      shares = 0,
      level = "custom",
    },
    overheadLimit = <unset>,
  },
}

```

```
~ # vim-cmd /hostsvc/rsrc/pool_config_set --cpu-min 999 pool2
```

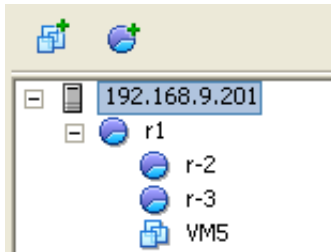
```
~ # vim-cmd /hostsvc/rsrc/pool_config_get pool2
```

```

(vim.ResourceConfigSpec) {
  dynamicType = <unset>,
  entity = 'vim.ResourcePool:pool2',
  changeVersion = "2010-08-13T23:02:21.874847Z",
  lastModified = "2010-08-13T23:02:21.874847Z",
  cpuAllocation = (vim.ResourceAllocationInfo) {
    dynamicType = <unset>,
    reservation = 999,
    expandableReservation = false,
    limit = -1,
    shares = (vim.SharesInfo) {
      dynamicType = <unset>,
      shares = 0,
      level = "custom",
    },
    overheadLimit = <unset>,
  },
  memoryAllocation = (vim.ResourceAllocationInfo) {
    dynamicType = <unset>,
    reservation = 0,
    expandableReservation = false,
    limit = -1,
    shares = (vim.SharesInfo) {
      dynamicType = <unset>,
      shares = 0,
      level = "custom",
    },
    overheadLimit = <unset>,
  },
}

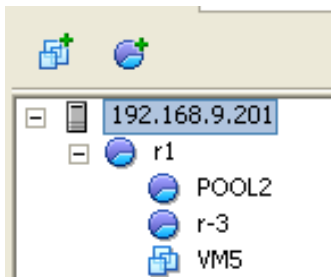
```

hostsvc/rsrc/rename



```
~ # vim-cmd /hostsvc/rsrc/rename pool2 POOL2
```

```
~ #
```

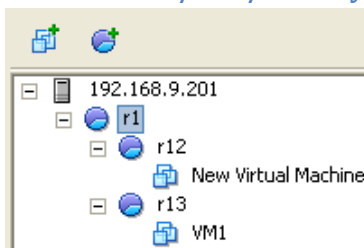


hostsvc/rsrc/vms_get

```
~ # vim-cmd /hostsvc/rsrc/vms_get pool1
```

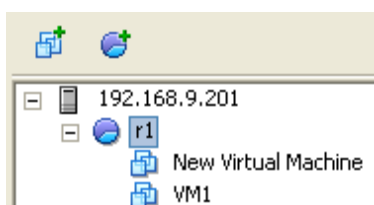
```
(ManagedObjectReference) [
  'vim.VirtualMachine:192'
]
```

hostsvc/rsrc/destroy_children



```
~ # vim-cmd /hostsvc/rsrc/destroy_children pool1
```

```
~ #
```



hostsvc/rsrc/pool_runtime_get

~ # vim-cmd /hostsvc/rsrc/pool_runtime_get pool1

```
(vim.ResourcePool.RuntimeInfo) {
  dynamicType = <unset>,
  memory = (vim.ResourcePool.ResourceUsage) {
    dynamicType = <unset>,
    reservationUsed = 0,
    reservationUsedForVm = 0,
    unreservedForPool = 875560960,
    unreservedForVm = 875560960,
    overallUsage = -2048,
    maxUsage = 875560960,
  },
  cpu = (vim.ResourcePool.ResourceUsage) {
    dynamicType = <unset>,
    reservationUsed = 999,
    reservationUsedForVm = 0,
    unreservedForPool = 997,
    unreservedForVm = 997,
    overallUsage = 0,
    maxUsage = 1996,
  },
  overallStatus = "green",
}
```

* hostsvc/rsrc/vm_config_get

hostsvc/hostsummary

~ # vim-cmd /hostsvc/hostsummary

```
(vim.host.Summary) {
  dynamicType = <unset>,
  host = 'vim.HostSystem:ha-host',
  hardware = (vim.host.Summary.HardwareSummary) {
    dynamicType = <unset>,
    vendor = "VMware, Inc.",
    model = "VMware Virtual Platform",
    uuid = "564d985c-68ac-a9b8-0486-1d8d95829b75",
    otherIdentifyingInfo = (vim.host.SystemIdentificationInfo) [
      (vim.host.SystemIdentificationInfo) {
        dynamicType = <unset>,
        identifierValue = "No Asset Tag",
        identifierType = (vim.ElementDescription) {
          dynamicType = <unset>,
          label = "AssetTag",
          summary = "AssetTag",
          key = "AssetTag",
        },
      },
    ]
  },
  memorySize = 2356490240,
```

```

cpuModel = "Intel(R) Core(TM)2 Duo CPU   T7500 @ 2.20GHz",
cpuMhz = 2186,
numCpuPkgs = 1,
numCpuCores = 1,
numCpuThreads = 1,
numNics = 6,
numHBAs = 4,
},
runtime = (vim.host.RuntimeInfo) {
  dynamicType = <unset>,
  connectionState = "connected",
  powerState = "poweredOn",
  standbyMode = <unset>,
  inMaintenanceMode = false,
  bootTime = "2010-08-13T21:49:20.859635Z",
  healthSystemRuntime = (vim.host.HealthStatusSystem.Runtime) {
    dynamicType = <unset>,
    systemHealthInfo = (vim.host.SystemHealthInfo) {
      dynamicType = <unset>,
    }
  }
}
...

```

hostsvc/standby_mode_enter

```
~ # vim-cmd /hostsvc/standby_mode_enter
```

Entering Standby Mode...

```
'vim.Task:haTask-ha-host-vim.HostSystem.enterStandbyMode-1059'
```

hostsvc/storage/

hostsvc/storage/diagnostic/

hostsvc/storage/diagnostic/activate

```
~ # vim-cmd hostsvc/storage/diagnostic/activate mpx.vmhba1:C0:T0:L0 7
```

```
~ # vim-cmd hostsvc/storage/diagnostic/get_active
```

```

(vim.host.DiagnosticPartition) {
  dynamicType = <unset>,
  storageType = "directAttached",
  diagnosticType = "singleHost",
  slots = 1,
  id = (vim.host.ScsiDisk.Partition) {
    dynamicType = <unset>,
    diskName = "mpx.vmhba1:C0:T0:L0",
    partition = 7,
  },
}

```

hostsvc/storage/diagnostic/get_active

```
~ # vim-cmd hostsvc/storage/diagnostic/get_active
```

```

(vim.host.DiagnosticPartition) {
  dynamicType = <unset>,

```

```

storageType = "directAttached",
diagnosticType = "singleHost",
slots = 1,
id = (vim.host.ScsiDisk.Partition) {
  dynamicType = <unset>,
  diskName = "mpx.vmhba1:C0:T0:L0",
  partition = 7,
},
}

```

hostsvc/storage/diagnostic/query_create_option

~ # vim-cmd hostsvc/storage/diagnostic/query_create_option directAttached singleHost

(vim.host.DiagnosticPartition.CreateOption) []

** hostsvc/storage/diagnostic/create*

hostsvc/storage/diagnostic/query_available

~ # vim-cmd hostsvc/storage/diagnostic/query_available

```

(vim.host.DiagnosticPartition) [
  (vim.host.DiagnosticPartition) {
    dynamicType = <unset>,
    storageType = "directAttached",
    diagnosticType = "singleHost",
    slots = 1,
    id = (vim.host.ScsiDisk.Partition) {
      dynamicType = <unset>,
      diskName = "mpx.vmhba1:C0:T0:L0",
      partition = 7,
    },
  }
]

```

hostsvc/storage/diagnostic/deactivate

~ # vim-cmd hostsvc/storage/diagnostic/deactivate

~ # vim-cmd hostsvc/storage/diagnostic/get_active

~ #

** hostsvc/storage/diagnostic/query_create_desc*

*** hostsvc/storage/multipath_lun_setpolicy**

hostsvc/storage/mockup/

- * hostsvc/storage/mockup/clear*
- * hostsvc/storage/mockup/scsidisk_add*
- * hostsvc/storage/mockup/topology_add*
- * hostsvc/storage/mockup/hostbusadapter_add*
- * hostsvc/storage/mockup/scsidisk_remove*
- * hostsvc/storage/mockup/topology_remove*
- * hostsvc/storage/mockup/hostbusadapter_remove*
- * hostsvc/storage/mockup/scsilun_add*
- * hostsvc/storage/mockup/multipath_path_activate*
- * hostsvc/storage/mockup/scsilun_remove*
- * hostsvc/storage/multipath_path_disable**

hostsvc/storage/provider/*hostsvc/storage/provider/device_info*

~ # vim-cmd hostsvc/storage/provider/device_info

```
(vim.host.StorageDeviceInfo) {
  dynamicType = <unset>,
  hostBusAdapter = (vim.host.HostBusAdapter) [
    (vim.host.BlockHba) {
      dynamicType = <unset>,
      key = "key-vim.host.BlockHba-vmhba0",
      device = "vmhba0",
      bus = 0,
      status = "unknown",
      model = "Virtual Machine Chipset",
      driver = "ata_piix",
      pci = "00:07.1",
    },
    (vim.host.ParallelScsiHba) {
      dynamicType = <unset>,
      key = "key-vim.host.ParallelScsiHba-vmhba1",
      device = "vmhba1",
      bus = 0,
      status = "unknown",
      model = "LSI Logic Parallel SCSI Controller",
      driver = "mptspi",
      pci = "00:10.0",
    },
  ],
  (vim.host.BlockHba) {
    dynamicType = <unset>,
```

```

key = "key-vim.host.BlockHba-vmhba32",
device = "vmhba32",
bus = 0,
status = "unknown",
model = "Virtual Machine Chipset",
driver = "ata_piix",
pci = "00:07.1",
},
(vim.host.InternetScsiHba) {
dynamicType = <unset>,
key = "key-vim.host.InternetScsiHba-vmhba33",
device = "vmhba33",
bus = -1,
status = "online",
model = "iSCSI Software Adapter",
driver = "iscsi_vmk",
pci = "UNKNOWN - NULL PCI DEV IN VMKCTL",
isSoftwareBased = true,
discoveryCapabilities = (vim.host.InternetScsiHba.DiscoveryCapabilities) {
dynamicType = <unset>,
...

```

** hostsvc/storage/provider/enable_stack_trace*

hostsvc/storage/provider/fs_info

~ # vim-cmd hostsvc/storage/provider/fs_info

```

(hostd.host.FileSystemVolumeInfo) {
dynamicType = <unset>,
volumeTypes = (string) [
"vmfs",
"nfs"
],
}

```

Logger output:

```

[2010-08-13 23:55:03.668 FFC64E90 info 'Vmfs2UpgradeModelLock'] [Init] undoAttempts(15), timeoutMsec(2000)
[2010-08-13 23:55:03.668 FFC64E90 verbose 'FSVolumeProvider'] FSVolumeVmkImplProvider::Start() called
[2010-08-13 23:55:03.668 FFC64E90 error 'FSVolumeProvider'] FSVolumeVmkImplProvider::Start() called

```

** hostsvc/storage/multipath_path_enable*

** hostsvc/storage/datastore_browser_search*

hostsvc/storage/partition_get

~ # vim-cmd hostsvc/storage/partition_get

/dev/disks/t10.F405E46494C45400C63513452373D264F42433D243A66525

```

(vim.host.DiskPartitionInfo) {
dynamicType = <unset>,
deviceName = "/dev/disks/t10.F405E46494C45400C63513452373D264F42433D243A66525",
spec = (vim.host.DiskPartitionInfo.Specification) {
dynamicType = <unset>,
partitionFormat = <unset>,

```

```

chs = (vim.host.DiskDimensions.Chs) {
  dynamicType = <unset>,
  cylinder = 183,
  head = 255,
  sector = 63,
},
totalSectors = 2949120,
partition = (vim.host.DiskPartitionInfo.Partition) [
  (vim.host.DiskPartitionInfo.Partition) {
    dynamicType = <unset>,
    partition = 1,
    startSector = 128,
    endSector = 2939894,
    type = "vmfs",
    guid = <unset>,
    logical = false,
    attributes = 0,
  }
],
},
layout = (vim.host.DiskPartitionInfo.Layout) {
  dynamicType = <unset>,
  total = (vim.host.DiskDimensions.Lba) {
    dynamicType = <unset>,
    blockSize = 8225280,
    block = 183,
  },
  partition = (vim.host.DiskPartitionInfo.BlockRange) [
    (vim.host.DiskPartitionInfo.BlockRange) {
      dynamicType = <unset>,
      partition = 1,
      type = "vmfs",
      start = (vim.host.DiskDimensions.Lba) {
        dynamicType = <unset>,
        blockSize = 8225280,
        block = 0,
      },
      end = (vim.host.DiskDimensions.Lba) {
        dynamicType = <unset>,
        blockSize = 8225280,
        block = 182,
      },
    }
  ],
},
}

```

hostsvc/storage/fs_info

~ # vim-cmd hostsvc/storage/fs_info

```
(vim.host.FileSystemMountInfo) [
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,
    path = "/vmfs/volumes/4c654f65-75b87896-ef8e-000c29829b75",
    accessMode = "readWrite",
    accessible = true,
  },
  volume = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "of-1",
    capacity = 3489660928,
    blockSizeMb = 1,
    maxBlocks = 262144,
    majorVersion = 3,
    version = "3.46",
    uuid = "4c654f65-75b87896-ef8e-000c29829b75",
    extent = (vim.host.ScsiDisk.Partition) [
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4",
        partition = 1,
      },
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "10.F405E46494C4540074454613D465D24476E605D223655536",
        partition = 1,
      }
    ],
    vmfsUpgradable = false,
    taggedForVmfsUpgrade = <unset>,
    forceMountedInfo = (vim.host.ForceMountedInfo) null,
  },
  vStorageSupport = "vStorageUnknown",
},
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,
    path = "/vmfs/volumes/4c62c8c2-ecb7f710-88ac-000c29829b75",
    accessMode = "readWrite",
    accessible = true,
  },
  volume = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "FUNB",
    capacity = 4026531840,
    blockSizeMb = 1,
  },
}
```

```

maxBlocks = 262144,
majorVersion = 3,
version = "3.46",
uuid = "4c62c8c2-ecb7f710-88ac-000c29829b75",
extent = (vim.host.ScsiDisk.Partition) [
  (vim.host.ScsiDisk.Partition) {
    dynamicType = <unset>,
    diskName = "eui.51942814b35657a6",
    partition = 1,
  }
],
vmfsUpgradable = false,
taggedForVmfsUpgrade = <unset>,
forceMountedInfo = (vim.host.ForceMountedInfo) null,
},
vStorageSupport = "vStorageUnknown",
},
(vim.host.FileSystemMountInfo) {
dynamicType = <unset>,
mountInfo = (vim.host.MountInfo) {
dynamicType = <unset>,
path = "/vmfs/volumes/4c5cc8e0-80fe1968-98ec-000c29829b7f",
accessMode = "readWrite",
accessible = true,
},
volume = (vim.host.VmfsVolume) {
dynamicType = <unset>,
type = "VMFS",
name = "datastore1",
capacity = 4294967296,
blockSizeMb = 1,
maxBlocks = 262144,
majorVersion = 3,
version = "3.46",
uuid = "4c5cc8e0-80fe1968-98ec-000c29829b7f",
extent = (vim.host.ScsiDisk.Partition) [
  (vim.host.ScsiDisk.Partition) {
    dynamicType = <unset>,
    diskName = "mpx.vmhba1:C0:T0:L0",
    partition = 3,
  }
],
vmfsUpgradable = false,
taggedForVmfsUpgrade = <unset>,
forceMountedInfo = (vim.host.ForceMountedInfo) null,
},
vStorageSupport = "vStorageUnknown",
},
(vim.host.FileSystemMountInfo) {
dynamicType = <unset>,
mountInfo = (vim.host.MountInfo) {
dynamicType = <unset>,
path = "/vmfs/volumes/4c656728-065ce6bc-aea3-000c29829b75",
accessMode = "readWrite",

```

```

    accessible = true,
  },
  volume = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "of-3",
    capacity = 1342177280,
    blockSizeMb = 1,
    maxBlocks = 262144,
    majorVersion = 3,
    version = "3.46",
    uuid = "4c656728-065ce6bc-aea3-000c29829b75",
    extent = (vim.host.ScsiDisk.Partition) [
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "t10.F405E46494C454008775C4432566D2772515A6D295D68496",
        partition = 1,
      }
    ],
    vmfsUpgradable = false,
    taggedForVmfsUpgrade = <unset>,
    forceMountedInfo = (vim.host.ForceMountedInfo) null,
  },
  vStorageSupport = "vStorageUnknown",
},
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,
    path = "/vmfs/volumes/4c65661f-1a2e18b5-783e-000c29829b75",
    accessMode = "readWrite",
    accessible = true,
  },
  volume = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "snap-0d853437-of-2",
    capacity = 1342177280,
    blockSizeMb = 1,
    maxBlocks = 262144,
    majorVersion = 3,
    version = "3.46",
    uuid = "4c65661f-1a2e18b5-783e-000c29829b75",
    extent = (vim.host.ScsiDisk.Partition) [
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "t10.F405E46494C45400C63513452373D264F42433D243A66525",
        partition = 1,
      }
    ],
    vmfsUpgradable = false,
    taggedForVmfsUpgrade = <unset>,
    forceMountedInfo = (vim.host.ForceMountedInfo) null,
  },
},

```

```

    vStorageSupport = "vStorageUnknown",
  },
  (vim.host.FileSystemMountInfo) {
    dynamicType = <unset>,
    mountInfo = (vim.host.MountInfo) {
      dynamicType = <unset>,
      path = "/vmfs/volumes/be455b13-e1a6c8e3",
      accessMode = "readOnly",
      accessible = true,
    },
    volume = (vim.host.NasVolume) {
      dynamicType = <unset>,
      type = "NFS",
      name = "ISO",
      capacity = 250056011776,
      remoteHost = "192.168.9.222",
      remotePath = "/media/dwl/public/ISO_Library",
      userName = <unset>,
    },
    vStorageSupport = "vStorageUnsupported",
  },
  (vim.host.FileSystemMountInfo) {
    dynamicType = <unset>,
    mountInfo = (vim.host.MountInfo) {
      dynamicType = <unset>,
      path = "/vmfs/volumes/1c445c7d-fd56af7a",
      accessMode = "readWrite",
      accessible = true,
    },
    volume = (vim.host.NasVolume) {
      dynamicType = <unset>,
      type = "NFS",
      name = "NFS",
      capacity = 250056011776,
      remoteHost = "192.168.9.222",
      remotePath = "/media/dwl/public/NFS",
      userName = <unset>,
    },
    vStorageSupport = "vStorageUnsupported",
  }
]

```

*** [hostsvc/storage/partition_layout_add](#)**

hostsvc/storage/hba_info

~ # vim-cmd hostsvc/storage/hba_info

```
(vim.host.HostBusAdapter) [
(vim.host.BlockHba) {
  dynamicType = <unset>,
  key = "key-vim.host.BlockHba-vmhba0",
  device = "vmhba0",
  bus = 0,
  status = "unknown",
  model = "Virtual Machine Chipset",
  driver = "ata_piix",
  pci = "00:07.1",
},
(vim.host.ParallelScsiHba) {
  dynamicType = <unset>,
  key = "key-vim.host.ParallelScsiHba-vmhba1",
  device = "vmhba1",
  bus = 0,
  status = "unknown",
  model = "LSI Logic Parallel SCSI Controller",
  driver = "mptspi",
  pci = "00:10.0",
},
(vim.host.BlockHba) {
  dynamicType = <unset>,
  key = "key-vim.host.BlockHba-vmhba32",
  device = "vmhba32",
  bus = 0,
  status = "unknown",
  model = "Virtual Machine Chipset",
  driver = "ata_piix",
  pci = "00:07.1",
},
(vim.host.InternetScsiHba) {
  dynamicType = <unset>,
  key = "key-vim.host.InternetScsiHba-vmhba33",
  device = "vmhba33",
  bus = -1,
  status = "online",
  model = "iSCSI Software Adapter",
  driver = "iscsi_vmk",
  pci = "UNKNOWN - NULL PCI DEV IN VMKCTL",
  isSoftwareBased = true,
  discoveryCapabilities = (vim.host.InternetScsiHba.DiscoveryCapabilities) {
    dynamicType = <unset>,
    iSnsDiscoverySettable = false,
    slpDiscoverySettable = false,
  },
  ...
}
```

hostsvc/storage/partition_layout_get




```
~ # vim-cmd hostsvc/storage/partition_layout_get
/dev/disks/t10.F405E46494C45400C63513452373D264F42433D243A66525
```

```
(vim.host.DiskPartitionInfo.Layout) {
  dynamicType = <unset>,
  total = (vim.host.DiskDimensions.Lba) {
    dynamicType = <unset>,
    blockSize = 8225280,
    block = 183,
  },
  partition = (vim.host.DiskPartitionInfo.BlockRange) [
    (vim.host.DiskPartitionInfo.BlockRange) {
      dynamicType = <unset>,
      partition = 1,
      type = "vmfs",
      start = (vim.host.DiskDimensions.Lba) {
        dynamicType = <unset>,
        blockSize = 8225280,
        block = 0,
      },
      end = (vim.host.DiskDimensions.Lba) {
        dynamicType = <unset>,
        blockSize = 8225280,
        block = 182,
      },
    }
  ],
}
```

hostsvc/storage/hba_rescan

```
~ # vim-cmd hostsvc/storage/hba_rescan vmhba33
```

```
~ #
```

Recent Tasks		
Name	Target	Status
 Rescan HBA	 192.168.9.201	 Completed

* hostsvc/storage/partition_layout_set

hostsvc/storage/info

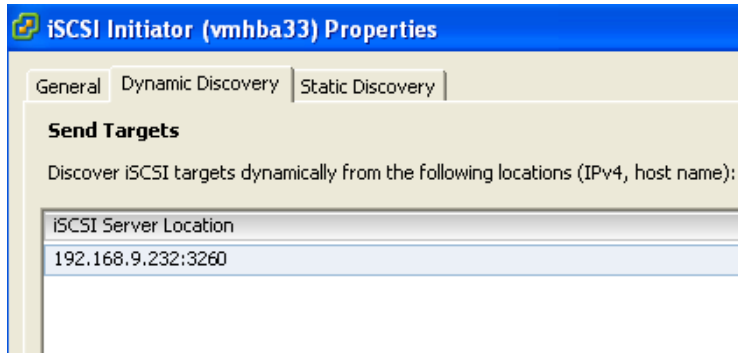
~ # vim-cmd hostsvc/storage/info | less

```
(vim.host.StorageDeviceInfo) {
  dynamicType = <unset>,
  hostBusAdapter = (vim.host.HostBusAdapter) [
    (vim.host.BlockHba) {
      dynamicType = <unset>,
      key = "key-vim.host.BlockHba-vmhba0",
      device = "vmhba0",
      bus = 0,
      status = "unknown",
      model = "Virtual Machine Chipset",
      driver = "ata_piix",
      pci = "00:07.1",
    },
    (vim.host.ParallelScsiHba) {
      dynamicType = <unset>,
      key = "key-vim.host.ParallelScsiHba-vmhba1",
      device = "vmhba1",
      bus = 0,
      status = "unknown",
      model = "LSI Logic Parallel SCSI Controller",
      driver = "mptspi",
      pci = "00:10.0",
    },
    (vim.host.BlockHba) {
      dynamicType = <unset>,
      key = "key-vim.host.BlockHba-vmhba32",
      device = "vmhba32",
      bus = 0,
      status = "unknown",
      model = "Virtual Machine Chipset",
      driver = "ata_piix",
      pci = "00:07.1",
    },
    (vim.host.InternetScsiHba) {
      dynamicType = <unset>,
      key = "key-vim.host.InternetScsiHba-vmhba33",
      device = "vmhba33",
      bus = -1,
      status = "online",
      model = "iSCSI Software Adapter",
      driver = "iscsi_vmk",
      pci = "UNKNOWN - NULL PCI DEV IN VMKCTL",
      isSoftwareBased = true,
      discoveryCapabilities = (vim.host.InternetScsiHba.DiscoveryCapabilities) {
        dynamicType = <unset>,
      }
    }
  ]
}
...

```

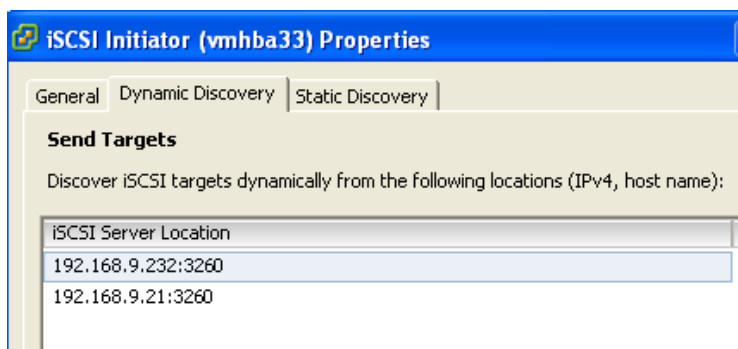
* [hostsvc/storage/partition_remove](#)

hostsvc/storage/iscsi_add_send_target



```
~ # vim-cmd hostsvc/storage/iscsi_add_send_target vmhba33 192.168.9.21
```

```
~ #
```



```
* hostsvc/storage/partition_spec_add
```

```
* hostsvc/storage/iscsi_add_static_target
```

hostsvc/storage/partition_spec_get

```
~ # vim-cmd hostsvc/storage/partition_spec_get /dev/disks/mpx.vmhba1:CO:T0:L0
```

```
(vim.host.DiskPartitionInfo.Specification) {
  dynamicType = <unset>,
  partitionFormat = <unset>,
  chs = (vim.host.DiskDimensions.Chs) {
    dynamicType = <unset>,
    cylinder = 9216,
    head = 64,
    sector = 32,
  },
  totalSectors = 18874368,
  partition = (vim.host.DiskPartitionInfo.Partition) [
    (vim.host.DiskPartitionInfo.Partition) {
      dynamicType = <unset>,
      partition = 1,
      startSector = 8192,
      endSector = 1843199,
      type = "extended",
      guid = <unset>,
      logical = false,
      attributes = 0,
```

```
},  
(vim.host.DiskPartitionInfo.Partition) {  
    dynamicType = <unset>,  
    partition = 2,  
    startSector = 1843200,  
    endSector = 10229759,  
    type = "0x06",  
    guid = <unset>,  
    logical = false,  
    attributes = 0,  
},  
(vim.host.DiskPartitionInfo.Partition) {  
    dynamicType = <unset>,  
    partition = 3,  
    startSector = 10229760,  
    endSector = 18874367,  
    type = "vmfs",  
    guid = <unset>,  
    logical = false,  
    attributes = 0,  
},  
(vim.host.DiskPartitionInfo.Partition) {  
    dynamicType = <unset>,  
    partition = 4,  
    startSector = 32,  
    endSector = 8191,  
    type = "0x04",  
    guid = <unset>,  
    logical = false,  
    attributes = -128,  
},  
(vim.host.DiskPartitionInfo.Partition) {  
    dynamicType = <unset>,  
    partition = 5,  
    startSector = 8224,  
    endSector = 520191,  
    type = "0x06",  
    guid = <unset>,  
    logical = true,  
    attributes = 0,  
},  
(vim.host.DiskPartitionInfo.Partition) {  
    dynamicType = <unset>,  
    partition = 6,  
    startSector = 520224,  
    endSector = 1032191,  
    type = "0x06",  
    guid = <unset>,  
    logical = true,  
    attributes = 0,  
},  
(vim.host.DiskPartitionInfo.Partition) {  
    dynamicType = <unset>,  
    partition = 7,
```

```

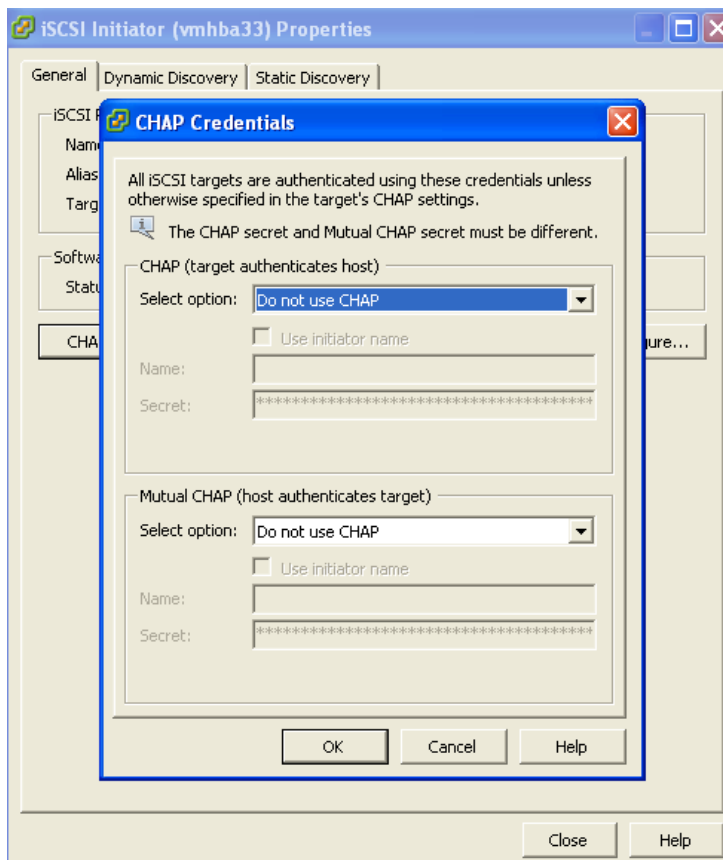
startSector = 1032224,
endSector = 1257471,
type = "vmkDiagnostic",
guid = <unset>,
logical = true,
attributes = 0,
},
(vim.host.DiskPartitionInfo.Partition) {
dynamicType = <unset>,
partition = 8,
startSector = 1257504,
endSector = 1843199,
type = "0x06",
guid = <unset>,
logical = true,
attributes = 0,
}
},
}
}

```

hostsvc/storage/iscsi_disable_chap

```
~ # vim-cmd hostsvc/storage/iscsi_disable_chap vmhba33
```

```
~ #
```



hostsvc/storage/refresh

~ # vim-cmd hostsvc/storage/refresh

~ #

Storage Adapters			Refresh	Rescan All...
Device	Type	WWN		
iSCSI Software Adapter				
vmhba33	iSCSI	iqn.1998-01.com.vmware:esxi-21-30358a96:qwer...		
Virtual Machine Chipset				
vmhba0	Block SCSI			
vmhba32	Block SCSI			
LSI Logic Parallel SCSI Controller				
vmhba1	SCSI			

* hostsvc/storage/iscsi_discovery

hostsvc/storage/scsilun_info

~ # vim-cmd hostsvc/storage/scsilun_info | less

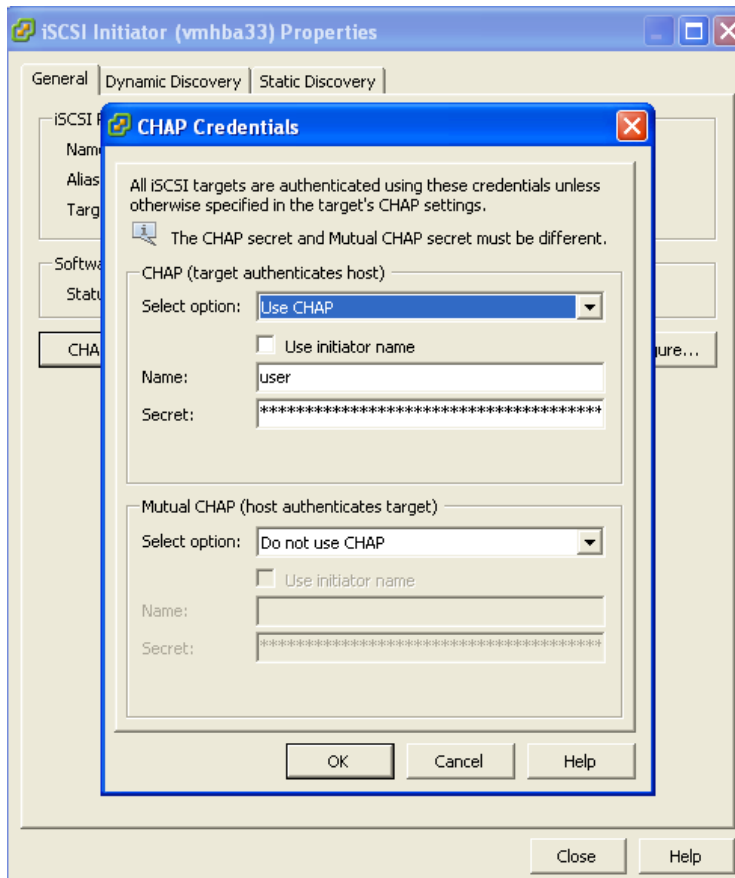
```
(vim.host.Scsilun) [
(vim.host.Scsilun) {
  dynamicType = <unset>,
  deviceName = "/vmfs/devices/cdrom/mpx.vmhba32:C0:T0:L0",
  deviceType = "cdrom",
  key = "key-vim.host.Scsilun-0005000000766d68626133323a303a30",
  uuid = "0005000000766d68626133323a303a30",
  descriptor = (vim.host.Scsilun.Descriptor) [
    (vim.host.Scsilun.Descriptor) {
      dynamicType = <unset>,
      quality = "lowQuality",
      id = "mpx.vmhba32:C0:T0:L0",
    },
    (vim.host.Scsilun.Descriptor) {
      dynamicType = <unset>,
      quality = "lowQuality",
      id = "vml.0005000000766d68626133323a303a30",
    },
    (vim.host.Scsilun.Descriptor) {
      dynamicType = <unset>,
      quality = "lowQuality",
      id = "0005000000766d68626133323a303a30",
    }
  ],
  canonicalName = "mpx.vmhba32:C0:T0:L0",
  displayName = "Local NECVMWar CD-ROM (mpx.vmhba32:C0:T0:L0)",
  lunType = "cdrom",
  vendor = "NECVMWar",
  model = "VMware IDE CDR10",
  revision = "1.00",
  scsiLevel = 5,
  serialNumber = "unavailable",
  durableName = (vim.host.Scsilun.DurableName) null,
  standardInquiry = (byte) [
...

```

hostsvc/storage/iscsi_enable_chap

~ # vim-cmd hostsvc/storage/iscsi_enable_chap vmhba33 user password

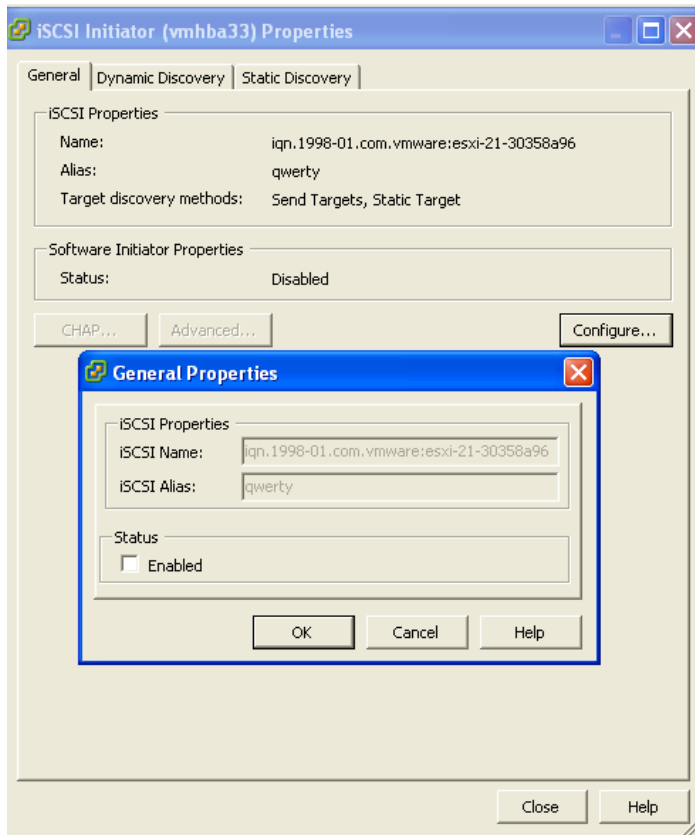
~ #



hostsvc/storage/software_iscsi_enabled

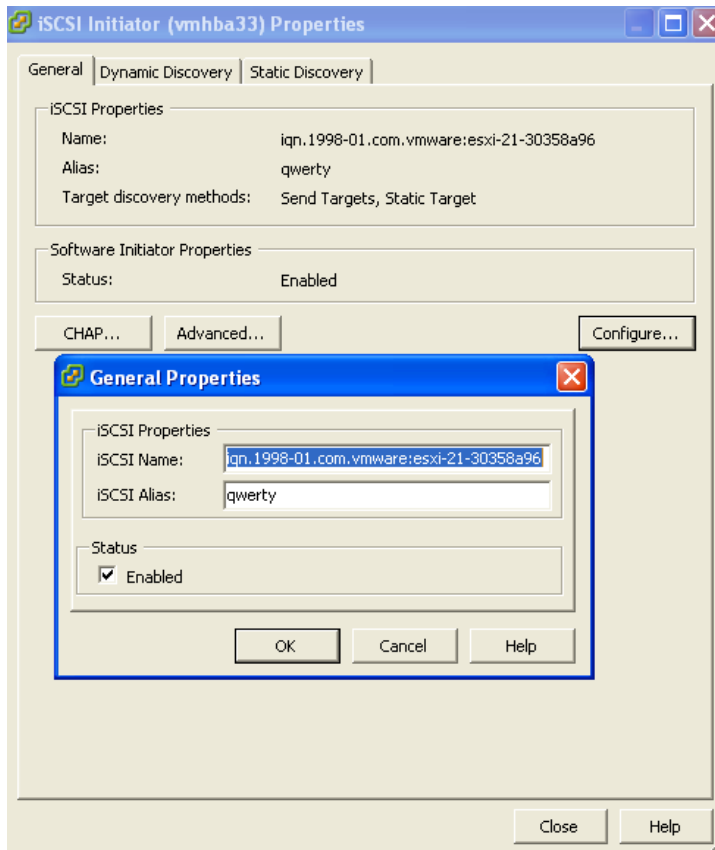
```
~ # vim-cmd hostsvc/storage/software_iscsi_enabled false
```

```
~ #
```



```
~ # vim-cmd hostsvc/storage/software_iscsi_enabled true
```

```
~ #
```



* `hostsvc/storage/iscsi_ip_config`

`hostsvc/storage/topology_info`

~ # `vim-cmd hostsvc/storage/topology_info`

```
(vim.host.ScsiTopology) {
  dynamicType = <unset>,
  adapter = (vim.host.ScsiTopology.Interface) [
    (vim.host.ScsiTopology.Interface) {
      dynamicType = <unset>,
      key = "key-vim.host.ScsiTopology.Interface-vmhba0",
      adapter = <vim.host.BlockHba:key-vim.host.BlockHba-vmhba0>,
    },
    (vim.host.ScsiTopology.Interface) {
      dynamicType = <unset>,
      key = "key-vim.host.ScsiTopology.Interface-vmhba1",
      adapter = <vim.host.ParallelScsiHba:key-vim.host.ParallelScsiHba-vmhba1>,
      target = (vim.host.ScsiTopology.Target) [
        (vim.host.ScsiTopology.Target) {
          dynamicType = <unset>,
          key = "key-vim.host.ScsiTopology.Target-vmhba1:0:0",
          target = 0,
          lun = (vim.host.ScsiTopology.Lun) [
            (vim.host.ScsiTopology.Lun) {
              dynamicType = <unset>,
              key = "key-vim.host.ScsiTopology.Lun-000000000766d686261313a303a30",
              lun = 0,
              scsiLun = <vim.host.ScsiDisk:key-vim.host.ScsiDisk-000000000766d686261313a303a30>,
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ],
  transport = (vim.host.ParallelScsiTargetTransport) {
    dynamicType = <unset>,
  },
}
],
(vim.host.ScsiTopology.Interface) {
  dynamicType = <unset>,
  key = "key-vim.host.ScsiTopology.Interface-vmhba32",
  adapter = <vim.host.BlockHba:key-vim.host.BlockHba-vmhba32>,
  target = (vim.host.ScsiTopology.Target) [
    (vim.host.ScsiTopology.Target) {
      dynamicType = <unset>,
      key = "key-vim.host.ScsiTopology.Target-vmhba32:0:0",
      target = 0,
      lun = (vim.host.ScsiTopology.Lun) [
        (vim.host.ScsiTopology.Lun) {
          dynamicType = <unset>,
          key = "key-vim.host.ScsiTopology.Lun-0005000000766d68626133323a303a30",
          lun = 0,
        }
      ]
    }
  ]
}
...

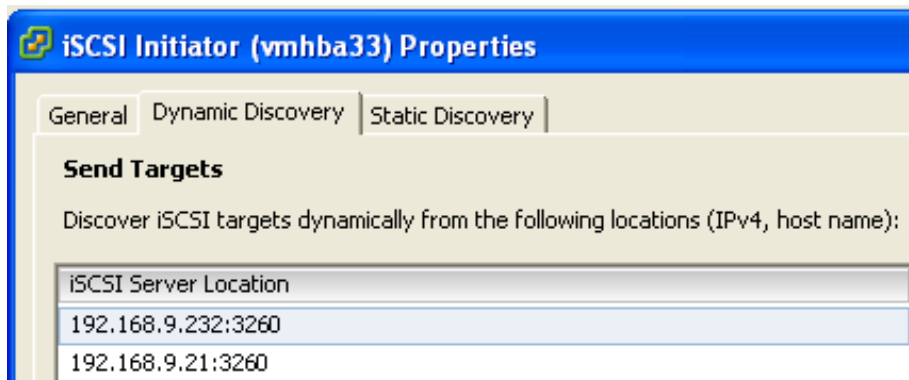
```

* [hostsvc/storage/iscsi_ip_use_dhcp](#)

[hostsvc/storage/upgrade_vm_layout](#)

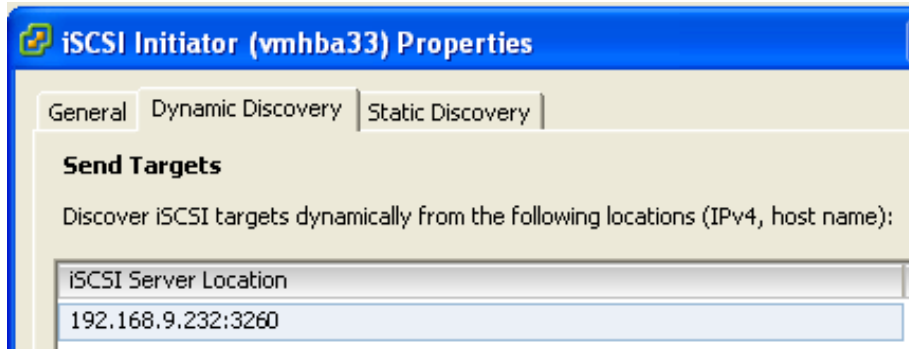
Relayout all legacy Vms from COS to a VMFS volume.

[hostsvc/storage/iscsi_remove_send_target](#)



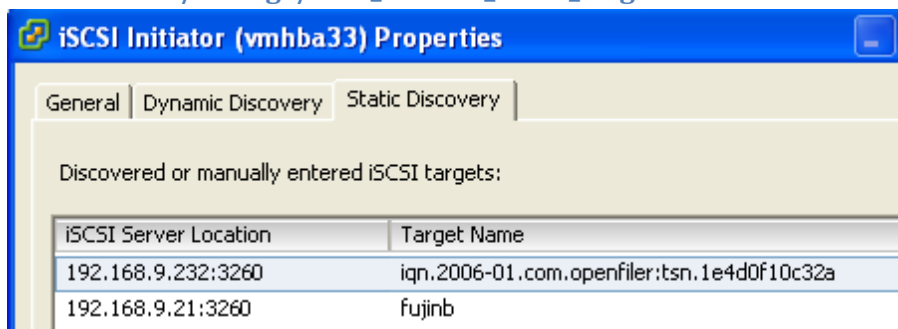
```
~ # vim-cmd hostsvc/storage/iscsi_remove_send_target vmhba33 192.168.9.21
```

```
~ #
```



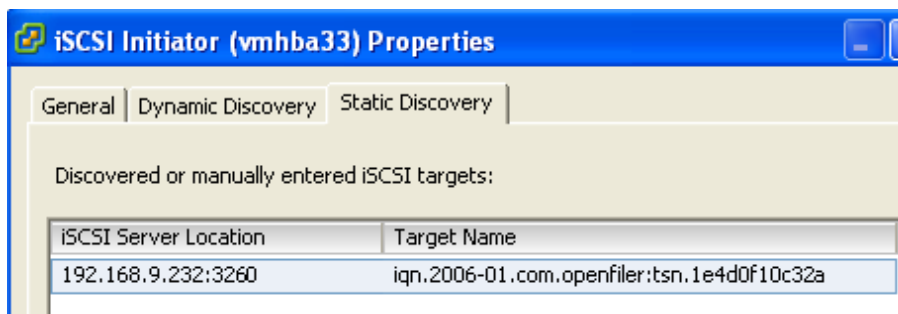
```
* hostsvc/storage/vmfs_attach_extent
```

```
hostsvc/storage/iscsi_remove_static_target
```



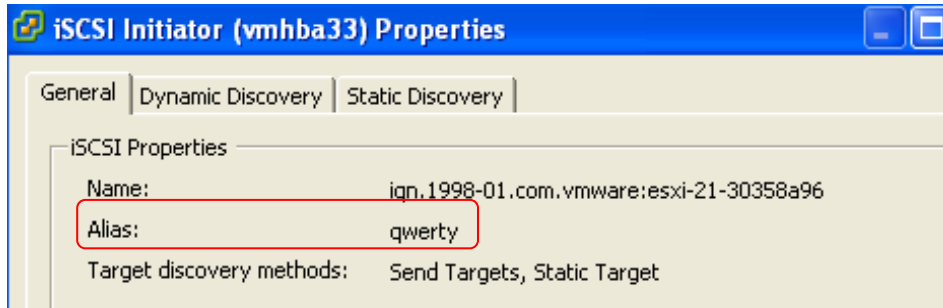
```
~ # vim-cmd hostsvc/storage/iscsi_remove_static_target vmhba33 192.168.9.21 fujinb
```

```
~ #
```



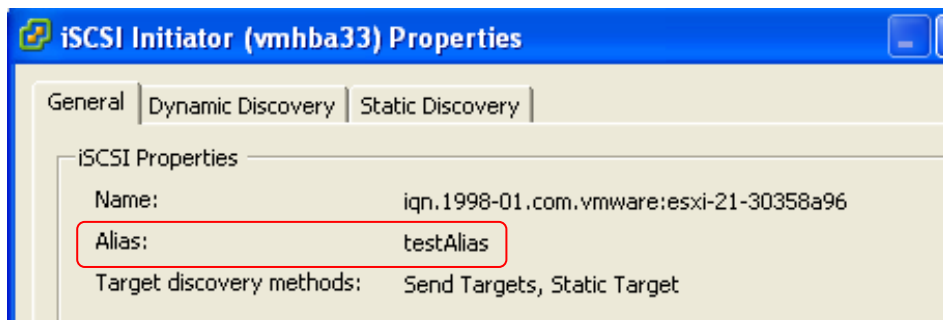
```
* hostsvc/storage/vmfs_format
```

hostsvc/storage/iscsi_set_alias



```
~ # vim-cmd hostsvc/storage/iscsi_set_alias vmhba33 testAlias
```

```
~ #
```



hostsvc/storage/vmfs_rescan

```
~ # vim-cmd hostsvc/storage/vmfs_rescan
```

```
(vim.host.FileSystemMountInfo) [
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,
    path = "/vmfs/volumes/4c5cc8e0-80fe1968-98ec-000c29829b7f",
    accessMode = "readWrite",
    accessible = true,
  },
  volume = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "datastore1",
    capacity = 4294967296,
    blockSizeMb = 1,
    maxBlocks = 262144,
    majorVersion = 3,
    version = "3.46",
    uuid = "4c5cc8e0-80fe1968-98ec-000c29829b7f",
    extent = (vim.host.ScsiDisk.Partition) [
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "mpx.vmhba1:C0:T0:L0",
        partition = 3,
      }
    ]
  },
}
```

```

    vmfsUpgradable = false,
    taggedForVmfsUpgrade = <unset>,
    forceMountedInfo = (vim.host.ForceMountedInfo) null,
  },
  vStorageSupport = "vStorageUnknown",
},
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,
    path = "/vmfs/volumes/4c65661f-1a2e18b5-783e-000c29829b75",
    accessMode = "readWrite",
    accessible = true,
  },
  volume = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "snap-0d853437-of-2",
    capacity = 1342177280,
    blockSizeMb = 1,
    maxBlocks = 262144,
    majorVersion = 3,
    version = "3.46",
    uuid = "4c65661f-1a2e18b5-783e-000c29829b75",
    extent = (vim.host.ScsiDisk.Partition) [
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "t10.F405E46494C45400C63513452373D264F42433D243A66525",
        partition = 1,
      }
    ],
    vmfsUpgradable = false,
    taggedForVmfsUpgrade = <unset>,
    forceMountedInfo = (vim.host.ForceMountedInfo) null,
  },
  vStorageSupport = "vStorageUnknown",
},
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,
    path = "/vmfs/volumes/4c656728-065ce6bc-aea3-000c29829b75",
    accessMode = "readWrite",
    accessible = true,
  },
  volume = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "of-3",
    capacity = 1342177280,
    blockSizeMb = 1,
    maxBlocks = 262144,
    majorVersion = 3,
    version = "3.46",
  }
}

```

```

    uuid = "4c656728-065ce6bc-aea3-000c29829b75",
    extent = (vim.host.ScsiDisk.Partition) [
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "t10.F405E46494C454008775C4432566D2772515A6D295D68496",
        partition = 1,
      }
    ],
    vmfsUpgradable = false,
    taggedForVmfsUpgrade = <unset>,
    forceMountedInfo = (vim.host.ForceMountedInfo) null,
  },
  vStorageSupport = "vStorageUnknown",
},
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,
    path = "/vmfs/volumes/4c654f65-75b87896-ef8e-000c29829b75",
    accessMode = "readWrite",
    accessible = true,
  },
  volume = (vim.host.VmfsVolume) {
    dynamicType = <unset>,
    type = "VMFS",
    name = "of-1",
    capacity = 3489660928,
    blockSizeMb = 1,
    maxBlocks = 262144,
    majorVersion = 3,
    version = "3.46",
    uuid = "4c654f65-75b87896-ef8e-000c29829b75",
    extent = (vim.host.ScsiDisk.Partition) [
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "t10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4",
        partition = 1,
      },
      (vim.host.ScsiDisk.Partition) {
        dynamicType = <unset>,
        diskName = "t10.F405E46494C4540074454613D465D24476E605D223655536",
        partition = 1,
      }
    ],
    vmfsUpgradable = false,
    taggedForVmfsUpgrade = <unset>,
    forceMountedInfo = (vim.host.ForceMountedInfo) null,
  },
  vStorageSupport = "vStorageUnknown",
},
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,




```

```

    path = "/vmfs/volumes/be455b13-e1a6c8e3",
    accessMode = "readOnly",
    accessible = true,
  },
  volume = (vim.host.NasVolume) {
    dynamicType = <unset>,
    type = "NFS",
    name = "ISO",
    capacity = 250056011776,
    remoteHost = "192.168.9.222",
    remotePath = "/media/dwl/public/ISO_Library",
    userName = <unset>,
  },
  vStorageSupport = "vStorageUnsupported",
},
(vim.host.FileSystemMountInfo) {
  dynamicType = <unset>,
  mountInfo = (vim.host.MountInfo) {
    dynamicType = <unset>,
    path = "/vmfs/volumes/1c445c7d-fd56af7a",
    accessMode = "readWrite",
    accessible = true,
  },
  volume = (vim.host.NasVolume) {
    dynamicType = <unset>,
    type = "NFS",
    name = "NFS",
    capacity = 250056011776,
    remoteHost = "192.168.9.222",
    remotePath = "/media/dwl/public/NFS",
    userName = <unset>,
  },
  vStorageSupport = "vStorageUnsupported",
}
}
]

```

Recent Tasks

Name	Target	Status
 Rescan VMFS	 192.168.9.201	 Completed

* [hostsvc/storage/iscsi_set_name](#)

[hostsvc/storage/vmfs_upgrade](#)

Usage: vmfs_upgrade VmfsName

Upgrade a VMFS2 volume to VMFS3.

hostsvc/storage/multipath_info

~ # vim-cmd hostsvc/storage/multipath_info | more

```
(vim.host.MultipathInfo) {
  dynamicType = <unset>,
  lun = (vim.host.MultipathInfo.LogicalUnit) [
    (vim.host.MultipathInfo.LogicalUnit) {
      dynamicType = <unset>,
      key = "key-vim.host.MultipathInfo.LogicalUnit-010004000078574c3452662d7752516a2d596d4869564952545541",
      id = "010004000078574c3452662d7752516a2d596d4869564952545541",
      lun = <vim.host.ScsiDisk:key-vim.host.ScsiDisk-010004000078574c3452662d7752516a2d596d4869564952545541>,
      path = (vim.host.MultipathInfo.Path) [
        (vim.host.MultipathInfo.Path) {
          dynamicType = <unset>,
          key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware:esxi-21-30358a96-00023d000001,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C454008775C4432566D2772515A6D295D68496",
          name = "iqn.1998-01.com.vmware:esxi-21-30358a96-00023d000001,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C454008775C4432566D2772515A6D295D68496",
          pathState = "standby",
          state = "active",
          isWorkingPath = false,
          adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
          lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-010004000078574c3452662d7752516a2d596d4869564952545541>,
          transport = (vim.host.InternetScsiTargetTransport) {
            dynamicType = <unset>,
            iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
            iScsiAlias = "",
            address = (string) [
              "192.168.9.232:3260"
            ],
          },
        },
      ],
    },
  ],
  (vim.host.MultipathInfo.Path) {
    dynamicType = <unset>,
    key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware:esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C454008775C4432566D2772515A6D295D68496",
    name = "iqn.1998-01.com.vmware:esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C454008775C4432566D2772515A6D295D68496",
    pathState = "active",
    state = "active",
    isWorkingPath = true,
    adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
    lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-010004000078574c3452662d7752516a2d596d4869564952545541>,
    transport = (vim.host.InternetScsiTargetTransport) {
      dynamicType = <unset>,
      iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
      iScsiAlias = "",
      address = (string) [
        "192.168.9.232:3260"
      ],
    },
  },
],
}
```

```

policy = (vim.host.MultipathInfo.FixedLogicalUnitPolicy) {
  dynamicType = <unset>,
  policy = "VMW_PSP_FIXED",
  prefer = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-
t10.F405E46494C454008775C4432566D2772515A6D295D68496",
},
storageArrayTypePolicy = (vim.host.MultipathInfo.LogicalUnitStorageArrayTypePolicy) {
  dynamicType = <unset>,
  policy = "VMW_SATP_DEFAULT_AA",
},
},
(vim.host.MultipathInfo.LogicalUnit) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.LogicalUnit-0100010000475464314d562d44676e502d32565563564952545541",
  id = "0100010000475464314d562d44676e502d32565563564952545541",
  lun = <vim.host.ScsiDisk:key-vim.host.ScsiDisk-0100010000475464314d562d44676e502d32565563564952545541>,
  path = (vim.host.MultipathInfo.Path) [
    (vim.host.MultipathInfo.Path) {
      dynamicType = <unset>,
      key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000001,iqn.2006-
01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C4540074454613D465D24476E605D223655536",
      name = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000001,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-
t10.F405E46494C4540074454613D465D24476E605D223655536",
      pathState = "standby",
      state = "active",
      isWorkingPath = false,
      adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
      lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-
0100010000475464314d562d44676e502d32565563564952545541>,
      transport = (vim.host.InternetScsiTargetTransport) {
        dynamicType = <unset>,
        iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
        iScsiAlias = "",
        address = (string) [
          "192.168.9.232:3260"
        ],
      },
    },
  ],
},
(vim.host.MultipathInfo.Path) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-
01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C4540074454613D465D24476E605D223655536",
  name = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-
t10.F405E46494C4540074454613D465D24476E605D223655536",
  pathState = "active",
  state = "active",
  isWorkingPath = true,
  adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
  lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-
0100010000475464314d562d44676e502d32565563564952545541>,
  transport = (vim.host.InternetScsiTargetTransport) {
    dynamicType = <unset>,
    iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
    iScsiAlias = "",
    address = (string) [
      "192.168.9.232:3260"
    ],
  },
},

```

```

    ],
  },
}
],
policy = (vim.host.MultipathInfo.FixedLogicalUnitPolicy) {
  dynamicType = <unset>,
  policy = "VMW_PSP_FIXED",
  prefer = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler.tsn.1e4d0f10c32a,t,1-t10.F405E46494C4540074454613D465D24476E605D223655536",
},
storageArrayTypePolicy = (vim.host.MultipathInfo.LogicalUnitStorageArrayTypePolicy) {
  dynamicType = <unset>,
  policy = "VMW_SATP_DEFAULT_AA",
},
},
(vim.host.MultipathInfo.LogicalUnit) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.LogicalUnit-010002000034626f4c31372d774536692d4d52706d564952545541",
  id = "010002000034626f4c31372d774536692d4d52706d564952545541",
  lun = <vim.host.ScsiDisk:key-vim.host.ScsiDisk-010002000034626f4c31372d774536692d4d52706d564952545541>,
  path = (vim.host.MultipathInfo.Path) [
    (vim.host.MultipathInfo.Path) {
      dynamicType = <unset>,
      key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000001,iqn.2006-01.com.openfiler.tsn.1e4d0f10c32a,t,1-t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
      name = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000001,iqn.2006-01.com.openfiler.tsn.1e4d0f10c32a,t,1-t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
      pathState = "standby",
      state = "active",
      isWorkingPath = false,
      adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
      lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-010002000034626f4c31372d774536692d4d52706d564952545541>,
      transport = (vim.host.InternetScsiTargetTransport) {
        dynamicType = <unset>,
        iScsiName = "iqn.2006-01.com.openfiler.tsn.1e4d0f10c32a",
        iScsiAlias = "",
        address = (string) [
          "192.168.9.232:3260"
        ],
      },
    },
  ],
},
(vim.host.MultipathInfo.Path) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler.tsn.1e4d0f10c32a,t,1-t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
  name = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler.tsn.1e4d0f10c32a,t,1-t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
  pathState = "active",
  state = "active",
  isWorkingPath = true,
  adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
  lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-010002000034626f4c31372d774536692d4d52706d564952545541>,
  transport = (vim.host.InternetScsiTargetTransport) {
    dynamicType = <unset>,

```

```

        iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
        iScsiAlias = "",
        address = (string) [
            "192.168.9.232:3260"
        ],
    },
}
],
policy = (vim.host.MultipathInfo.FixedLogicalUnitPolicy) {
    dynamicType = <unset>,
    policy = "VMW_PSP_FIXED",
    prefer = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-
t10.F405E46494C454004326F6C41373D277546396D2D42507D6",
},
storageArrayTypePolicy = (vim.host.MultipathInfo.LogicalUnitStorageArrayTypePolicy) {
    dynamicType = <unset>,
    policy = "VMW_SATP_DEFAULT_AA",
},
},
(vim.host.MultipathInfo.LogicalUnit) {
    dynamicType = <unset>,
    key = "key-vim.host.MultipathInfo.LogicalUnit-000000000766d686261313a303a30",
    id = "000000000766d686261313a303a30",
    lun = <vim.host.ScsiDisk:key-vim.host.ScsiDisk-000000000766d686261313a303a30>,
    path = (vim.host.MultipathInfo.Path) [
        (vim.host.MultipathInfo.Path) {
            dynamicType = <unset>,
            key = "key-vim.host.MultipathInfo.Path-pscsi.vmhba1-pscsi.0:0-mpx.vmhba1:C0:T0:L0",
            name = "pscsi.vmhba1-pscsi.0:0-mpx.vmhba1:C0:T0:L0",
            pathState = "active",
            state = "active",
            isWorkingPath = true,
            adapter = <vim.host.ParallelScsiHba:key-vim.host.ParallelScsiHba-vmhba1>,
            lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-000000000766d686261313a303a30>,
            transport = (vim.host.ParallelScsiTargetTransport) {
                dynamicType = <unset>,
            },
        },
    ],
},
policy = (vim.host.MultipathInfo.FixedLogicalUnitPolicy) {
    dynamicType = <unset>,
    policy = "VMW_PSP_FIXED",
    prefer = "pscsi.vmhba1-pscsi.0:0-mpx.vmhba1:C0:T0:L0",
},
storageArrayTypePolicy = (vim.host.MultipathInfo.LogicalUnitStorageArrayTypePolicy) {
    dynamicType = <unset>,
    policy = "VMW_SATP_LOCAL",
},
},
(vim.host.MultipathInfo.LogicalUnit) {
    dynamicType = <unset>,
    key = "key-vim.host.MultipathInfo.LogicalUnit-0005000000766d68626133323a303a30",
    id = "0005000000766d68626133323a303a30",
    lun = <vim.host.ScsiLun:key-vim.host.ScsiLun-0005000000766d68626133323a303a30>,

```

```

path = (vim.host.MultipathInfo.Path) [
(vim.host.MultipathInfo.Path) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.Path-ide.vmhba32-ide.0:0-mpx.vmhba32:C0:T0:L0",
  name = "ide.vmhba32-ide.0:0-mpx.vmhba32:C0:T0:L0",
  pathState = "active",
  state = "active",
  isWorkingPath = true,
  adapter = <vim.host.BlockHba:key-vim.host.BlockHba-vmhba32>,
  lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-0005000000766d68626133323a303a30>,
  transport = (vim.host.BlockAdapterTargetTransport) {
    dynamicType = <unset>,
  },
}
],
policy = (vim.host.MultipathInfo.FixedLogicalUnitPolicy) {
  dynamicType = <unset>,
  policy = "VMW_PSP_FIXED",
  prefer = "ide.vmhba32-ide.0:0-mpx.vmhba32:C0:T0:L0",
},
storageArrayTypePolicy = (vim.host.MultipathInfo.LogicalUnitStorageArrayTypePolicy) {
  dynamicType = <unset>,
  policy = "VMW_SATP_LOCAL",
},
},
(vim.host.MultipathInfo.LogicalUnit) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.LogicalUnit-010000000035313934323831344233353635374136494d41474546",
  id = "010000000035313934323831344233353635374136494d41474546",
  lun = <vim.host.ScsiDisk:key-vim.host.ScsiDisk-010000000035313934323831344233353635374136494d41474546>,
  path = (vim.host.MultipathInfo.Path) [
(vim.host.MultipathInfo.Path) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware:esxi-21-30358a96--",
  name = "iqn.1998-01.com.vmware:esxi-21-30358a96--",
  pathState = "dead",
  state = "dead",
  isWorkingPath = false,
  adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
  lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-010000000035313934323831344233353635374136494d41474546>,
  transport = (vim.host.InternetScsiTargetTransport) {
    dynamicType = <unset>,
    iScsiName = "",
    iScsiAlias = "",
  },
}
],
policy = (vim.host.MultipathInfo.FixedLogicalUnitPolicy) {
  dynamicType = <unset>,
  policy = "VMW_PSP_FIXED_AP",
  prefer = "iqn.1998-01.com.vmware:esxi-21-30358a96--",
},
storageArrayTypePolicy = (vim.host.MultipathInfo.LogicalUnitStorageArrayTypePolicy) {

```

```

dynamicType = <unset>,
policy = "VMW_SATP_DEFAULT_AA",
},
},
(vim.host.MultipathInfo.LogicalUnit) {
dynamicType = <unset>,
key = "key-vim.host.MultipathInfo.LogicalUnit-010000000444c47364a5a2d576f51762d43304a4f564952545541",
id = "010000000444c47364a5a2d576f51762d43304a4f564952545541",
lun = <vim.host.ScsiDisk:key-vim.host.ScsiDisk-010000000444c47364a5a2d576f51762d43304a4f564952545541>,
path = (vim.host.MultipathInfo.Path) [
(vim.host.MultipathInfo.Path) {
dynamicType = <unset>,
key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000001,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4",
name = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000001,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4",
pathState = "standby",
state = "active",
isWorkingPath = false,
adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-010000000444c47364a5a2d576f51762d43304a4f564952545541>,
transport = (vim.host.InternetScsiTargetTransport) {
dynamicType = <unset>,
iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
iScsiAlias = "",
address = (string) [
"192.168.9.232:3260"
],
},
},
},
(vim.host.MultipathInfo.Path) {
dynamicType = <unset>,
key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4",
name = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4",
pathState = "active",
state = "active",
isWorkingPath = true,
adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-010000000444c47364a5a2d576f51762d43304a4f564952545541>,
transport = (vim.host.InternetScsiTargetTransport) {
dynamicType = <unset>,
iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
iScsiAlias = "",
address = (string) [
"192.168.9.232:3260"
],
},
},
}
},
policy = (vim.host.MultipathInfo.FixedLogicalUnitPolicy) {
dynamicType = <unset>,
policy = "VMW_PSP_FIXED",
}
}

```

```

    prefer = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-
t10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4",
  },
  storageArrayTypePolicy = (vim.host.MultipathInfo.LogicalUnitStorageArrayTypePolicy) {
    dynamicType = <unset>,
    policy = "VMW_SATP_DEFAULT_AA",
  },
},
(vim.host.MultipathInfo.LogicalUnit) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.LogicalUnit-01000300006c53315432372d464f42332d346a5652564952545541",
  id = "01000300006c53315432372d464f42332d346a5652564952545541",
  lun = <vim.host.ScsiDisk:key-vim.host.ScsiDisk-01000300006c53315432372d464f42332d346a5652564952545541>,
  path = (vim.host.MultipathInfo.Path) [
    (vim.host.MultipathInfo.Path) {
      dynamicType = <unset>,
      key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000001,iqn.2006-
01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C45400C63513452373D264F42433D243A66525",
      name = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000001,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-
t10.F405E46494C45400C63513452373D264F42433D243A66525",
      pathState = "standby",
      state = "active",
      isWorkingPath = false,
      adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
      lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-
01000300006c53315432372d464f42332d346a5652564952545541>,
      transport = (vim.host.InternetScsiTargetTransport) {
        dynamicType = <unset>,
        iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
        iScsiAlias = "",
        address = (string) [
          "192.168.9.232:3260"
        ],
      },
    },
  ],
},
(vim.host.MultipathInfo.Path) {
  dynamicType = <unset>,
  key = "key-vim.host.MultipathInfo.Path-iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-
01.com.openfiler:tsn.1e4d0f10c32a,t,1-t10.F405E46494C45400C63513452373D264F42433D243A66525",
  name = "iqn.1998-01.com.vmware.esxi-21-30358a96-00023d000002,iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a,t,1-
t10.F405E46494C45400C63513452373D264F42433D243A66525",
  pathState = "active",
  state = "active",
  isWorkingPath = true,
  adapter = <vim.host.InternetScsiHba:key-vim.host.InternetScsiHba-vmhba33>,
  lun = <vim.host.MultipathInfo.LogicalUnit:key-vim.host.MultipathInfo.LogicalUnit-
01000300006c53315432372d464f42332d346a5652564952545541>,
  transport = (vim.host.InternetScsiTargetTransport) {
    dynamicType = <unset>,
    iScsiName = "iqn.2006-01.com.openfiler:tsn.1e4d0f10c32a",
    iScsiAlias = "",
    address = (string) [
      "192.168.9.232:3260"
    ],
  },
},
}

```


hostsvc/summary/scsilun

~ # vim-cmd hostsvc/summary/scsilun

Device	Type	Vendor	Model	UUID	Hba
mpx.vmhba1:C0:T0:L0	disk	VMware,	VMware Virtual S	000000000766d686261313a303a30	vmhba1
mpx.vmhba32:C0:T0:L0 vmhba32	cdrom	NECVMWAr	VMware IDE CDR10	0005000000766d68626133323a303a30	
t10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4 0100000000444c47364a5a2d576f51762d43304a4f564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C4540074454613D465D24476E605D223655536 0100010000475464314d562d44676e502d32565563564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C454004326F6C41373D277546396D2D42507D6 010002000034626f4c31372d774536692d4d52706d564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C45400C63513452373D264F42433D243A66525 01000300006c53315432372d464f42332d346a5652564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C454008775C4432566D2772515A6D295D68496 010004000078574c3452662d7752516a2d596d4869564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C4540044C47463A4A5D275F61567D23403A4F4 0100000000444c47364a5a2d576f51762d43304a4f564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C4540074454613D465D24476E605D223655536 0100010000475464314d562d44676e502d32565563564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C454004326F6C41373D277546396D2D42507D6 010002000034626f4c31372d774536692d4d52706d564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C45400C63513452373D264F42433D243A66525 01000300006c53315432372d464f42332d346a5652564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
t10.F405E46494C454008775C4432566D2772515A6D295D68496 010004000078574c3452662d7752516a2d596d4869564952545541	disk	OPNFILER	VIRTUAL-DISK		vmhba33
eui.51942814b35657a6 vmhba33	disk	ROCKET	IMAGEFILE	010000000035313934323831344233353635374136494d41474546	

* hostsvc/logout

hostsvc/start_local_tsm

Troubleshooting Mode Options	Local Tech Support Mode
Enable Local Tech Support Disable Remote Tech Support (SSH) Modify Tech Support timeout Restart Management Agents	Local TSM is Disabled Change current state of Local Tech Support Mode

~ # vim-cmd hostsvc/start_local_tsm

~ #

Troubleshooting Mode Options	Local Tech Support Mode
Disable Local Tech Support Disable Remote Tech Support (SSH) Modify Tech Support timeout Restart Management Agents	Local TSM is Enabled Change current state of Local Tech Support Mode

hostsvc/vmotion/

* **hostsvc/vmotion/ipconfig_get**

* **hostsvc/vmotion/ipconfig_set**

hostsvc/vmotion/netconfig_get

~ # vim-cmd hostsvc/vmotion/netconfig_get

```
(vim.host.VMotionSystem.NetConfig) {
  dynamicType = <unset>,
  candidateVnic = (vim.host.VirtualNic) [
    (vim.host.VirtualNic) {
      dynamicType = <unset>,
      device = "vmk0",
      key = "VMotionConfig.vmotion.key-vim.host.VirtualNic-vmk0",
      portgroup = "Management",
      spec = (vim.host.VirtualNic.Specification) {
        dynamicType = <unset>,
        ip = (vim.host.IpConfig) {
          dynamicType = <unset>,
          dhcp = false,
          ipAddress = "192.168.9.201",
          subnetMask = "255.255.255.0",
          ipV6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
        },
        mac = "00:50:56:71:bd:7a",
        distributedVirtualPort = (vim.dvs.PortConnection) null,
        portgroup = "Management",
        mtu = 1500,
        tsoEnabled = true,
      },
      port = <unset>,
    },
    (vim.host.VirtualNic) {
      dynamicType = <unset>,
      device = "vmk1",
      key = "VMotionConfig.vmotion.key-vim.host.VirtualNic-vmk1",
      portgroup = "VMk-SWiSCSI-PG-1",
      spec = (vim.host.VirtualNic.Specification) {
        dynamicType = <unset>,
        ip = (vim.host.IpConfig) {
          dynamicType = <unset>,
          dhcp = false,
          ipAddress = "192.168.9.202",
          subnetMask = "255.255.255.0",
          ipV6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
        },
        mac = "00:50:56:77:00:94",
        distributedVirtualPort = (vim.dvs.PortConnection) null,
        portgroup = "VMk-SWiSCSI-PG-1",
        mtu = 1500,
        tsoEnabled = true,
      },
    },
  ],
}
```

```

    port = <unset>,
  },
  (vim.host.VirtualNic) {
    dynamicType = <unset>,
    device = "vmk2",
    key = "VMotionConfig.vmotion.key-vim.host.VirtualNic-vmk2",
    portgroup = "VMk-SWiSCSI-PG-2",
    spec = (vim.host.VirtualNic.Specification) {
      dynamicType = <unset>,
      ip = (vim.host.IpConfig) {
        dynamicType = <unset>,
        dhcp = false,
        ipAddress = "192.168.9.203",
        subnetMask = "255.255.255.0",
        ipv6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
      },
      mac = "00:50:56:7f:d2:ea",
      distributedVirtualPort = (vim.dvs.PortConnection) null,
      portgroup = "VMk-SWiSCSI-PG-2",
      mtu = 1500,
      tsoEnabled = true,
    },
    port = <unset>,
  },
  (vim.host.VirtualNic) {
    dynamicType = <unset>,
    device = "vmk3",
    key = "VMotionConfig.vmotion.key-vim.host.VirtualNic-vmk3",
    portgroup = "pgVMKTest",
    spec = (vim.host.VirtualNic.Specification) {
      dynamicType = <unset>,
      ip = (vim.host.IpConfig) {
        dynamicType = <unset>,
        dhcp = false,
        ipAddress = "192.168.100.100",
        subnetMask = "255.255.255.0",
        ipv6Config = (vim.host.IpConfig.IpV6AddressConfiguration) null,
      },
      mac = "00:50:56:7b:6b:a2",
      distributedVirtualPort = (vim.dvs.PortConnection) null,
      portgroup = "pgVMKTest",
      mtu = 1500,
      tsoEnabled = true,
    },
    port = <unset>,
  }
},
selectedVnic = <unset>,
}

```

hostsvc/vmotion/vnic_set

247 | Evaluation (57 days remaining) | vsTest Properties

Configuration | Performance | Configuration

View: Virtual Switch | vNetwork

Networking

- VMK-5WVISC5I-PG-1
 - vmk1 : 192.168.9.202
- Virtual Switch: vS5-SWVISC5I-2
 - Virtual Machine Port Group: VM-2
 - VMkernel Port: VMK-5WVISC5I-PG-2
 - vmk2 : 192.168.9.203
- Virtual Switch: vsTest
 - VMkernel Port: pgVMKTest
 - vmk3 : 192.168.100.100

Ports | Network Adapters

Configuration	Summary
vSwitch	8 Ports
pgVMKTest	vMotion and IP ...

Port Properties

Network Label: pgVMKTest
 VLAN ID: None (0)
vMotion: Disabled
 Fault Tolerance logging: Disabled
 Management traffic: Disabled

NIC Settings

MAC Address: 00:50:56:7b:6b:a2

IP Settings

IP Address: 192.168.100.100
 Subnet Mask: 255.255.255.0

Effective Policies

Security

Promiscuous Mode: Reject

~ # vim-cmd hostsvc/vmotion/vnic_set vmk3

~ #

VMK2 : 192.168.9.203

Virtual Switch: vsTest

- VMkernel Port: pgVMKTest
 - vmk3 : 192.168.100.100

Virtual Switch: TS

- Virtual Machine Port Group: TSpg

vsTest Properties

Ports | Network Adapters

Configuration	Summary
vSwitch	8 Ports
pgVMKTest	vMotion and IP ...

Port Properties

Network Label: pgVMKTest
 VLAN ID: None (0)
vMotion: Enabled
 Fault Tolerance logging: Disabled
 Management traffic: Disabled

hostsvc/vmotion/vnic_unset

~ # vim-cmd hostsvc/vmotion/vnic_unset vmk3

~ #

Virtual Switch: vsTest

- VMkernel Port: pgVMKTest
 - vmk3 : 192.168.100.100

Virtual Switch: TS

- Virtual Machine Port Group: TSpg

vsTest Properties

Ports | Network Adapters

Configuration	Summary
vSwitch	8 Ports
pgVMKTest	vMotion and IP ...

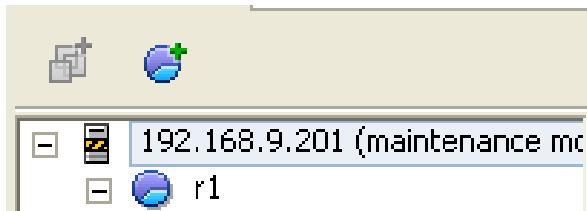
Port Properties

Network Label: pgVMKTest
 VLAN ID: None (0)
vMotion: Disabled
 Fault Tolerance logging: Disabled
 Management traffic: Disabled

hostsvc/maintenance_mode_enter

~ # vim-cmd hostsvc/maintenance_mode_enter

'vim.Task:haTask-ha-host-vim.HostSystem.enterMaintenanceMode-4406'






* **hostsvc/start_remote_tsm**

* **hostsvc/connect**

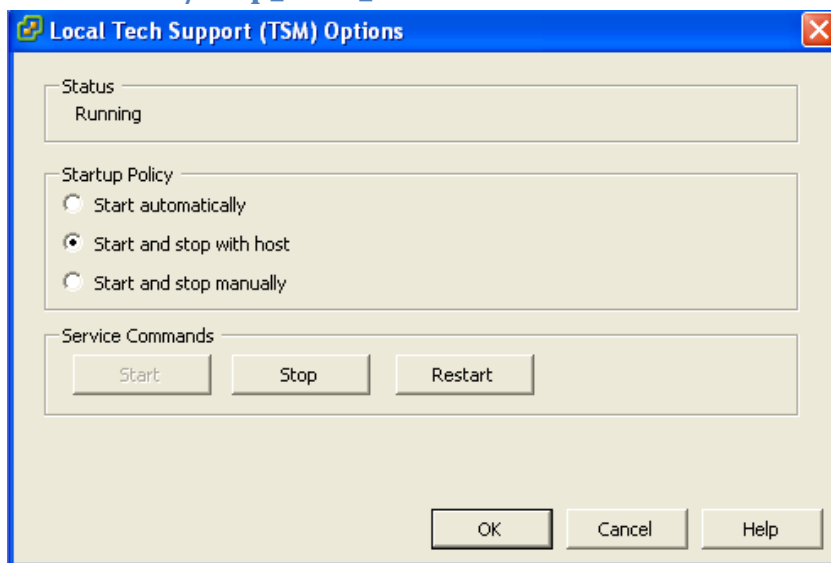
hostsvc/maintenance_mode_exit

~ # vim-cmd hostsvc/maintenance_mode_exit

'vim.Task:haTask-ha-host-vim.HostSystem.exitMaintenanceMode-4418'

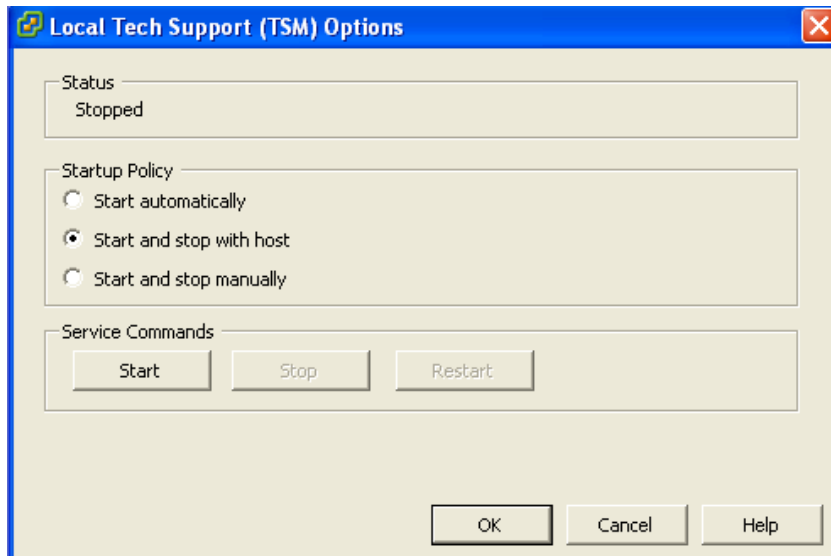
Recent Tasks		
Name	Target	Status
 Exit maintenance mode	 192.168.9.201	 Completed

* **hostsvc/stop_local_tsm**



```
~ # vim-cmd hostsvc/stop_local_tsm
```

```
~ #
```



hostsvc/cpuinfo

```
~ # vim-cmd hostsvc/cpuinfo
```

```
(vim.host.CpuSchedulerSystem.HyperThreadScheduleInfo) {
  dynamicType = <unset>,
  available = false,
  active = false,
  config = true,
}
```

* hostsvc/memoryinfo

* hostsvc/stop_remote_tsm

* hostsvc/disable_local_tsm

* hostsvc/pci_add

hostsvc/task_list

```
~ # vim-cmd hostsvc/task_list
```

```
(ManagedObjectReference) [
  'vim.Task:haTask-ha-host-vim.host.VMotionSystem.deselectVnic-4387',
  'vim.Task:haTask-ha-host-vim.HostSystem.enterMaintenanceMode-4406',
  'vim.Task:haTask-ha-host-vim.HostSystem.exitMaintenanceMode-4418',
  'vim.Task:haTask-ha-host-vim.host.ServiceSystem.stop-4444'
]
```

* **hostsvc/disable_remote_tsm**

* **hostsvc/pci_remove**

proxysvc/

* proxysvc/add_np_service

* proxysvc/add_tcp_service login

* proxysvc/connect

* proxysvc/logout

proxysvc/port_info

~ # vim-cmd proxysvc/port_info

Http Port: 80

Https Port: 443

* proxysvc/remove_service

proxysvc/service_list

~ # vim-cmd proxysvc/service_list

```
(vim.ProxyService.EndpointSpec) [
  (vim.ProxyService.LocalServiceSpec) {
    dynamicType = <unset>,
    serverNamespace = "",
    accessMode = "httpsWithRedirect",
    port = 8309,
  },
  (vim.ProxyService.LocalServiceSpec) {
    dynamicType = <unset>,
    serverNamespace = "/client/clients.xml",
    accessMode = "httpAndHttps",
    port = 8309,
  },
  (vim.ProxyService.LocalServiceSpec) {
    dynamicType = <unset>,
    serverNamespace = "/ha-nfc",
    accessMode = "httpAndHttps",
    port = 12001,
  },
  (vim.ProxyService.NamedPipeServiceSpec) {
    dynamicType = <unset>,
    serverNamespace = "/mob",
    accessMode = "httpsWithRedirect",
    pipeName = "/var/run/vmware/proxy-mob",
  },
  (vim.ProxyService.LocalServiceSpec) {
    dynamicType = <unset>,
    serverNamespace = "/nfc",
    accessMode = "httpAndHttps",
    port = 12000,
  },
  (vim.ProxyService.LocalServiceSpec) {
```

```
dynamicType = <unset>,
serverNamespace = "/sdk",
accessMode = "httpsWithRedirect",
port = 8307,
},
(vim.ProxyService.NamedPipeTunnelSpec) {
dynamicType = <unset>,
serverNamespace = "/sdkTunnel",
accessMode = "httpOnly",
pipeName = "/var/run/vmware/proxy-sdk-tunnel",
},
(vim.ProxyService.LocalServiceSpec) {
dynamicType = <unset>,
serverNamespace = "/ui",
accessMode = "httpsWithRedirect",
port = 8308,
},
(vim.ProxyService.LocalServiceSpec) {
dynamicType = <unset>,
serverNamespace = "/vpxa",
accessMode = "httpsOnly",
port = 8089,
},
(vim.ProxyService.LocalServiceSpec) {
dynamicType = <unset>,
serverNamespace = "/wsman",
accessMode = "httpsWithRedirect",
port = 8889,
}
]
```

supportsvc_cmds/

- * **supportsvc_cmds/connect**
- * **supportsvc_cmds/login**
- * **supportsvc_cmds/logout**
- * **supportsvc_cmds/printstatstofile**

vmsvc/

~ # vim-cmd vmsvc/getallvms

Vmid	Name	File	Guest OS	Version	Annotation
176	ubuntuServer8.04-2	[NFS] ubuntuServer8.04-2/ubuntuServer8.04-2.vmx	ubuntuGuest	vmx-07	ubuntu Server 8.04 LTS x86
192	VM5	[NFS] VM5/VM5.vmx	winXPProGuest	vmx-07	
208	VM1	[NFS] VM1/VM1.vmx	winXPProGuest	vmx-07	
224	ubuntuServer8.04-1	[NFS] ubuntuServer8.04-1/ubuntuServer8.04-1.vmx	ubuntuGuest	vmx-07	ubuntu Server 8.04 LTS x86
240	New Virtual Machine	[of-3] New Virtual Machine/New Virtual Machine.vmx	windows7Server64Guest	vmx-07	
256	VM2	[NFS] VM2/VM2.vmx	winXPProGuest	vmx-07	

*** vmsvc/acquiremksticket***** vmsvc/acquireticket***** vmsvc/connect***** vmsvc/convert.toTemplate***** vmsvc/convert.toVm***** vmsvc/createdummyvm***** vmsvc/destroy***** vmsvc/device.connection***** vmsvc/device.connusbdev***** vmsvc/device.disconnusbdev***** vmsvc/device.diskadd***** vmsvc/device.diskaddexisting***** vmsvc/device.diskremove****vmsvc/device.getdevices**

~ # vim-cmd vmsvc/device.getdevices 80

Devices:

```
(vim.vm.VirtualHardware) {
  dynamicType = <unset>,
  numCPU = 1,
  memoryMB = 512,
  device = (vim.vm.device.VirtualDevice) [
    (vim.vm.device.VirtualIDEController) {
      dynamicType = <unset>,
      key = 200,
      deviceInfo = (vim.Description) {
```

```

    dynamicType = <unset>,
    label = "IDE 0",
    summary = "IDE 0",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = <unset>,
  unitNumber = <unset>,
  busNumber = 0,
  device = (int) [
    3000
  ],
},
(vim.vm.device.VirtualIDEController) {
  dynamicType = <unset>,
  key = 201,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "IDE 1",
    summary = "IDE 1",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = <unset>,
  unitNumber = <unset>,
  busNumber = 1,
  device = (int) [
    3002
  ],
},
(vim.vm.device.VirtualPS2Controller) {
  dynamicType = <unset>,
  key = 300,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "PS2 controller 1",
    summary = "PS2 controller 1",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = <unset>,
  unitNumber = <unset>,
  busNumber = 0,
  device = (int) [
    600,
    700
  ],
},
(vim.vm.device.VirtualPCIController) {
  dynamicType = <unset>,
  key = 100,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "PCI controller 1",

```

```

    summary = "PCI controller 1",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = <unset>,
  unitNumber = <unset>,
  busNumber = 0,
  device = (int) [
    500,
    12000,
    4000,
    4001,
    1000
  ],
},
(vim.vm.device.VirtualSIOController) {
  dynamicType = <unset>,
  key = 400,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "SIO controller 1",
    summary = "SIO controller 1",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = <unset>,
  unitNumber = <unset>,
  busNumber = 0,
},
(vim.vm.device.VirtualKeyboard) {
  dynamicType = <unset>,
  key = 600,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "Keyboard ",
    summary = "Keyboard",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = 300,
  unitNumber = 0,
},
(vim.vm.device.VirtualPointingDevice) {
  dynamicType = <unset>,
  key = 700,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "Pointing device",
    summary = "Pointing device; Device",
  },
  backing = (vim.vm.device.VirtualPointingDevice.DeviceBackingInfo) {
    dynamicType = <unset>,
    deviceName = "",
    useAutoDetect = false,

```

```

    hostPointingDevice = "autodetect",
  },
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = 300,
  unitNumber = 1,
},
(vim.vm.device.VirtualVideoCard) {
  dynamicType = <unset>,
  key = 500,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "Video card ",
    summary = "Video card",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = 100,
  unitNumber = 0,
  videoRamSizeInKB = 33792,
  numDisplays = <unset>,
  useAutoDetect = false,
  enable3DSupport = false,
},
(vim.vm.device.VirtualVMCIDevice) {
  dynamicType = <unset>,
  key = 12000,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "VMCI device",
    summary = "Device on the virtual machine PCI bus that provides support for the virtual machine communication interface",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = 100,
  unitNumber = 17,
  id = <unset>,
  allowUnrestrictedCommunication = false,
},
(vim.vm.device.VirtualDisk) {
  dynamicType = <unset>,
  key = 3000,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "Hard disk 1",
    summary = "8388608 KB",
  },
  backing = (vim.vm.device.VirtualDisk.FlatVer2BackingInfo) {
    dynamicType = <unset>,
    fileName = "[NFS] XP-1/XP-1.vmdk",
    datastore = 'vim.Datastore:192.168.9.222:/media/dwl/public/NFS',
    diskMode = "persistent",
    split = false,
    writeThrough = false,
    thinProvisioned = true,
  },

```

```

    eagerlyScrub = <unset>,
    uuid = "6000C29e-d75f-456d-c2b0-c57f10365942",
    contentId = "22354fb9",
    changeld = <unset>,
    parent = (vim.vm.device.VirtualDisk.FlatVer2BackingInfo) null,
  },
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = 200,
  unitNumber = 0,
  capacityInKB = 8388608,
  shares = (vim.SharesInfo) null,
},
(vim.vm.device.VirtualCdrom) {
  dynamicType = <unset>,
  key = 3002,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "CD/DVD Drive 1",
    summary = "Remote device /usr/lib/vmware/isoimages/windows.iso",
  },
  backing = (vim.vm.device.VirtualCdrom.RemotePassthroughBackingInfo) {
    dynamicType = <unset>,
    deviceName = "/usr/lib/vmware/isoimages/windows.iso",
    useAutoDetect = false,
    exclusive = true,
  },
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) {
    dynamicType = <unset>,
    startConnected = false,
    allowGuestControl = true,
    connected = false,
    status = "untried",
  },
  controllerKey = 201,
  unitNumber = 0,
},
(vim.vm.device.VirtualPCNet32) {
  dynamicType = <unset>,
  key = 4000,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "Network adapter 1",
    summary = "VM Network",
  },
  backing = (vim.vm.device.VirtualEthernetCard.NetworkBackingInfo) {
    dynamicType = <unset>,
    deviceName = "VM Network",
    useAutoDetect = false,
    network = 'vim.Network:HaNetwork-VM Network',
    inPassthroughMode = <unset>,
  },
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) {
    dynamicType = <unset>,
    startConnected = true,

```

```

    allowGuestControl = true,
    connected = false,
    status = "untried",
  },
  controllerKey = 100,
  unitNumber = 7,
  addressType = "assigned",
  macAddress = "00:50:56:85:00:09",
  wakeOnLanEnabled = true,
},
(vim.vm.device.VirtualPCNet32) {
  dynamicType = <unset>,
  key = 4001,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "Network adapter 2",
    summary = "VM Network",
  },
  backing = (vim.vm.device.VirtualEthernetCard.NetworkBackingInfo) {
    dynamicType = <unset>,
    deviceName = "VM Network",
    useAutoDetect = false,
    network = 'vim.Network:HaNetwork-VM Network',
    inPassthroughMode = <unset>,
  },
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) {
    dynamicType = <unset>,
    startConnected = false,
    allowGuestControl = true,
    connected = false,
    status = "untried",
  },
  controllerKey = 100,
  unitNumber = 8,
  addressType = "assigned",
  macAddress = "00:50:56:85:00:0a",
  wakeOnLanEnabled = true,
},
(vim.vm.device.VirtualBusLogicController) {
  dynamicType = <unset>,
  key = 1000,
  deviceInfo = (vim.Description) {
    dynamicType = <unset>,
    label = "SCSI controller 0",
    summary = "BusLogic",
  },
  backing = (vim.vm.device.VirtualDevice.BackingInfo) null,
  connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
  controllerKey = 100,
  unitNumber = 3,
  busNumber = 0,
  device = (int) [
    2000
  ],
},

```

```

hotAddRemove = true,
sharedBus = "noSharing",
scsiCtrlUnitNumber = 7,
},
(vim.vm.device.VirtualDisk) {
dynamicType = <unset>,
key = 2000,
deviceInfo = (vim.Description) {
dynamicType = <unset>,
label = "Hard disk 2",
summary = "8192 KB",
},
backing = (vim.vm.device.VirtualDisk.FlatVer2BackingInfo) {
dynamicType = <unset>,
fileName = "[NFS] XP-1/XP-1_1.vmdk",
datastore = 'vim.Datastore:192.168.9.222:/media/dwl/public/NFS',
diskMode = "persistent",
split = false,
writeThrough = false,
thinProvisioned = true,
eagerlyScrub = <unset>,
uuid = "6000C294-6016-8534-6e26-6a00f7a426e5",
contentId = "443fe4ee3d80d2526497472a250feb06",
changeId = <unset>,
parent = (vim.vm.device.VirtualDisk.FlatVer2BackingInfo) null,
},
connectable = (vim.vm.device.VirtualDevice.ConnectInfo) null,
controllerKey = 1000,
unitNumber = 0,
capacityInKB = 8192,
shares = (vim.SharesInfo) {
dynamicType = <unset>,
shares = 1000,
level = "normal",
},
},
},
}
}

```

*** vmsvc/device.toolsSyncSet**

*** vmsvc/device.vmiadd**

*** vmsvc/device.vmiremove**

*** vmsvc/devices.createnic**

vmsvc/get.capability

~ # vim-cmd vmsvc/get.capability 80

```
(vim.vm.Capability) {
  dynamicType = <unset>,
  snapshotOperationsSupported = true,
  multipleSnapshotsSupported = true,
  snapshotConfigSupported = true,
  poweredOffSnapshotsSupported = true,
  memorySnapshotsSupported = true,
  revertToSnapshotSupported = true,
  quiescedSnapshotsSupported = true,
  disableSnapshotsSupported = false,
  lockSnapshotsSupported = false,
  consolePreferencesSupported = false,
  cpuFeatureMaskSupported = true,
  s1AcpiManagementSupported = true,
  settingScreenResolutionSupported = true,
  toolsAutoUpdateSupported = true,
  vmNpivWwnSupported = true,
  npivWwnOnNonRdmVmSupported = true,
  vmNpivWwnDisableSupported = true,
  vmNpivWwnUpdateSupported = true,
  swapPlacementSupported = true,
  swapPreservationSupported = false,
  toolsSyncTimeSupported = true,
  virtualMmuUsageSupported = true,
  diskSharesSupported = true,
  bootOptionsSupported = true,
  settingVideoRamSizeSupported = true,
  settingDisplayTopologySupported = true,
  recordReplaySupported = true,
  changeTrackingSupported = true,
}
```

vmsvc/get.config

~ # vim-cmd vmsvc/get.config 80

Configuration:

```
(vim.vm.ConfigInfo) {
  dynamicType = <unset>,
  changeVersion = "2010-08-20T10:37:24.663528Z",
  modified = "1970-01-01T00:00:00Z",
  name = "XP-1",
  guestFullName = "Microsoft Windows XP Professional (32-bit)",
  version = "vmx-07",
  uuid = "42054c7a-8d9d-c36e-44f8-05855b5aec3e",
  instanceUuid = "5005dd1e-61c9-d323-216d-30eed4585753",
  npivWorldWideNameType = "",
  npivDesiredNodeWwns = <unset>,
  npivDesiredPortWwns = <unset>,
}
```

```

npivTemporaryDisabled = true,
npivOnNonRdmDisks = <unset>,
locationId = "564d356b-a048-75d7-a381-884dcf5fe6a8",
template = false,
guestId = "winXPProGuest",
alternateGuestName = "Microsoft Windows XP Professional (32-bit)",
annotation = "FEGFE",
files = (vim.vm.FileInfo) {
    dynamicType = <unset>,
    vmPathName = "[NFS] XP-1/XP-1.vmx",
    snapshotDirectory = "[NFS] XP-1",
    suspendDirectory = "[NFS] XP-1",
    logDirectory = "[NFS] XP-1",
},
tools = (vim.vm.ToolsConfigInfo) {
    dynamicType = <unset>,
    toolsVersion = 8194,
    afterPowerOn = true,
    afterResume = true,
    beforeGuestStandby = true,
    beforeGuestShutdown = true,
    beforeGuestReboot = <unset>,
    toolsUpgradePolicy = "manual",
    pendingCustomization = <unset>,
    syncTimeWithHost = false,
},
flags = (vim.vm.FlagInfo) {
    dynamicType = <unset>,
    disableAcceleration = false,
    enableLogging = true,
    useToe = false,
    runWithDebugInfo = false,
    monitorType = "release",
    htSharing = "any",
    snapshotDisabled = <unset>,
    snapshotLocked = <unset>,
    diskUuidEnabled = false,
    virtualMmuUsage = "automatic",
    virtualExecUsage = "hvAuto",
    snapshotPowerOffBehavior = "powerOff",
    recordReplayEnabled = false,
},
consolePreferences = (vim.vm.ConsolePreferences) null,
defaultPowerOps = (vim.vm.DefaultPowerOpInfo) {
    dynamicType = <unset>,
    powerOffType = "preset",
    suspendType = "preset",
    resetType = "preset",
    defaultPowerOffType = "soft",
    defaultSuspendType = "hard",
    defaultResetType = "soft",
    standbyAction = "powerOnSuspend",
},
hardware = (vim.vm.VirtualHardware) {

```

```

dynamicType = <unset>,
numCPU = 0,
memoryMB = 28941319,
},
cpuAllocation = (vim.ResourceAllocationInfo) {
dynamicType = <unset>,
reservation = 0,
expandableReservation = <unset>,
limit = -1,
shares = (vim.SharesInfo) {
dynamicType = <unset>,
shares = 1000,
level = "normal",
},
overheadLimit = <unset>,
},
memoryAllocation = (vim.ResourceAllocationInfo) {
dynamicType = <unset>,
reservation = 512,
expandableReservation = <unset>,
limit = -1,
shares = (vim.SharesInfo) {
dynamicType = <unset>,
shares = 5120,
level = "normal",
},
overheadLimit = <unset>,
},
memoryHotAddEnabled = false,
cpuHotAddEnabled = false,
cpuHotRemoveEnabled = false,
hotPlugMemoryLimit = 512,
hotPlugMemoryIncrementSize = 0,
cpuAffinity = (vim.vm.AffinityInfo) null,
memoryAffinity = (vim.vm.AffinityInfo) null,
networkShaper = (vim.vm.NetworkShaperInfo) null,
extraConfig = (vim.option.OptionValue) [
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "nvram",
value = "XP-1.nvram",
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "deploymentPlatform",
value = "windows",
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "virtualHW.productCompatibility",
value = "hosted",
},
(vim.option.OptionValue) {
dynamicType = <unset>,

```

```

    key = "unity.customColor",
    value = "#C0C0C0",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "pciBridge0.present",
    value = "true",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "pciBridge4.present",
    value = "true",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "snapshot.action",
    value = "keep",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "pciBridge4.virtualDev",
    value = "pcieRootPort",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "vmware.tools.requiredversion",
    value = "8195",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "ethernet0.pciSlotNumber",
    value = "32",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "evcCompatibilityMode",
    value = "false",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "guestCPUID.0",
    value = "00000006756e65476c65746e49656e69",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "guestCPUID.1",
    value = "0000066200010800800020010febfbff",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "guestCPUID.80000001",
    value = "00000000000000000000000000000000120100800",
  },
  (vim.option.OptionValue) {

```

```

dynamicType = <unset>,
key = "pciBridge4.functions",
value = "8",
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "hostCPUID.0",
value = "00000006756e65476c65746e49656e69",
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "hostCPUID.1",
value = "00000f62000208000000e43dbfebfbff",
},
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "hostCPUID.80000001",
value = "000000000000000000000000120100800",
},
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "ide0:0.redo",
value = "",
},
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "pciBridge0.pciSlotNumber",
value = "17",
},
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "pciBridge4.pciSlotNumber",
value = "21",
},
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "pciBridge5.pciSlotNumber",
value = "22",
},
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "pciBridge6.pciSlotNumber",
value = "23",
},
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "pciBridge7.pciSlotNumber",
value = "24",
},
},
(vim.option.OptionValue) {
dynamicType = <unset>,
key = "replay.supported",
value = "false",
},
},

```

```
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "userCPUID.0",
  value = "00000006756e65476c65746e49656e69",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "userCPUID.1",
  value = "00000f62000208000000e43dbfebfbff",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "userCPUID.80000001",
  value = "00000000000000000000000000000000120100800",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "vmci0.pciSlotNumber",
  value = "36",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "vmotion.checkpointFBSize",
  value = "34603008",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "tools.deployPkg.fileName",
  value = "",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "pciBridge5.present",
  value = "true",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "replay.allowFT",
  value = "false",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "svga.numDisplays",
  value = "1",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "svga.maxWidth",
  value = "800",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "svga.maxHeight",
  value = "600",
}
```

```

},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "machine.id",
  value = "",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "ethernet1.pciSlotNumber",
  value = "33",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "checkpoint.vmState",
  value = "",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "scsi0.pciSlotNumber",
  value = "34",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "scsi0:0.ctkEnabled",
  value = "false",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "scsi0:0.redo",
  value = "",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "sched.swap.derivedName",
  value = "/vmfs/volumes/1c445c7d-fd56af7a/XP-1/XP-1-4895505e.vswp",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "pciBridge5.virtualDev",
  value = "pcieRootPort",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "pciBridge5.functions",
  value = "8",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "pciBridge6.present",
  value = "true",
},
(vim.option.OptionValue) {
  dynamicType = <unset>,
  key = "pciBridge6.virtualDev",

```

```

    value = "pcieRootPort",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "pciBridge6.functions",
    value = "8",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "pciBridge7.present",
    value = "true",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "pciBridge7.virtualDev",
    value = "pcieRootPort",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "pciBridge7.functions",
    value = "8",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "vmware.tools.internalversion",
    value = "8194",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "vmware.tools.requiredversion",
    value = "8194",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "vmware.tools.installstate",
    value = "none",
  },
  (vim.option.OptionValue) {
    dynamicType = <unset>,
    key = "vmware.tools.lastInstallStatus",
    value = "unknown",
  }
},
datastoreUrl = (vim.vm.ConfigInfo.DatastoreUrlPair) [
  (vim.vm.ConfigInfo.DatastoreUrlPair) {
    dynamicType = <unset>,
    name = "NFS",
    url = "/vmfs/volumes/1c445c7d-fd56af7a/",
  }
],
swapPlacement = "inherit",
swapDirectory = <unset>,
preserveSwapOnPowerOff = <unset>,
bootOptions = (vim.vm.BootOptions) {

```

```

dynamicType = <unset>,
bootDelay = 0,
enterBIOSSetup = false,
},
ftInfo = (vim.vm.FaultToleranceConfigInfo) null,
vAppConfig = (vim.vApp.VmConfigInfo) null,
vAssertsEnabled = false,
changeTrackingEnabled = false,
}

```

vmsvc/get.config.cpuidmask

~ # vim-cmd vmsvc/get.config.cpuidmask 80

```

(vim.host.CpuidInfo) [
(vim.host.CpuidInfo) {
dynamicType = <unset>,
level = 0,
vendor = <unset>,
eax = <unset>,
ebx = <unset>,
ecx = <unset>,
edx = <unset>,
},
(vim.host.CpuidInfo) {
dynamicType = <unset>,
level = 0,
vendor = "amd",
eax = <unset>,
ebx = <unset>,
ecx = <unset>,
edx = <unset>,
},
(vim.host.CpuidInfo) {
dynamicType = <unset>,
level = 1,
vendor = <unset>,
eax = "-----0-0-----",
ebx = <unset>,
ecx = "-----0-----0-----",
edx = <unset>,
},
(vim.host.CpuidInfo) {
dynamicType = <unset>,
level = 1,
vendor = "amd",
eax = <unset>,
ebx = <unset>,
ecx = <unset>,
edx = <unset>,
},
(vim.host.CpuidInfo) {
dynamicType = <unset>,

```

```

    level = -2147483648,
    vendor = <unset>,
    eax = <unset>,
    ebx = <unset>,
    ecx = <unset>,
    edx = <unset>,
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483648,
    vendor = "amd",
    eax = <unset>,
    ebx = <unset>,
    ecx = <unset>,
    edx = <unset>,
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = <unset>,
    eax = "-----",
    ebx = "-----",
    ecx = "-----",
    edx = "-----H-----",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = "amd",
    eax = "-----",
    ebx = "-----",
    ecx = "-----",
    edx = "-----H-----",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483640,
    vendor = <unset>,
    eax = <unset>,
    ebx = <unset>,
    ecx = <unset>,
    edx = <unset>,
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483640,
    vendor = "amd",
    eax = <unset>,
    ebx = <unset>,
    ecx = <unset>,
    edx = <unset>,
}
]

```

vmsvc/get.configoption

~ # vim-cmd vmsvc/get.configoption 80

```
(vim.vm.ConfigOption) {
  dynamicType = <unset>,
  version = "vmx-07",
  description = "Hardware version 7 virtual machine",
  guestOSDescriptor = (vim.vm.GuestOsDescriptor) [
    (vim.vm.GuestOsDescriptor) {
      dynamicType = <unset>,
      id = "windows7Guest",
      family = "windowsGuest",
      fullName = "Microsoft Windows 7 (32-bit)",
      supportedMaxCPUs = 8,
      supportedMinMemMB = 512,
      supportedMaxMemMB = 4096,
      recommendedMemMB = 1024,
      recommendedColorDepth = 8,
      supportedDiskControllerList = (vmodl.TypeName) [
        "vim.vm.device.VirtualLsiLogicController",
        "vim.vm.device.VirtualLsiLogicSASController",
        "vim.vm.device.VirtualIDEController"
      ],
      recommendedSCSIController = "vim.vm.device.VirtualLsiLogicSASController",
      recommendedDiskController = "vim.vm.device.VirtualLsiLogicSASController",
      supportedNumDisks = 16,
      recommendedDiskSizeMB = 24576,
      supportedEthernetCard = (vmodl.TypeName) [
        "vim.vm.device.VirtualE1000",
        "vim.vm.device.VirtualVmxnet3"
      ],
      recommendedEthernetCard = "vim.vm.device.VirtualE1000",
      supportsSlaveDisk = true,
      cpuFeatureMask = (vim.host.CpuldInfo) [
        (vim.host.CpuldInfo) {
          dynamicType = <unset>,
          level = -2147483647,
          vendor = <unset>,
          eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
          ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
          ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRx",
          edx = "RRxR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
        },
        (vim.host.CpuldInfo) {
          dynamicType = <unset>,
          level = 1,
          vendor = "amd",
          eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
          ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
          ecx = "xRRR:RRRR:HRRR:RRRR:RRxR:RRHR:RRRR:0RRH",
          edx = "RRR0:RHHH:HRRR:HRHH:HHHH:HR1H:HHHH:HHHH",
        },
      ],
    },
  ],
  recommendedEthernetCard = "vim.vm.device.VirtualE1000",
  supportsSlaveDisk = true,
  cpuFeatureMask = (vim.host.CpuldInfo) [
    (vim.host.CpuldInfo) {
      dynamicType = <unset>,
      level = -2147483647,
      vendor = <unset>,
      eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
      ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
      ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRx",
      edx = "RRxR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
    },
    (vim.host.CpuldInfo) {
      dynamicType = <unset>,
      level = 1,
      vendor = "amd",
      eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
      ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
      ecx = "xRRR:RRRR:HRRR:RRRR:RRxR:RRHR:RRRR:0RRH",
      edx = "RRR0:RHHH:HRRR:HRHH:HHHH:HR1H:HHHH:HHHH",
    },
  ],
}
```

```

dynamicType = <unset>,
level = 0,
vendor = <unset>,
eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
dynamicType = <unset>,
level = 0,
vendor = "amd",
eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
dynamicType = <unset>,
level = 1,
vendor = <unset>,
eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ecx = "xRRR:R0HR:HR0H:H0RR:00xR:R0H0:0000:00HH",
edx = "0000:HHHH:HHHR:HHHH:HHHH:HR1H:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
dynamicType = <unset>,
level = -2147483647,
vendor = "amd",
eax = "RRRR:xxxx:xxxx:xxxx:RRxx:xxxx:xxxx:xxxx",
ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ecx = "RRRR:RRRR:RRRR:RRRR:RR00:R00H:HHH0:H00x",
edx = "HHxR:H0HH:HHRH:RRHH:HHHH:xR1H:HHHH:HHHH",
}
},
supportsWakeOnLan = true,
supportsVMI = false,
supportsMemoryHotAdd = true,
supportsCpuHotAdd = true,
supportsCpuHotRemove = false,
},
(vim.vm.GuestOsDescriptor) {
dynamicType = <unset>,
id = "windows7_64Guest",
family = "windowsGuest",
fullName = "Microsoft Windows 7 (64-bit)",
supportedMaxCPUs = 8,
supportedMinMemMB = 512,
supportedMaxMemMB = 131072,
recommendedMemMB = 2048,
recommendedColorDepth = 8,
supportedDiskControllerList = (vmodl.TypeName) [
"vim.vm.device.VirtualLsiLogicController",

```

```

"vim.vm.device.VirtualLsiLogicSASController",
"vim.vm.device.VirtualIDEController"
],
recommendedSCSIController = "vim.vm.device.VirtualLsiLogicSASController",
recommendedDiskController = "vim.vm.device.VirtualLsiLogicSASController",
supportedNumDisks = 16,
recommendedDiskSizeMB = 32768,
supportedEthernetCard = (vmodl.TypeName) [
"vim.vm.device.VirtualE1000",
"vim.vm.device.VirtualVmxnet3"
],
recommendedEthernetCard = "vim.vm.device.VirtualE1000",
supportsSlaveDisk = true,
cpuFeatureMask = (vim.host.CpuldInfo) [
(vim.host.CpuldInfo) {
dynamicType = <unset>,
level = -2147483647,
vendor = <unset>,
eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRH",
edx = "RRTR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
},
(vim.host.CpuldInfo) {
dynamicType = <unset>,
level = 1,
vendor = "amd",
eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ecx = "xRRR:RRRR:HRRR:RRRR:RRHR:RRHR:RRRR:0RRH",
edx = "RRR0:RHHH:HRRR:HRHH:HHHH:HR1H:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
dynamicType = <unset>,
level = 0,
vendor = <unset>,
eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
dynamicType = <unset>,
level = 0,
vendor = "amd",
eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
dynamicType = <unset>,
level = 1,
vendor = <unset>,

```

```

    eax = "RRRR:HHHH:HHHH:xxx:RRxx:HHHH:xxx:xxx",
    ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ecx = "xRRR:R0HR:HR0H:H0RR:00HR:R0H0:0000:00HH",
    edx = "0000:HHHH:HHHR:HHHH:HHHH:HR1H:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = "amd",
    eax = "RRRR:xxx:xxx:xxx:RRxx:xxx:xxx:xxx",
    ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RR00:R00H:HHH0:H00H",
    edx = "HHTR:H0HH:HHRH:RRHH:HHHH:xR1H:HHHH:HHHH",
}
],
supportsWakeOnLan = true,
supportsVMI = false,
supportsMemoryHotAdd = true,
supportsCpuHotAdd = true,
supportsCpuHotRemove = false,
},
(vim.vm.GuestOsDescriptor) {
    dynamicType = <unset>,
    id = "windows7Server64Guest",
    family = "windowsGuest",
    fullName = "Microsoft Windows Server 2008 R2 (64-bit)",
    supportedMaxCPUs = 8,
    supportedMinMemMB = 512,
    supportedMaxMemMB = 261120,
    recommendedMemMB = 4096,
    recommendedColorDepth = 8,
    supportedDiskControllerList = (vmodl.TypeName) [
        "vim.vm.device.ParaVirtualSCSIController",
        "vim.vm.device.VirtualLsiLogicController",
        "vim.vm.device.VirtualLsiLogicSASController",
        "vim.vm.device.VirtualIDEController"
    ],
    recommendedSCSIController = "vim.vm.device.VirtualLsiLogicSASController",
    recommendedDiskController = "vim.vm.device.VirtualLsiLogicSASController",
    supportedNumDisks = 16,
    recommendedDiskSizeMB = 40960,
    supportedEthernetCard = (vmodl.TypeName) [
        "vim.vm.device.VirtualE1000",
        "vim.vm.device.VirtualVmxnet2",
        "vim.vm.device.VirtualVmxnet3"
    ],
    recommendedEthernetCard = "vim.vm.device.VirtualE1000",
    supportsSlaveDisk = true,
    cpuFeatureMask = (vim.host.CpuldInfo) [
        (vim.host.CpuldInfo) {
            dynamicType = <unset>,
            level = -2147483647,
            vendor = <unset>,
            eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",

```

```

    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRH",
    edx = "RRTR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = "amd",
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "xRRR:RRRR:HRRR:RRRR:RRHR:RRHR:RRRR:0RRH",
    edx = "RRR0:RHHH:HRRR:HRHH:HHHH:HR1H:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = <unset>,
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = "amd",
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = <unset>,
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "xRRR:R0HR:HR0H:H0RR:00HR:R0H0:0000:00HH",
    edx = "0000:HHHH:HHHR:HHHH:HHHH:HR1H:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = "amd",
    eax = "RRRR:xxxx:xxxx:xxxx:RRxx:xxxx:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RRR0:R00H:HHH0:H00H",
    edx = "HHTR:H0HH:HHRH:RRHH:HHHH:xR1H:HHHH:HHHH",
}
}
],
supportsWakeOnLan = true,
supportsVMI = false,
supportsMemoryHotAdd = true,
supportsCpuHotAdd = true,

```

```

    supportsCpuHotRemove = false,
  },
  (vim.vm.GuestOsDescriptor) {
    dynamicType = <unset>,
    id = "winLonghornGuest",
    family = "windowsGuest",
    fullName = "Microsoft Windows Server 2008 (32-bit)",
    supportedMaxCPUs = 8,
    supportedMinMemMB = 512,
    supportedMaxMemMB = 65536,
    recommendedMemMB = 2048,
    recommendedColorDepth = 8,
    supportedDiskControllerList = (vmodl.TypeName) [
      "vim.vm.device.ParaVirtualSCSIController",
      "vim.vm.device.VirtualLsiLogicController",
      "vim.vm.device.VirtualLsiLogicSASController",
      "vim.vm.device.VirtualIDController"
    ],
    recommendedSCSIController = "vim.vm.device.VirtualLsiLogicSASController",
    recommendedDiskController = "vim.vm.device.VirtualLsiLogicSASController",
    supportedNumDisks = 16,
    recommendedDiskSizeMB = 40960,
    supportedEthernetCard = (vmodl.TypeName) [
      "vim.vm.device.VirtualE1000",
      "vim.vm.device.VirtualVmxnet2",
      "vim.vm.device.VirtualVmxnet3"
    ],
    recommendedEthernetCard = "vim.vm.device.VirtualE1000",
    supportsSlaveDisk = true,
    cpuFeatureMask = (vim.host.CpuidInfo) [
      (vim.host.CpuidInfo) {
        dynamicType = <unset>,
        level = -2147483647,
        vendor = <unset>,
        eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
        ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
        ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRx",
        edx = "RRxR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
      },
      (vim.host.CpuidInfo) {
        dynamicType = <unset>,
        level = 1,
        vendor = "amd",
        eax = "RRRR:HHHH:HHHH:xxx:RRx:HHHH:xxx:xxx",
        ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
        ecx = "xRRR:RRRR:HRRR:RRRR:RRxR:RRHR:RRRR:0RRH",
        edx = "RRR0:RHHH:HRRR:HRHH:HHHH:HR1H:HHHH:HHHH",
      },
      (vim.host.CpuidInfo) {
        dynamicType = <unset>,
        level = 0,
        vendor = <unset>,
        eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
        ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
      }
    ]
  }
}

```

```

    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = "amd",
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = <unset>,
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "xRRR:R0HR:HR0H:H0RR:00xR:R0H0:0000:00HH",
    edx = "0000:HHHH:HHHR:HHHH:HHHH:HR1H:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = "amd",
    eax = "RRRR:xxxx:xxxx:xxxx:RRxx:xxxx:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RR00:R00H:HHH0:H00x",
    edx = "HHxR:H0HH:HHRH:RRHH:HHHH:xR1H:HHHH:HHHH",
}
],
supportsWakeOnLan = true,
supportsVMI = false,
supportsMemoryHotAdd = true,
supportsCpuHotAdd = false,
supportsCpuHotRemove = false,
},
(vim.vm.GuestOsDescriptor) {
    dynamicType = <unset>,
    id = "winLonghorn64Guest",
    family = "windowsGuest",
    fullName = "Microsoft Windows Server 2008 (64-bit)",
    supportedMaxCPUs = 8,
    supportedMinMemMB = 512,
    supportedMaxMemMB = 261120,
    recommendedMemMB = 4096,
    recommendedColorDepth = 8,
    supportedDiskControllerList = (vmodl.TypeName) [
        "vim.vm.device.ParaVirtualSCSIController",
        "vim.vm.device.VirtualLsiLogicController",
        "vim.vm.device.VirtualLsiLogicSASController",
        "vim.vm.device.VirtualIDEController"
    ],
    recommendedSCSIController = "vim.vm.device.VirtualLsiLogicSASController",

```

```

recommendedDiskController = "vim.vm.device.VirtualLsiLogicSASController",
supportedNumDisks = 16,
recommendedDiskSizeMB = 40960,
supportedEthernetCard = (vmtoolsd.TypeName) [
  "vim.vm.device.VirtualE1000",
  "vim.vm.device.VirtualVmxnet2",
  "vim.vm.device.VirtualVmxnet3"
],
recommendedEthernetCard = "vim.vm.device.VirtualE1000",
supportsSlaveDisk = true,
cpuFeatureMask = (vim.host.CpuidInfo) [
  (vim.host.CpuidInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = <unset>,
    eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRH",
    edx = "RRTR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
  },
  (vim.host.CpuidInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = "amd",
    eax = "RRRR:HHHH:HHHH:xxx:RRxx:HHHH:xxx:xxx",
    ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ecx = "xRRR:RRRR:HRRR:RRRR:RRHR:RRHR:RRRR:0RRH",
    edx = "RRR0:RHHH:HRRR:HRHH:HHHH:HR1H:HHHH:HHHH",
  },
  (vim.host.CpuidInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = <unset>,
    eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
  },
  (vim.host.CpuidInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = "amd",
    eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
  },
  (vim.host.CpuidInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = <unset>,
    eax = "RRRR:HHHH:HHHH:xxx:RRxx:HHHH:xxx:xxx",
    ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ecx = "xRRR:R0HR:HR0H:H0RR:00HR:R0H0:0000:00HH",

```

```

    edx = "0000:HHHH:HHHR:HHHH:HHHH:HR1H:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = "amd",
    eax = "RRRR:xxx:xxx:xxx:RRxx:xxx:xxx:xxx",
    ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RR00:R00H:HHH0:H00H",
    edx = "HHTR:H0HH:HHRH:RRHH:HHHH:xR1H:HHHH:HHHH",
  }
],
supportsWakeOnLan = true,
supportsVMI = false,
supportsMemoryHotAdd = true,
supportsCpuHotAdd = true,
supportsCpuHotRemove = false,
},
(vim.vm.GuestOsDescriptor) {
  dynamicType = <unset>,
  id = "winNetEnterpriseGuest",
  family = "windowsGuest",
  fullName = "Microsoft Windows Server 2003, Enterprise Edition (32-bit)",
  supportedMaxCPUs = 8,
  supportedMinMemMB = 512,
  supportedMaxMemMB = 65536,
  recommendedMemMB = 1024,
  recommendedColorDepth = 8,
  supportedDiskControllerList = (vmodl.TypeName) [
    "vim.vm.device.ParaVirtualSCSIController",
    "vim.vm.device.VirtualBusLogicController",
    "vim.vm.device.VirtualLsiLogicController",
    "vim.vm.device.VirtualLsiLogicSASController",
    "vim.vm.device.VirtualIDEController"
  ],
  recommendedSCSIController = "vim.vm.device.VirtualLsiLogicController",
  recommendedDiskController = "vim.vm.device.VirtualLsiLogicController",
  supportedNumDisks = 16,
  recommendedDiskSizeMB = 8192,
  supportedEthernetCard = (vmodl.TypeName) [
    "vim.vm.device.VirtualE1000",
    "vim.vm.device.VirtualPCNet32",
    "vim.vm.device.VirtualVmxnet",
    "vim.vm.device.VirtualVmxnet2",
    "vim.vm.device.VirtualVmxnet3"
  ],
  recommendedEthernetCard = "vim.vm.device.VirtualE1000",
  supportsSlaveDisk = true,
  cpuFeatureMask = (vim.host.CpuldInfo) [
    (vim.host.CpuldInfo) {
      dynamicType = <unset>,
      level = -2147483647,
      vendor = <unset>,
      eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",

```

```

    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRx",
    edx = "RRxR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = "amd",
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "xRRR:RRRR:HRRR:RRRR:RRxR:RRHR:RRRR:0RRH",
    edx = "RRR0:RHHH:HRRR:HRHH:H0HH:HR1H:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = <unset>,
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = "amd",
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = <unset>,
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "xRRR:R0HR:HR0H:H0RR:00xR:ROH0:0000:00HH",
    edx = "0000:HHHH:HHHR:HHHH:H0HH:HR1H:HHHH:HHHH",
},
(vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = "amd",
    eax = "RRRR:xxxx:xxxx:xxxx:RRxx:xxxx:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RR00:R00H:HHH0:H00x",
    edx = "HHxR:H0HH:HHRH:RRHH:H0HH:xR1H:HHHH:HHHH",
}
},
supportsWakeOnLan = true,
supportsVMI = false,
supportsMemoryHotAdd = true,
supportsCpuHotAdd = false,

```

```

    supportsCpuHotRemove = false,
  },
  (vim.vm.GuestOsDescriptor) {
    dynamicType = <unset>,
    id = "winNetEnterprise64Guest",
    family = "windowsGuest",
    fullName = "Microsoft Windows Server 2003, Enterprise Edition (64-bit)",
    supportedMaxCPUs = 8,
    supportedMinMemMB = 512,
    supportedMaxMemMB = 261120,
    recommendedMemMB = 1024,
    recommendedColorDepth = 8,
    supportedDiskControllerList = (vmodl.TypeName) [
      "vim.vm.device.ParaVirtualSCSIController",
      "vim.vm.device.VirtualLsiLogicController",
      "vim.vm.device.VirtualLsiLogicSASController",
      "vim.vm.device.VirtualIDController"
    ],
    recommendedSCSIController = "vim.vm.device.VirtualLsiLogicController",
    recommendedDiskController = "vim.vm.device.VirtualLsiLogicController",
    supportedNumDisks = 16,
    recommendedDiskSizeMB = 8192,
    supportedEthernetCard = (vmodl.TypeName) [
      "vim.vm.device.VirtualE1000",
      "vim.vm.device.VirtualVmxnet2",
      "vim.vm.device.VirtualVmxnet3"
    ],
    recommendedEthernetCard = "vim.vm.device.VirtualE1000",
    supportsSlaveDisk = true,
    cpuFeatureMask = (vim.host.CpuldInfo) [
      (vim.host.CpuldInfo) {
        dynamicType = <unset>,
        level = -2147483647,
        vendor = <unset>,
        eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
        ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
        ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRH",
        edx = "RRTR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
      },
      (vim.host.CpuldInfo) {
        dynamicType = <unset>,
        level = 1,
        vendor = "amd",
        eax = "RRRR:HHHH:HHHH:xxx:RRxx:HHHH:xxx:xxx",
        ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
        ecx = "xRRR:RRRR:HRRR:RRRR:RRHR:RRHR:RRRR:0RRH",
        edx = "RRR0:RHHH:HRRR:HRHH:HHHH:HR1H:HHHH:HHHH",
      },
      (vim.host.CpuldInfo) {
        dynamicType = <unset>,
        level = 0,
        vendor = <unset>,
        eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
        ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
      }
    ]
  }
}

```

```

    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = "amd",
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = <unset>,
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "xRRR:R0HR:HR0H:H0RR:00HR:R0H0:0000:00HH",
    edx = "0000:HHHH:HHHR:HHHH:HHHH:HR1H:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = "amd",
    eax = "RRRR:xxxx:xxxx:xxxx:RRxx:xxxx:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RR00:R00H:HHH0:H00H",
    edx = "HHTR:H0HH:HHRH:RRHH:HHHH:xR1H:HHHH:HHHH",
  }
],
supportsWakeOnLan = true,
supportsVMI = false,
supportsMemoryHotAdd = true,
supportsCpuHotAdd = false,
supportsCpuHotRemove = false,
},
(vim.vm.GuestOsDescriptor) {
  dynamicType = <unset>,
  id = "winNetDatacenterGuest",
  family = "windowsGuest",
  fullName = "Microsoft Windows Server 2003, Datacenter Edition (32-bit)",
  supportedMaxCPUs = 8,
  supportedMinMemMB = 512,
  supportedMaxMemMB = 131072,
  recommendedMemMB = 1024,
  recommendedColorDepth = 8,
  supportedDiskControllerList = (vmodl.TypeName) [
    "vim.vm.device.ParaVirtualSCSIController",
    "vim.vm.device.VirtualBusLogicController",
    "vim.vm.device.VirtualLsiLogicController",
    "vim.vm.device.VirtualLsiLogicSASController",
    "vim.vm.device.VirtualIDEController"
  ],
},

```

```

recommendedSCSIController = "vim.vm.device.VirtualLsiLogicController",
recommendedDiskController = "vim.vm.device.VirtualLsiLogicController",
supportedNumDisks = 16,
recommendedDiskSizeMB = 8192,
supportedEthernetCard = (vmobl.TypeName) [
  "vim.vm.device.VirtualE1000",
  "vim.vm.device.VirtualVmxnet2",
  "vim.vm.device.VirtualVmxnet3"
],
recommendedEthernetCard = "vim.vm.device.VirtualE1000",
supportsSlaveDisk = true,
cpuFeatureMask = (vim.host.CpuldInfo) [
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = <unset>,
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRx",
    edx = "RRxR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = "amd",
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ecx = "xRRR:RRRR:HRRR:RRRR:RRxR:RRHR:RRRR:0RRH",
    edx = "RRR0:RHHH:HRRR:HRHH:H0HH:HR1H:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = <unset>,
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = "amd",
    eax = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = <unset>,
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx",
  }
]

```

```

    ecx = "xRRR:R0HR:HR0H:H0RR:00xR:R0H0:0000:00HH",
    edx = "0000:HHHH:HHHR:HHHH:H0HH:HR1H:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = -2147483647,
    vendor = "amd",
    eax = "RRRR:xxx:xxx:xxx:RRxx:xxx:xxx:xxx",
    ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ecx = "RRRR:RRRR:RRRR:RRRR:RR00:R00H:HHH0:H00x",
    edx = "HHxR:H0HH:HHRH:RRHH:H0HH:xR1H:HHHH:HHHH",
  }
],
supportsWakeOnLan = true,
supportsVMI = false,
supportsMemoryHotAdd = true,
supportsCpuHotAdd = false,
supportsCpuHotRemove = false,
},
(vim.vm.GuestOsDescriptor) {
  dynamicType = <unset>,
  id = "winNetDatacenter64Guest",
  family = "windowsGuest",
  fullName = "Microsoft Windows Server 2003, Datacenter Edition (64-bit)",
  supportedMaxCPUs = 8,
  supportedMinMemMB = 512,
  supportedMaxMemMB = 261120,
  recommendedMemMB = 1024,
  recommendedColorDepth = 8,
  supportedDiskControllerList = (vmodl.TypeName) [
    "vim.vm.device.ParaVirtualSCSIController",
    "vim.vm.device.VirtualLsiLogicController",
    "vim.vm.device.VirtualLsiLogicSASController",
    "vim.vm.device.VirtualIDEController"
  ],
  recommendedSCSIController = "vim.vm.device.VirtualLsiLogicController",
  recommendedDiskController = "vim.vm.device.VirtualLsiLogicController",
  supportedNumDisks = 16,
  recommendedDiskSizeMB = 8192,
  supportedEthernetCard = (vmodl.TypeName) [
    "vim.vm.device.VirtualE1000",
    "vim.vm.device.VirtualVmxnet2",
    "vim.vm.device.VirtualVmxnet3"
  ],
  recommendedEthernetCard = "vim.vm.device.VirtualE1000",
  supportsSlaveDisk = true,
  cpuFeatureMask = (vim.host.CpuldInfo) [
    (vim.host.CpuldInfo) {
      dynamicType = <unset>,
      level = -2147483647,
      vendor = <unset>,
      eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
      ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
      ecx = "RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRR:RRRH",
    }
  ]
}

```

```

    edx = "RRTR:H0RR:RRRH:RRRR:RRRR:xRRR:RRRR:RRRR",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,
    vendor = "amd",
    eax = "RRRR:HHHH:HHHH:xxxx:RRxx:HHHH:xxxx:xxxx",
    ebx = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ecx = "xRRR:RRRR:RRRR:RRRR:RRHR:RRHR:RRRR:0RRH",
    edx = "RRR0:RHHH:RRRR:RRHH:HHHH:HR1H:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = <unset>,
    eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 0,
    vendor = "amd",
    eax = "xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx",
    ebx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    ecx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
    edx = "HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH",
  },
  (vim.host.CpuldInfo) {
    dynamicType = <unset>,
    level = 1,

```

...

vmsvc/get.datastores

```
~ # vim-cmd vmsvc/get.datastores 80
```

```

name      NFS
url       /vmfs/volumes/1c445c7d-fd56af7a
capacity  250056011776
freeSpace 63673634816
accessible 1
type      NFS
multipleHostAccess <unset>

```

vmsvc/get.disabledmethods

~ # vim-cmd vmsvc/get.disabledmethods 80

Disabled methods:

```
(vmodl.MethodName) [
  "vim.ExtensibleManagedObject.setCustomValue",
  "vim.ManagedEntity.destroy",
  "vim.ManagedEntity.addTag",
  "vim.ManagedEntity.removeTag",
  "vim.VirtualMachine.refreshStorageInfo",
  "vim.VirtualMachine.retrieveBackupAgent",
  "vim.VirtualMachine.revertToCurrentSnapshot",
  "vim.VirtualMachine.removeAllSnapshots",
  "vim.VirtualMachine.upgradeVirtualHardware",
  "vim.VirtualMachine.extractOvfEnvironment",
  "vim.VirtualMachine.powerOn",
  "vim.VirtualMachine.answer",
  "vim.VirtualMachine.customize",
  "vim.VirtualMachine.checkCustomizationSpec",
  "vim.VirtualMachine.migrate",
  "vim.VirtualMachine.relocate",
  "vim.VirtualMachine.clone",
  "vim.VirtualMachine.exportVm",
  "vim.VirtualMachine.markAsTemplate",
  "vim.VirtualMachine.markAsVirtualMachine",
  "vim.VirtualMachine.unregister",
  "vim.VirtualMachine.resetGuestInformation",
  "vim.VirtualMachine.unmountToolsInstaller",
  "vim.VirtualMachine.defragmentAllDisks",
  "vim.VirtualMachine.createSecondary",
  "vim.VirtualMachine.turnOffFaultTolerance",
  "vim.VirtualMachine.makePrimary",
  "vim.VirtualMachine.terminateFaultTolerantVM",
  "vim.VirtualMachine.disableSecondary",
  "vim.VirtualMachine.enableSecondary",
  "vim.VirtualMachine.setDisplayTopology",
  "vim.VirtualMachine.stopRecording",
  "vim.VirtualMachine.startReplaying",
  "vim.VirtualMachine.stopReplaying",
  "vim.VirtualMachine.promoteDisks",
  "vim.VirtualMachine.queryChangedDiskAreas",
  "vim.VirtualMachine.queryUnownedFiles"
]
```

vmsvc/get.environment

~ # vim-cmd vmsvc/get.environment 80

```
(vim.vm.ConfigTarget) {
  dynamicType = <unset>,
  numCpus = 2,
  numCpuCores = 2,
  numNumaNodes = 1,
  datastore = (vim.vm.DatastoreInfo) [
    (vim.vm.DatastoreInfo) {
      dynamicType = <unset>,
      name = "Local-1",
      datastore = (vim.Datastore.Summary) {
        dynamicType = <unset>,
        datastore = 'vim.Datastore:4c40d759-46642d80-9a35-0017312598e5',
        name = "Local-1",
        url = "/vmfs/volumes/4c40d759-46642d80-9a35-0017312598e5",
        capacity = 494726545408,
        freeSpace = 494136197120,
        uncommitted = 0,
        accessible = true,
        multipleHostAccess = <unset>,
        type = "VMFS",
      },
      capability = (vim.Datastore.Capability) {
        dynamicType = <unset>,
        directoryHierarchySupported = true,
        rawDiskMappingsSupported = true,
        perFileThinProvisioningSupported = true,
      },
      maxFileSize = 274877906944,
      mode = "readWrite",
    },
    (vim.vm.DatastoreInfo) {
      dynamicType = <unset>,
      name = "Local-2",
      datastore = (vim.Datastore.Summary) {
        dynamicType = <unset>,
        datastore = 'vim.Datastore:4bb06602-f1cec4f0-58a9-0017312598e5',
        name = "Local-2",
        url = "/vmfs/volumes/4bb06602-f1cec4f0-58a9-0017312598e5",
        capacity = 499826819072,
        freeSpace = 252800139264,
        uncommitted = 41173334016,
        accessible = true,
        multipleHostAccess = <unset>,
        type = "VMFS",
      },
      capability = (vim.Datastore.Capability) {
        dynamicType = <unset>,
        directoryHierarchySupported = true,
        rawDiskMappingsSupported = true,
        perFileThinProvisioningSupported = true,
      },
    },
  ],
}
```

```

    },
    maxFileSize = 274877906944,
    mode = "readWrite",
  },
  (vim.vm.DatastoreInfo) {
    dynamicType = <unset>,
    name = "ISO",
    datastore = (vim.Datastore.Summary) {
      dynamicType = <unset>,
      datastore = 'vim.Datastore:192.168.9.222:/media/dwl/public/ISO_Library',
      name = "ISO",
      url = "/vmfs/volumes/be455b13-e1a6c8e3",
      capacity = 250056011776,
      freeSpace = 63673634816,
      uncommitted = 0,
      accessible = true,
      multipleHostAccess = <unset>,
      type = "NFS",
    },
    capability = (vim.Datastore.Capability) {
      dynamicType = <unset>,
      directoryHierarchySupported = true,
      rawDiskMappingsSupported = false,
      perFileThinProvisioningSupported = true,
    },
    maxFileSize = 9223372036854775807,
    mode = "readWrite",
  },
  (vim.vm.DatastoreInfo) {
    dynamicType = <unset>,
    name = "NFS",
    datastore = (vim.Datastore.Summary) {
      dynamicType = <unset>,
      datastore = 'vim.Datastore:192.168.9.222:/media/dwl/public/NFS',
      name = "NFS",
      url = "/vmfs/volumes/1c445c7d-fd56af7a",
      capacity = 250056011776,
      freeSpace = 63673634816,
      uncommitted = 0,
      accessible = true,
      multipleHostAccess = <unset>,
      type = "NFS",
    },
    capability = (vim.Datastore.Capability) {
      dynamicType = <unset>,
      directoryHierarchySupported = true,
      rawDiskMappingsSupported = false,
      perFileThinProvisioningSupported = true,
    },
    maxFileSize = 9223372036854775807,
    mode = "readWrite",
  }
},
network = (vim.vm.NetworkInfo) [

```

```

(vim.vm.NetworkInfo) {
  dynamicType = <unset>,
  name = "VM Network",
  network = (vim.Network.Summary) {
    dynamicType = <unset>,
    network = 'vim.Network:HaNetwork-VM Network',
    name = "VM Network",
    accessible = true,
    ipPoolName = "",
  },
}
],
maxMemMBOptimalPerf = 8187,
resourcePool = (vim.ResourcePool.RuntimeInfo) {
  dynamicType = <unset>,
  memory = (vim.ResourcePool.ResourceUsage) {
    dynamicType = <unset>,
    reservationUsed = 664797184,
    reservationUsedForVm = 664797184,
    unreservedForPool = 5769265152,
    unreservedForVm = 5769265152,
    overallUsage = 389099520,
    maxUsage = 6434062336,
  },
  cpu = (vim.ResourcePool.ResourceUsage) {
    dynamicType = <unset>,
    reservationUsed = 0,
    reservationUsedForVm = 0,
    unreservedForPool = 4977,
    unreservedForVm = 4977,
    overallUsage = 116,
    maxUsage = 4977,
  },
  overallStatus = "green",
},
autoVmotion = <unset>,
}

```

vmsvc/get.filelayout

~ # vim-cmd vmsvc/get.filelayout 80

```

(vim.vm.FileInfo) {
  dynamicType = <unset>,
  vmPathName = "[NFS] XP-1/XP-1.vmx",
  snapshotDirectory = "[NFS] XP-1",
  suspendDirectory = "[NFS] XP-1",
  logDirectory = "[NFS] XP-1",
}
(vim.vm.FileLayout) {
  dynamicType = <unset>,
  configFile = (string) [
    "XP-1-ae53b06d.vms",

```

```

"XP-1.vmsd",
"XP-1.nvram",
"XP-1.vmx"
],
logFile = (string) [
"vmware-32.log",
"vmware-33.log",
"vmware-34.log",
"vmware-35.log",
"vmware-36.log",
"vmware-37.log",
"vmware.log"
],
disk = (vim.vm.FileLayout.DiskLayout) [
(vim.vm.FileLayout.DiskLayout) {
dynamicType = <unset>,
key = 2000,
diskFile = (string) [
"[NFS] XP-1/XP-1_1.vmdk"
],
},
(vim.vm.FileLayout.DiskLayout) {
dynamicType = <unset>,
key = 3000,
diskFile = (string) [
"[NFS] XP-1/XP-1.vmdk"
],
}
],
swapFile = "[NFS] XP-1/XP-1-4895505e.vswp",
}

```

vmsvc/get.guest

~ # vim-cmd vmsvc/get.guest 80

Guest information:

```

(vim.vm.GuestInfo) {
dynamicType = <unset>,
toolsStatus = "toolsOk",
toolsVersionStatus = "guestToolsCurrent",
toolsRunningStatus = "guestToolsRunning",
toolsVersion = "8194",
guestId = "winXPProGuest",
guestFamily = "windowsGuest",
guestFullName = "Microsoft Windows XP Professional (32-bit)",
hostName = "XP-1",
ipAddress = "0.0.0.0",
net = (vim.vm.GuestInfo.NicInfo) [
(vim.vm.GuestInfo.NicInfo) {
dynamicType = <unset>,
network = "VM Network",
}
]
}

```

```

    macAddress = "00:50:56:85:00:0a",
    connected = false,
    deviceConfigId = 4001,
  },
  (vim.vm.GuestInfo.NicInfo) {
    dynamicType = <unset>,
    network = "VM Network",
    macAddress = "00:50:56:85:00:09",
    connected = false,
    deviceConfigId = 4000,
  }
],
disk = (vim.vm.GuestInfo.DiskInfo) [
  (vim.vm.GuestInfo.DiskInfo) {
    dynamicType = <unset>,
    diskPath = "C:\",
    capacity = 8578932736,
    freeSpace = 3082121216,
  },
  (vim.vm.GuestInfo.DiskInfo) {
    dynamicType = <unset>,
    diskPath = "E:\",
    capacity = 8371712,
    freeSpace = 5800960,
  }
],
screen = (vim.vm.GuestInfo.ScreenInfo) {
  dynamicType = <unset>,
  width = 800,
  height = 600,
},
guestState = "running",
}

```

vmsvc/get.guestheartbeatStatus

```
~ # vim-cmd vmsvc/get.guestheartbeatStatus 80
```

green

vmsvc/get.managedentitystatus

Show the status for the virtual machine.

```
~ # vim-cmd vmsvc/get.managedentitystatus 80
```

green

vmsvc/get.networks

~ # vim-cmd vmsvc/get.networks 80

Networks:

```
(vim.Network.Summary) {
  dynamicType = <unset>,
  network = 'vim.Network:HaNetwork-VM Network',
  name = "VM Network",
  accessible = true,
  ipPoolName = "",
}
```

vmsvc/get.runtime

~ # vim-cmd vmsvc/get.runtime 80

Runtime information

```
(vim.vm.RuntimeInfo) {
  dynamicType = <unset>,
  host = 'vim.HostSystem:ha-host',
  connectionState = "connected",
  powerState = "poweredOn",
  faultToleranceState = "notConfigured",
  toolsInstallerMounted = false,
  suspendTime = <unset>,
  bootTime = "2010-08-20T10:39:09.976768Z",
  suspendInterval = 0,
  question = (vim.vm.QuestionInfo) null,
  memoryOverhead = 127344640,
  maxCpuUsage = 3211,
  maxMemoryUsage = 512,
  numMksConnections = 0,
  recordReplayState = "inactive",
  cleanPowerOff = <unset>,
  needSecondaryReason = <unset>,
}
```

vmsvc/get.snapshotinfo

~ # vim-cmd vmsvc/get.snapshotinfo 80

```
(vim.vm.SnapshotInfo) {
  dynamicType = <unset>,
  currentSnapshot = 'vim.vm.Snapshot:80-snapshot-5',
  rootSnapshotList = (vim.vm.SnapshotTree) [
    (vim.vm.SnapshotTree) {
      dynamicType = <unset>,
      snapshot = 'vim.vm.Snapshot:80-snapshot-4',
      vm = 'vim.VirtualMachine:80',
      name = "sn1",
      description = "",
      id = 4,
    }
  ]
}
```

```

createTime = "2010-08-20T11:57:43.598919Z",
state = "poweredOff",
quiesced = false,
backupManifest = <unset>,
childSnapshotList = (vim.vm.SnapshotTree) [
  (vim.vm.SnapshotTree) {
    dynamicType = <unset>,
    snapshot = 'vim.vm.Snapshot:80-snapshot-5',
    vm = 'vim.VirtualMachine:80',
    name = "sn2",
    description = "",
    id = 5,
    createTime = "2010-08-20T11:57:52.36261Z",
    state = "poweredOff",
    quiesced = false,
    backupManifest = <unset>,
    replaySupported = false,
  }
],
replaySupported = false,
}
],
}
}

```

vmsvc/get.summary

~ # vim-cmd vmsvc/get.summary 80

Listsummary:

```

(vim.vm.Summary) {
  dynamicType = <unset>,
  vm = 'vim.VirtualMachine:80',
  runtime = (vim.vm.RuntimeInfo) {
    dynamicType = <unset>,
    host = 'vim.HostSystem:ha-host',
    connectionState = "connected",
    powerState = "poweredOn",
    faultToleranceState = "notConfigured",
    toolsInstallerMounted = false,
    suspendTime = <unset>,
    bootTime = "2010-08-20T10:39:09.976768Z",
    suspendInterval = 0,
    question = (vim.vm.QuestionInfo) null,
    memoryOverhead = 127344640,
    maxCpuUsage = 3211,
    maxMemoryUsage = 512,
    numMksConnections = 0,
    recordReplayState = "inactive",
    cleanPowerOff = <unset>,
    needSecondaryReason = <unset>,
  },
  guest = (vim.vm.Summary.GuestSummary) {

```

```

dynamicType = <unset>,
guestId = "winXPProGuest",
guestFullName = "Microsoft Windows XP Professional (32-bit)",
toolsStatus = "toolsOK",
toolsVersionStatus = "guestToolsCurrent",
toolsRunningStatus = "guestToolsRunning",
hostName = "XP-1",
ipAddress = "0.0.0.0",
},
config = (vim.vm.Summary.ConfigSummary) {
dynamicType = <unset>,
name = "XP-1",
template = false,
vmPathName = "[NFS] XP-1/XP-1.vmx",
memorySizeMB = 512,
cpuReservation = <unset>,
memoryReservation = <unset>,
numCpu = 1,
numEthernetCards = 2,
numVirtualDisks = 2,
uuid = "42054c7a-8d9d-c36e-44f8-05855b5aec3e",
instanceUuid = "5005dd1e-61c9-d323-216d-30eed4585753",
guestId = "winXPProGuest",
guestFullName = "Microsoft Windows XP Professional (32-bit)",
annotation = "FEGFE",
product = (vim.vApp.ProductInfo) null,
installBootRequired = <unset>,
ftInfo = (vim.vm.FaultToleranceConfigInfo) null,
},
storage = (vim.vm.Summary.StorageSummary) {
dynamicType = <unset>,
committed = 8840799324,
uncommitted = 868089856,
unshared = 8840799324,
timestamp = "2010-08-20T10:37:24.993798Z",
},
quickStats = (vim.vm.Summary.QuickStats) {
dynamicType = <unset>,
overallCpuUsage = 109,
overallCpuDemand = <unset>,
guestMemoryUsage = 122,
hostMemoryUsage = 362,
guestHeartbeatStatus = "green",
distributedCpuEntitlement = <unset>,
distributedMemoryEntitlement = <unset>,
staticCpuEntitlement = <unset>,
staticMemoryEntitlement = <unset>,
privateMemory = <unset>,
sharedMemory = <unset>,
swappedMemory = <unset>,
balloonedMemory = <unset>,
consumedOverheadMemory = <unset>,
ftLogBandwidth = <unset>,
ftSecondaryLatency = <unset>,

```

```

    ftLatencyStatus = <unset>,
  },
  overallStatus = "green",
}

```

* **vmsvc/get.tasklist**

* **vmsvc/gethostconstraints**

* **vmsvc/login**

* **vmsvc/logout**

* **vmsvc/message**

Replies to the current pending question on the vm.

vmsvc/power.getstate

~ # vim-cmd vmsvc/power.getstate 80




Retrieved runtime info

Powered on

vmsvc/power.hibernate

~ # vim-cmd vmsvc/power.hibernate 80




~ #

Recent Tasks		
Name	Target	Status
 Initiated guest standby	 XP-1	 Completed

vmsvc/power.off

~ # vim-cmd vmsvc/power.off 80




Powering off VM:

Recent Tasks		
Name	Target	Status
 Power Off virtual machine	 XP-1	 Completed

vmsvc/power.reboot

~ # vim-cmd vmsvc/power.reboot 80

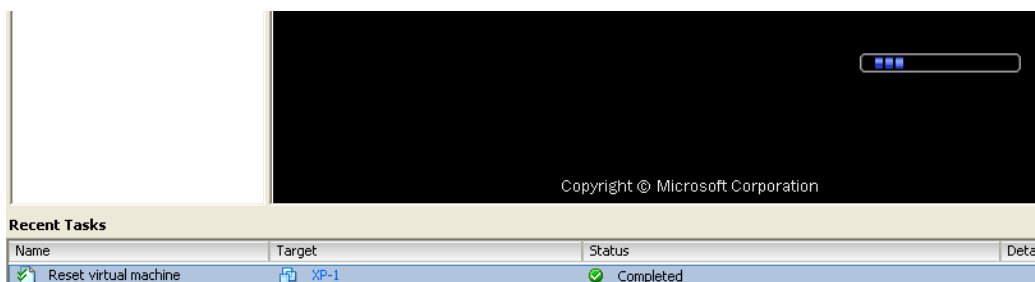
~ #

Recent Tasks		
Name	Target	Status
 Reboot guest	 XP-1	 Completed

vmsvc/power.reset

~ # vim-cmd vmsvc/power.reset 80

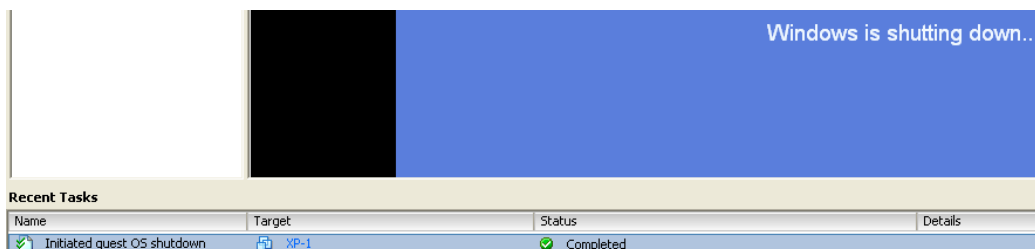
Reset VM:



vmsvc/power.shutdown

~ # vim-cmd vmsvc/power.shutdown 80

~ #



vmsvc/power.suspend

~ # vim-cmd vmsvc/power.suspend 80

Suspending VM:

Recent Tasks			
Name	Target	Status	Details
Suspend virtual machine	XP-1	Completed	

* vmsvc/power.suspendResume

* vmsvc/queryftcompat

* vmsvc/reload

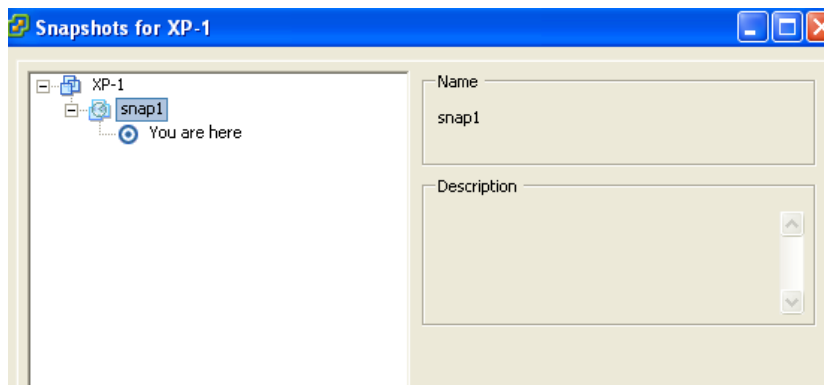
* vmsvc/setscreenres

vmsvc/snapshot.create

Usage: snapshot.create vmid [snapshotName] [snapshotDescription] [includeMemory] [quiesced]

```
~ # vim-cmd vmsvc/snapshot.create 80 snap1
```

Create Snapshot:



* vmsvc/snapshot.dumpoption

vmsvc/snapshot.get

```
~ # vim-cmd vmsvc/snapshot.get 80
```

Get Snapshot:

```
|--ROOT
```

```
--Snapshot Name      : snap1
```

```
--Snapshot Description :
```

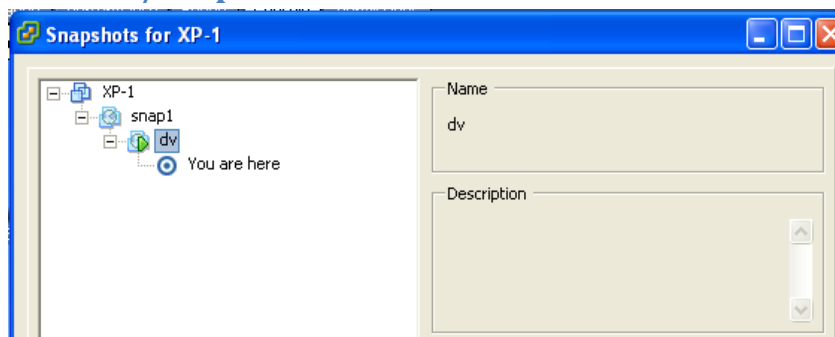
```
--Snapshot Created On : 8/20/2010 11:32:44
```

```
--Snapshot State     : powered off
```

* vmsvc/snapshot.remove

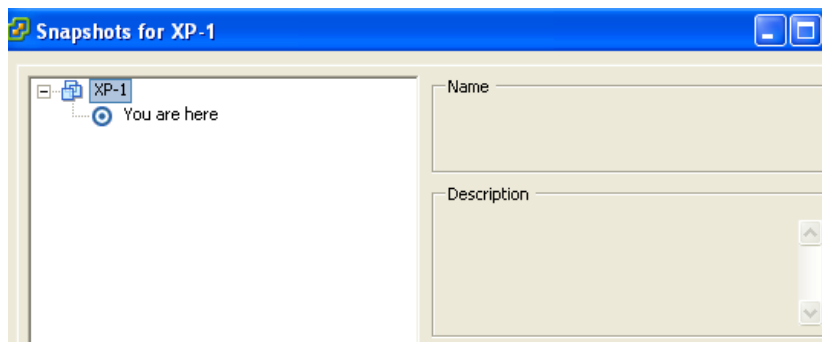
Usage: snapshot.remove vmid [removeChildren] [snapshotLevel] [snapshotIndex]

vmsvc/snapshot.removeall



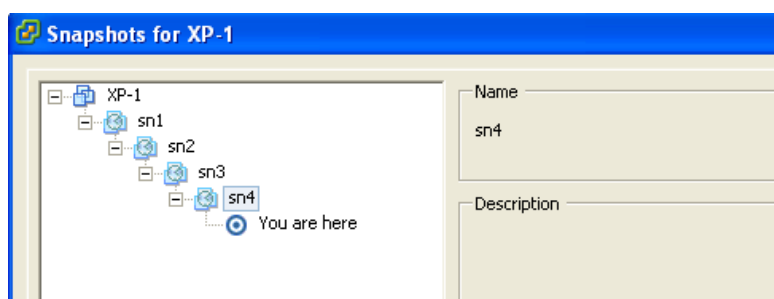
```
~ # vim-cmd vmsvc/snapshot.removeall 80
```

Remove All Snapshots:



vmsvc/snapshot.revert

Usage: snapshot.revert vmid suppressPowerOff [snapshotLevel] [snapshotIndex]



```
~ # vim-cmd vmsvc/snapshot.revert 80 2
```

Remove Snapshot:

```
|-ROOT
```

```
--Snapshot Name    : sn1
```

```
--Snapshot Description :
```

```
--Snapshot Created On : 8/20/2010 11:57:43
```

```
--Snapshot State    : powered off
```

```
--|-CHILD
```

```
----Snapshot Name    : sn2
```

```
----Snapshot Description :
```

```
----Snapshot Created On : 8/20/2010 11:57:52
```

```
----Snapshot State    : powered off
```

```
----|-CHILD
```

```
-----Snapshot Name    : sn3
```

```
-----Snapshot Description :
```

```
-----Snapshot Created On : 8/20/2010 11:59:56
```

```
-----Snapshot State    : powered off
```

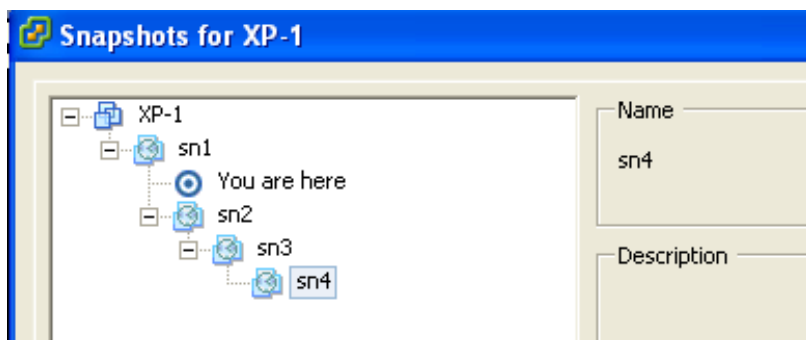
```
-----|-CHILD
```

```
-----Snapshot Name    : sn4
```

```
-----Snapshot Description :
```

```
-----Snapshot Created On : 8/20/2010 12:3:16
```

```
-----Snapshot State    : powered off
```



* **vmsvc/snapshot.setoption**

vmsvc/tools.cancelinstall

~ # vim-cmd vmsvc/tools.cancelinstall 80

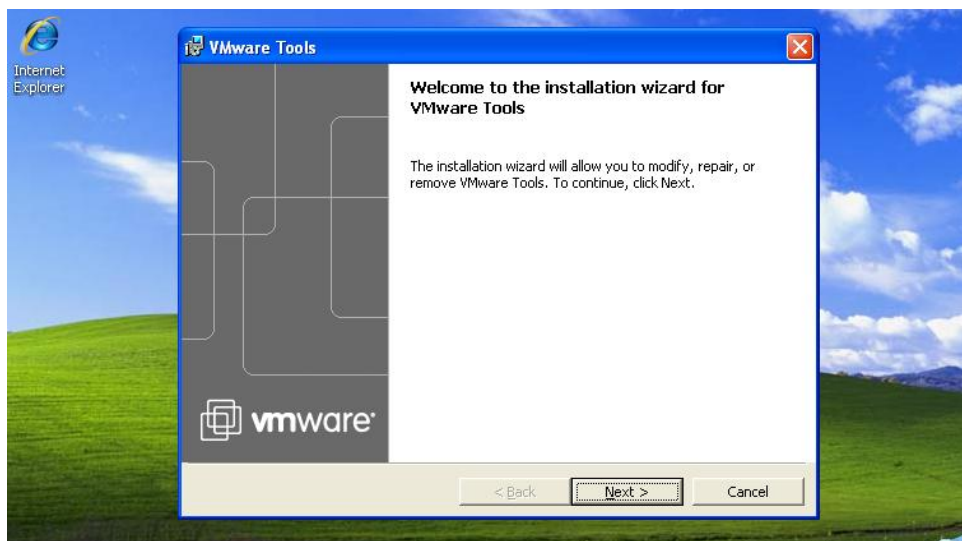
~ #

Recent Tasks		
Name	Target	Status
Unmount tools installer	XP-1	Completed
Initiated VMware Tools Installer...	XP-1	Completed

vmsvc/tools.install

~ # vim-cmd vmsvc/tools.install 80

~ #



* **vmsvc/tools.upgrade**

* **vmsvc/unregister**

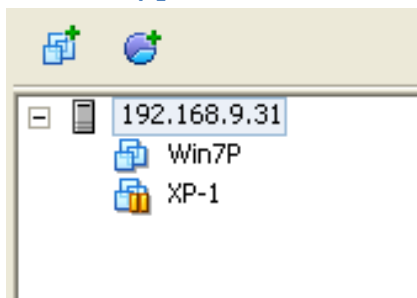
* **vmsvc/upgrade**

vmsvc/getallvms

~ # vim-cmd vmsvc/getallvms

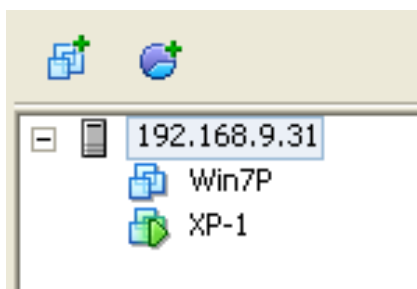
Vmid	Name	File	Guest OS	Version	Annotation
176	ubuntuServer8.04-2	[NFS] ubuntuServer8.04-2/ubuntuServer8.04-2.vmx	ubuntuGuest	vmx-07	ubuntu Server 8.04 LTS x86
192	VM5	[NFS] VM5/VM5.vmx	winXPProGuest	vmx-07	
208	VM1	[NFS] VM1/VM1.vmx	winXPProGuest	vmx-07	
224	ubuntuServer8.04-1	[NFS] ubuntuServer8.04-1/ubuntuServer8.04-1.vmx	ubuntuGuest	vmx-07	ubuntu Server 8.04 LTS x86
240	New Virtual Machine	[of-3] New Virtual Machine/New Virtual Machine.vmx	windows7Server64Guest	vmx-07	
256	VM2	[NFS] VM2/VM2.vmx	winXPProGuest	vmx-07	

vmsvc/power.on



~ # vim-cmd vmsvc/power.on 80

Powering on VM:



internalsvc/

internalsvc/perfcount/

- * [internalsvc/perfcount/enumerate](#)
- * [internalsvc/perfcount/query_execute](#)
- * [internalsvc/perfcount/query_list](#)
- * [internalsvc/perfcount/query_regex](#)
- * [internalsvc/perfcount/query_destroy](#)
- * [internalsvc/perfcount /query_info](#)
- * [internalsvc/perfcount/query_names](#)

internalsvc/perfcount /scoreboard

~ # vim-cmd internalsvc/perfcount/scoreboard|less

```
(2,0) => Vim::Host::FileSystemMountInfo - 0
(2,1) => Vim::Dvs::ProductSpec - 0
(9,0) => Vim::Host::StorageFactoryImpl - 1
(11,0) => Vmomi::DataArray<Vim::Host::ScsiLun::Descriptor> - 7
(17,0) => Vmomi::DataArray<Vim::Datastore::HostMount> - 0
(25,0) => Hostsvc::AdvancedOptionsVmk - 1
(25,1) => Vmsvc::LicFeatures - 2
(29,0) => Vmomi::DynamicProperty - 0
(32,0) => Bklistsvc::DiskManagerImpl - 1
(43,0) => Vim::Event::EventFilterSpec - 1
(44,0) => Vmacore::Xml::TextNodeImpl - 0
(44,1) => Vmomi::DataArray<Vim::HistoricalInterval> - 0
(45,0) => HostdCommon::VobMsg - 0
(47,0) => Vim::Vm::Device::VirtualFloppyOption::RemoteDeviceBackingOption - 2
(48,0) => Vmomi::Primitive<Vim::VirtualMachine::PowerState> - 0
(49,0) => Vmacore::Ssl::SSLServerSocketImpl::AsyncOp - 1
(51,0) => Vmomi::DataArray<Vmomi::KeyAnyValue> - 2
(52,0) => Hostsvc::OptionManagerImpl - 1
(52,1) => Vim::HostServiceTicket - 0
(54,0) => Cimsvc::WrapperInstanceEnumeration - 0
(56,0) => Hostsvc::FirmwareSystemImpl - 1
(56,1) => Vmsvc::VmConfig - 6
(58,0) => Vcsvc::HaAgentManager - 1
(60,0) => Vim::Vm::Device::VirtualVMCIDevice - 7
(70,0) => Vmomi::DataArray<Vim::Host::VirtualSwitch> - 1
(71,0) => Vmomi::DataArray<Vim::Option::OptionValue> - 6
(73,0) => HostdCommon::AdvancedOptionsRegistryImpl - 1
(74,0) => Vim::Vm::Device::VirtualKeyboard - 9
(76,0) => VirtualMachineNvramSearchResultImpl - 0
(78,0) => FolderSearchResultImpl - 0
(79,0) => Vmacore::Service::ApplImpl - 1
(83,0) => Vim::Host::VirtualNic - 3
(84,0) => Vim::Vm::Device::VirtualDevice::ConnectInfo - 20
```

(85,0) => Directorysvc::DirectorysvcPlugin - 1
(89,0) => Vmomi::DataArray<Vm::CustomFieldsManager::Value> - 0
(90,0) => Vmcore::Authorize::UserDataInfoPosix - 5
(91,0) => Solo::VmwareCLIImpl - 1
(94,0) => ISOImageItemizedSearch - 0
(97,0) => Solo::PatchUploadDocumentHandler - 1
(99,0) => Vmomi::DataArray<Vmomi::Core::PropertyCollector::FilterSpec> - 0
(101,0) => Vmomi::DataArray<Vmomi::DynamicProperty> - 1882
(101,1) => Vmomi::DataArray<Vm::Host::SystemIdentificationInfo> - 1
(101,2) => Vmomi::DataArray<Vmomi::Core::PropertyCollector::MissingProperty> - 0
(106,0) => Vmomi::DataArray<Vm::Internal::DynamicTypeManager::MethodInfo> - 20
(108,0) => Vmomi::DataArray<Vm::Host::PortGroup::Port> - 8

...

*** internalsvc/logout**

*** internalsvc/refresh_network**

*** internalsvc/access_address**

*** internalsvc/redirect_stderr**

*** internalsvc/refresh_pnic**

*** internalsvc/cold_quit**

*** internalsvc/redirect_stdout**

*** internalsvc/set_exception_variable**

*** internalsvc/connect**

*** internalsvc/refresh**

*** internalsvc/set_log_level**

*** internalsvc/host_mode_lock**

*** internalsvc/refresh_consolenic**

*** internalsvc/shutdown**

*** internalsvc/login**

*** internalsvc/refresh_datastores**

*** internalsvc/throw_exception**

internalsvc/loglist

~ # vim-cmd internalsvc/loglist

```
(hostd.LogsvclManager.LogInfo) [
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "ActiveDirectoryAuthentication",
    logLevel = "verbose",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "AgentMgr",
    logLevel = "verbose",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "AnnotationProvider",
    logLevel = "verbose",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "App",
    logLevel = "info",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "AppConfigOptionsProvider(Config.HostAgent.)",
    logLevel = "verbose",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "AuthenticationManager",
    logLevel = "verbose",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "BlklistsvclPlugin",
    logLevel = "verbose",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "Cimsvcl",
    logLevel = "verbose",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "ComputeResource",
    logLevel = "verbose",
  },
  (hostd.LogsvclManager.LogInfo) {
    dynamicType = <unset>,
    name = "Datastore Folder",
    logLevel = "verbose",
  },
],
```

```
(hostd.LogsvclManager.LogInfo) {
  dynamicType = <unset>,
  name = "DatastoreBrowser",
  logLevel = "verbose",
},
(hostd.LogsvclManager.LogInfo) {
  dynamicType = <unset>,
  name = "DatastoreBrowser",
  logLevel = "verbose",
},
(hostd.LogsvclManager.LogInfo) {
  dynamicType = <unset>,
  name = "DatastoreBrowser",
  logLevel = "verbose",
},
(hostd.LogsvclManager.LogInfo) {
  dynamicType = <unset>,
  name = "DatastoreBrowser",
  logLevel = "verbose",
},
(hostd.LogsvclManager.LogInfo) {
  dynamicType = <unset>,
  name = "DatastoreBrowser",
  logLevel = "verbose",
},
(hostd.LogsvclManager.LogInfo) {
  dynamicType = <unset>,
  name = "DatastoreBrowser",
  logLevel = "verbose",
},
(hostd.LogsvclManager.LogInfo) {
  dynamicType = <unset>,
  name = "DatastoreBrowser",
  logLevel = "verbose",
},
(hostd.LogsvclManager.LogInfo) {
  dynamicType = <unset>,
  name = "DatastoreBrowser",
  logLevel = "verbose",
},
...

```

*** internalsvc/refresh_gateway**

*** internalsvc/use_fds**

solo/

* [solo/connect](#)

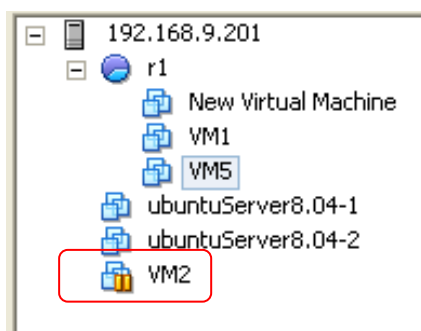
* [solo/login](#)

* [solo/querycgopt](#)

[solo/registervm](#)

~ # vim-cmd solo/registervm /vmfs/volumes/NFS/VM2/VM2.vmx

256



[solo/environment](#)

~ # vim-cmd solo/environment

```
(vim.vm.ConfigTarget) {
  dynamicType = <unset>,
  numCpus = 1,
  numCpuCores = 1,
  numNumaNodes = 1,
  datastore = (vim.vm.DatastoreInfo) [
    (vim.vm.DatastoreInfo) {
      dynamicType = <unset>,
      name = "of-1",
      datastore = (vim.Datastore.Summary) {
        dynamicType = <unset>,
        datastore = 'vim.Datastore:4c654f65-75b87896-ef8e-000c29829b75',
        name = "of-1",
        url = "/vmfs/volumes/4c654f65-75b87896-ef8e-000c29829b75",
        capacity = 3489660928,
        freeSpace = 3195011072,
        uncommitted = 0,
        accessible = true,
        multipleHostAccess = <unset>,
        type = "VMFS",
      },
    },
  ],
  capability = (vim.Datastore.Capability) {
    dynamicType = <unset>,
    directoryHierarchySupported = true,
    rawDiskMappingsSupported = true,
    perFileThinProvisioningSupported = true,
  },
}
```

```

    storageORMSupported = true,
  },
  maxFileSize = 274877906944,
  mode = "readWrite",
},
(vim.vm.DatastoreInfo) {
  dynamicType = <unset>,
  name = "datastore1",
  datastore = (vim.Datastore.Summary) {
    dynamicType = <unset>,
    datastore = 'vim.Datastore:4c5cc8e0-80fe1968-98ec-000c29829b7f',
    name = "datastore1",
    url = "/vmfs/volumes/4c5cc8e0-80fe1968-98ec-000c29829b7f",
    capacity = 4294967296,
    freeSpace = 3967811584,
    uncommitted = 0,
    accessible = true,
    multipleHostAccess = <unset>,
    type = "VMFS",
  },
...

```

* solo/logout

solo/querycfgoptdesc

~ # vim-cmd solo/querycfgoptdesc

```

(vim.vm.ConfigOptionDescriptor) [
  (vim.vm.ConfigOptionDescriptor) {
    dynamicType = <unset>,
    key = "vmx-03",
    description = "vm.ConfigOptionDescriptor.vmx-03.summary",
    createSupported = false,
    defaultConfigOption = false,
  },
  (vim.vm.ConfigOptionDescriptor) {
    dynamicType = <unset>,
    key = "vmx-04",
    description = "vm.ConfigOptionDescriptor.vmx-04.summary",
    createSupported = true,
    defaultConfigOption = false,
  },
  (vim.vm.ConfigOptionDescriptor) {
    dynamicType = <unset>,
    key = "vmx-07",
    description = "vm.ConfigOptionDescriptor.vmx-07.summary",
    createSupported = true,
    defaultConfigOption = true,
  }
]

```

vimsvc/

vimsvc/auth/

* vimsvc/auth/entity_permission_add

* vimsvc/auth/lockdown_mode_enter

vimsvc/auth/role_permissions

~ # vim-cmd vimsvc/auth/role_permissions NoAccess

```
(vim.AuthorizationManager.Permission) []
```

~ # vim-cmd vimsvc/auth/role_permissions ReadOnly

```
(vim.AuthorizationManager.Permission) [
```

```
(vim.AuthorizationManager.Permission) {
```

```
  dynamicType = <unset>,
```

```
  entity = 'vim.Folder:ha-folder-root',
```

```
  principal = "vi-user00",
```

```
  group = false,
```

```
  roleId = -2,
```

```
  propagate = true,
```

```
}
```

```
]
```

* vimsvc/auth/entity_permission_remove

* vimsvc/auth/lockdown_mode_exit

* vimsvc/auth/role_remove

* vimsvc/auth/entity_permissions

vimsvc/auth/permissions

~ # vim-cmd vimsvc/auth/permissions

```
(vim.AuthorizationManager.Permission) [
```

```
(vim.AuthorizationManager.Permission) {
```

```
  dynamicType = <unset>,
```

```
  entity = 'vim.Folder:ha-folder-root',
```

```
  principal = "vi-user00",
```

```
  group = false,
```

```
  roleId = -2,
```

```
  propagate = true,
```

```
},
```

```
(vim.AuthorizationManager.Permission) {
```

```
  dynamicType = <unset>,
```

```
  entity = 'vim.Folder:ha-folder-root',
```

```
  principal = "vi-admin00",
```

```
  group = false,
```

```
  roleId = -1,
```

```
  propagate = true,
```

```
},
```

```

(vim.AuthorizationManager.Permission) {
  dynamicType = <unset>,
  entity = 'vim.Folder:ha-folder-root',
  principal = "vpxuser",
  group = false,
  roleId = -1,
  propagate = true,
},
(vim.AuthorizationManager.Permission) {
  dynamicType = <unset>,
  entity = 'vim.Folder:ha-folder-root',
  principal = "dcui",
  group = false,
  roleId = -1,
  propagate = true,
},
(vim.AuthorizationManager.Permission) {
  dynamicType = <unset>,
  entity = 'vim.Folder:ha-folder-root',
  principal = "root",
  group = false,
  roleId = -1,
  propagate = true,
}
]
vimsvc/auth/roles
~ # vim-cmd vimsvc/auth/roles
(vim.AuthorizationManager.Role) [
(vim.AuthorizationManager.Role) {
  dynamicType = <unset>,
  roleId = -5,
  system = true,
  name = "NoAccess",
  info = (vim.Description) {
    dynamicType = <unset>,
    label = "No access",
    summary = "Used for restricting granted access",
  },
},
(vim.AuthorizationManager.Role) {
  dynamicType = <unset>,
  roleId = -4,
  system = true,
  name = "Anonymous",
  info = (vim.Description) {
    dynamicType = <unset>,
    label = "Anonymous",
    summary = "Not logged-in user (cannot be granted)",
  },
},
privilege = (string) [
  "System.Anonymous"
],
},
(vim.AuthorizationManager.Role) {

```

```

dynamicType = <unset>,
roleId = -3,
system = true,
name = "View",
info = (vim.Description) {
    dynamicType = <unset>,
    label = "View",
    summary = "Visibility access (cannot be granted)",
},
privilege = (string) [
    "System.Anonymous",
    "System.View"
],
},
(vim.AuthorizationManager.Role) {
dynamicType = <unset>,
roleId = -2,
system = true,
info = (vim.Description) {
    dynamicType = <unset>,
    label = "Read-only",
    summary = "See details of objects, but not make changes",
},
privilege = (string) [
    "System.Anonymous",
    "System.Read",
    "System.View"
],
},
(vim.AuthorizationManager.Role) {
dynamicType = <unset>,
roleId = -1,
system = true,
name = "Admin",
info = (vim.Description) {
    dynamicType = <unset>,
    label = "Administrator",
    summary = "Full access rights",
},
privilege = (string) [
    "Alarm.Acknowledge",
    "Alarm.Create",
    "Alarm.Delete",
    "Alarm.DisableActions",
    "Alarm.Edit",
    "Alarm.SetStatus",
    "Authorization.ModifyPermissions",
    "Authorization.ModifyRoles",
    "Authorization.ReassignRolePermissions",
    "DVPortgroup.Create",
    "DVPortgroup.Delete",
    "DVPortgroup.Modify",
    "DVPortgroup.PolicyOp",
    "DVPortgroup.ScopeOp",

```

"DVSwitch.Create",
"DVSwitch.Delete",
"DVSwitch.HostOp",
"DVSwitch.Modify",
"DVSwitch.Move",
"DVSwitch.PolicyOp",
"DVSwitch.PortConfig",
"DVSwitch.PortSetting",
"DVSwitch.ResourceManagement",
"Datacenter.Create",
"Datacenter.Delete",
"Datacenter.IpPoolConfig",
"Datacenter.Move",
"Datacenter.Rename",
"Datastore.AllocateSpace",
"Datastore.Browse",
"Datastore.Config",
"Datastore.Delete",
"Datastore.DeleteFile",
"Datastore.FileManagement",
"Datastore.Move",
"Datastore.Rename",
"Datastore.UpdateVirtualMachineFiles",
"Extension.Register",
"Extension.Unregister",
"Extension.Update",
"Folder.Create",
"Folder.Delete",
"Folder.Move",
"Folder.Rename",
"Global.CancelTask",
"Global.CapacityPlanning",
"Global.Diagnostics",
"Global.DisableMethods",
"Global.EnableMethods",
"Global.GlobalTag",
"Global.Health",
"Global.Licenses",
"Global.LogEvent",
"Global.ManageCustomFields",
"Global.Proxy",
"Global.ScriptAction",
"Global.ServiceManagers",
"Global.SetCustomField",
"Global.Settings",
"Global.SystemTag",
"Global.VCServer",
"Host.Cim.CimInteraction",
"Host.Config.AdvancedConfig",
"Host.Config.AuthenticationStore",
"Host.Config.AutoStart",
"Host.Config.Connection",
"Host.Config.Date Time",
"Host.Config.Firmware",

"Host.Config.Maintenance",
"Host.Config.Memory",
"Host.Config.NetService",
"Host.Config.Network",
"Host.Config.Patch",
"Host.Config.PciPassthru",
"Host.Config.Power",
"Host.Config.Resources",
"Host.Config.Settings",
"Host.Config.Snmp",
"Host.Config.Storage",
"Host.Config.SystemManagement",
"Host.Inventory.AddHostToCluster",
"Host.Inventory.AddStandaloneHost",
"Host.Inventory.CreateCluster",
"Host.Inventory.DeleteCluster",
"Host.Inventory.EditCluster",
"Host.Inventory.MoveCluster",
"Host.Inventory.MoveHost",
"Host.Inventory.RemoveHostFromCluster",
"Host.Inventory.RenameCluster",
"Host.Local.CreateVM",
"Host.Local.DeleteVM",
"Host.Local.InstallAgent",
"Host.Local.ManageUserGroups",
"Host.Local.ReconfigVM",
"Network.Assign",
"Network.Config",
"Network.Delete",
"Network.Move",
"Performance.ModifyIntervals",
"Profile.Clear",
"Profile.Create",
"Profile.Delete",
"Profile.Edit",
"Profile.Export",
"Profile.View",
"Resource.ApplyRecommendation",
"Resource.AssignVAppToPool",
"Resource.AssignVMToPool",
"Resource.ColdMigrate",
"Resource.CreatePool",
"Resource.DeletePool",
"Resource.EditPool",
"Resource.HotMigrate",
"Resource.QueryVMotion",
"Resource.RenamePool",
"ScheduledTask.Create",
"ScheduledTask.Delete",
"ScheduledTask.Edit",
"ScheduledTask.Run",
"Sessions.GlobalMessage",
"Sessions.ImpersonateUser",
"Sessions.TerminateSession",

"Sessions.ValidateSession",
"System.Anonymous",
"System.Read",
"System.View",
"Task.Create",
"Task.Update",
"VApp.ApplicationConfig",
"VApp.AssignResourcePool",
"VApp.AssignVApp",
"VApp.AssignVM",
"VApp.Clone",
"VApp.Create",
"VApp.Delete",
"VApp.Export",
"VApp.ExtractOvfEnvironment",
"VApp.Import",
"VApp.InstanceConfig",
"VApp.Move",
"VApp.PowerOff",
"VApp.PowerOn",
"VApp.Rename",
"VApp.ResourceConfig",
"VApp.Suspend",
"VApp.Unregister",
"VRMPolicy.Query",
"VRMPolicy.Update",
"VirtualMachine.Config.AddExistingDisk",
"VirtualMachine.Config.AddNewDisk",
"VirtualMachine.Config.AddRemoveDevice",
"VirtualMachine.Config.AdvancedConfig",
"VirtualMachine.Config.CPUCount",
"VirtualMachine.Config.ChangeTracking",
"VirtualMachine.Config.DiskExtend",
"VirtualMachine.Config.DiskLease",
"VirtualMachine.Config.EditDevice",
"VirtualMachine.Config.HostUSBDevice",
"VirtualMachine.Config.QueryFTCompatibility",
"VirtualMachine.Config.QueryUnownedFiles",
"VirtualMachine.Config.RawDevice",
"VirtualMachine.Config.ReloadFromPath",
"VirtualMachine.Config.RemoveDisk",
"VirtualMachine.Config.Rename",
"VirtualMachine.Config.ResetGuestInfo",
"VirtualMachine.Config.Resource",
"VirtualMachine.Config.Settings",
"VirtualMachine.Config.SwapPlacement",
"VirtualMachine.Config.Unlock",
"VirtualMachine.Config.UpgradeVirtualHardware",
"VirtualMachine.Interact.AnswerQuestion",
"VirtualMachine.Interact.Backup",
"VirtualMachine.Interact.ConsoleInteract",
"VirtualMachine.Interact.CreateScreenshot",
"VirtualMachine.Interact.CreateSecondary",
"VirtualMachine.Interact.DefragmentAllDisks",

```

"VirtualMachine.Interact.DeviceConnection",
"VirtualMachine.Interact.DisableSecondary",
"VirtualMachine.Interact.EnableSecondary",
"VirtualMachine.Interact.GuestControl",
"VirtualMachine.Interact.MakePrimary",
"VirtualMachine.Interact.PowerOff",
"VirtualMachine.Interact.PowerOn",
"VirtualMachine.Interact.Record",
"VirtualMachine.Interact.Replay",
"VirtualMachine.Interact.Reset",
"VirtualMachine.Interact.SetCDMedia",
"VirtualMachine.Interact.SetFloppyMedia",
"VirtualMachine.Interact.Suspend",
"VirtualMachine.Interact.TerminateFaultTolerantVM",
"VirtualMachine.Interact.ToolsInstall",
"VirtualMachine.Interact.TurnOffFaultTolerance",
"VirtualMachine.Inventory.Create",
"VirtualMachine.Inventory.CreateFromExisting",
"VirtualMachine.Inventory.Delete",
"VirtualMachine.Inventory.Move",
"VirtualMachine.Inventory.Register",
"VirtualMachine.Inventory.Unregister",
"VirtualMachine.Provisioning.Clone",
"VirtualMachine.Provisioning.CloneTemplate",
"VirtualMachine.Provisioning.CreateTemplateFromVM",
"VirtualMachine.Provisioning.Customize",
"VirtualMachine.Provisioning.DeployTemplate",
"VirtualMachine.Interact.ToolsInstall",
"VirtualMachine.Interact.TurnOffFaultTolerance",
"VirtualMachine.Inventory.Create",
"VirtualMachine.Inventory.CreateFromExisting",
"VirtualMachine.Inventory.Delete",
"VirtualMachine.Inventory.Move",
"VirtualMachine.Inventory.Register",
"VirtualMachine.Inventory.Unregister",
"VirtualMachine.Provisioning.Clone",
"VirtualMachine.Provisioning.CloneTemplate",
"VirtualMachine.Provisioning.CreateTemplateFromVM",
"VirtualMachine.Provisioning.Customize",
"VirtualMachine.Provisioning.DeployTemplate",
"VirtualMachine.Provisioning.DiskRandomAccess",
"VirtualMachine.Provisioning.DiskRandomRead",
"VirtualMachine.Provisioning.GetVmFiles",
"VirtualMachine.Provisioning.MarkAsTemplate",
"VirtualMachine.Provisioning.MarkAsVM",
"VirtualMachine.Provisioning.ModifyCustSpecs",
"VirtualMachine.Provisioning.PromoteDisks",
"VirtualMachine.Provisioning.PutVmFiles",
"VirtualMachine.Provisioning.ReadCustSpecs",
"VirtualMachine.State.CreateSnapshot",
"VirtualMachine.State.RemoveSnapshot",
"VirtualMachine.State.RenameSnapshot",
"VirtualMachine.State.RevertToSnapshot"
],

```

```

},
(vim.AuthorizationManager.Role) {
  dynamicType = <unset>,
  roleId = 10,
  system = false,
  name = "Clone of Read-only",
  info = (vim.Description) {
    dynamicType = <unset>,
    label = "Clone of Read-only",
    summary = "Clone of Read-only",
  },
  privilege = (string) [
    "System.Anonymous",
    "System.Read",
    "System.View"
  ],
}
]

```

vimsvc/auth/lockdown_is_enabled

```
~ # vim-cmd vimsvc/auth/lockdown_is_enabled
```

```
false
```

vimsvc/auth/privileges

```
~ # vim-cmd vimsvc/auth/privileges | less
```

```

(vim.AuthorizationManager.Privilege) [
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "System.Anonymous",
    onParent = false,
    name = "Anonymous",
    privGroupName = "System",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "System.View",
    onParent = false,
    name = "View",
    privGroupName = "System",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "System.Read",
    onParent = false,
    name = "Read",
    privGroupName = "System",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.ManageCustomFields",

```

```

    onParent = false,
    name = "ManageCustomFields",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.SetCustomField",
    onParent = false,
    name = "SetCustomField",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.LogEvent",
    onParent = false,
    name = "LogEvent",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    onParent = false,
    name = "CancelTask",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.Licenses",
    onParent = false,
    name = "Licenses",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.Diagnostics",
    onParent = false,
    name = "Diagnostics",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.Settings",
    onParent = false,
    name = "Settings",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.VCServer",
    onParent = false,
    name = "VCServer",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,

```

```

    privId = "Global.CapacityPlanning",
    onParent = false,
    name = "CapacityPlanning",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.ScriptAction",
    onParent = false,
    name = "ScriptAction",
    privGroupName = "Global",
  },
  :
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.Proxy",
    onParent = false,
    name = "Proxy",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.DisableMethods",
    onParent = false,
    name = "DisableMethods",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.EnableMethods",
    onParent = false,
    name = "EnableMethods",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.ServiceManagers",
    onParent = false,
    name = "ServiceManagers",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.Health",
    onParent = false,
    name = "Health",
    privGroupName = "Global",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Global.SystemTag",
    onParent = false,
    name = "SystemTag",
    privGroupName = "Global",
  },
}

```

```
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Global.GlobalTag",
  name = "GlobalTag",
  privGroupName = "Global",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Folder.Create",
  onParent = false,
  name = "Create",
  privGroupName = "Folder",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Folder.Delete",
  onParent = true,
  name = "Delete",
  privGroupName = "Folder",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Folder.Rename",
  onParent = false,
  name = "Rename",
  privGroupName = "Folder",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Folder.Move",
  onParent = false,
  name = "Move",
  privGroupName = "Folder",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datacenter.Create",
  onParent = false,
  name = "Create",
  privGroupName = "Datacenter",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datacenter.Delete",
  onParent = true,
  name = "Delete",
  privGroupName = "Datacenter",
},
  dynamicType = <unset>,
  privId = "Datacenter.Rename",
  onParent = false,
  name = "Rename",
  privGroupName = "Datacenter",
},
```

```
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datacenter.Move",
  onParent = false,
  name = "Move",
  privGroupName = "Datacenter",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datacenter.IpPoolConfig",
  onParent = false,
  name = "IpPoolConfig",
  privGroupName = "Datacenter",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.Rename",
  onParent = false,
  name = "Rename",
  privGroupName = "Datastore",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.Move",
  onParent = false,
  name = "Move",
  privGroupName = "Datastore",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.Delete",
  onParent = true,
  name = "Delete",
  privGroupName = "Datastore",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.Browse",
  onParent = false,
  privGroupName = "Datastore",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.DeleteFile",
  onParent = false,
  name = "DeleteFile",
  privGroupName = "Datastore",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.FileManagement",
  onParent = false,
  name = "FileManagement",
  privGroupName = "Datastore",
}
```

```
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.AllocateSpace",
  onParent = false,
  name = "AllocateSpace",
  privGroupName = "Datastore",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.Config",
  onParent = false,
  name = "Config",
  privGroupName = "Datastore",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Datastore.UpdateVirtualMachineFiles",
  onParent = false,
  name = "UpdateVirtualMachineFiles",
  privGroupName = "Datastore",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Network.Move",
  onParent = false,
  name = "Move",
  privGroupName = "Network",
},
(vim.AuthorizationManager.Privilege) {
  privId = "Network.Delete",
  onParent = true,
  name = "Delete",
  privGroupName = "Network",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Network.Config",
  onParent = false,
  name = "Config",
  privGroupName = "Network",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Network.Assign",
  onParent = false,
  name = "Assign",
  privGroupName = "Network",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "DVSwitch.Create",
  onParent = false,
  name = "Create",
```

```

    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.Modify",
    onParent = false,
    name = "Modify",
    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.HostOp",
    onParent = false,
    name = "HostOp",
    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.PolicyOp",
    onParent = false,
    name = "PolicyOp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.PortConfig",
    onParent = false,
    name = "PortConfig",
    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.PortSetting",
    onParent = false,
    name = "PortSetting",
    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.Delete",
    onParent = true,
    name = "Delete",
    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.Move",
    onParent = false,
    name = "Move",
    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.Vspan",
    onParent = false,

```

```

    name = "Vspan",
    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVSwitch.ResourceManagement",
    onParent = false,
    name = "ResourceManagement",
    privGroupName = "DVSwitch",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    onParent = false,
    name = "Create",
    privGroupName = "DVPortgroup",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVPortgroup.Modify",
    onParent = false,
    name = "Modify",
    privGroupName = "DVPortgroup",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVPortgroup.PolicyOp",
    onParent = false,
    name = "PolicyOp",
    privGroupName = "DVPortgroup",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVPortgroup.ScopeOp",
    onParent = false,
    name = "ScopeOp",
    privGroupName = "DVPortgroup",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "DVPortgroup.Delete",
    onParent = true,
    name = "Delete",
    privGroupName = "DVPortgroup",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Inventory.AddStandaloneHost",
    onParent = false,
    name = "AddStandaloneHost",
    privGroupName = "Host.Inventory",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Inventory.CreateCluster",

```

```

onParent = false,
name = "CreateCluster",
privGroupName = "Host.Inventory",
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Inventory.AddHostToCluster",
onParent = false,
name = "AddHostToCluster",
privGroupName = "Host.Inventory",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Inventory.RemoveHostFromCluster",
onParent = true,
name = "RemoveHostFromCluster",
privGroupName = "Host.Inventory",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Inventory.MoveCluster",
onParent = false,
name = "MoveCluster",
privGroupName = "Host.Inventory",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Inventory.RenameCluster",
onParent = false,
name = "RenameCluster",
privGroupName = "Host.Inventory",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Inventory.DeleteCluster",
onParent = true,
name = "DeleteCluster",
privGroupName = "Host.Inventory",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Inventory.EditCluster",
onParent = false,
name = "EditCluster",
privGroupName = "Host.Inventory",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Inventory.MoveHost",
name = "MoveHost",
privGroupName = "Host.Inventory",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.AuthenticationStore",

```

```

onParent = false,
name = "AuthenticationStore",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.SystemManagement",
onParent = false,
name = "SystemManagement",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.Connection",
onParent = false,
name = "Connection",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.Maintenance",
onParent = false,
name = "Maintenance",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.AutoStart",
onParent = false,
name = "AutoStart",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.HyperThreading",
onParent = false,
name = "HyperThreading",
privGroupName = "Host.Config",
},
dynamicType = <unset>,
privId = "Host.Config.Storage",
onParent = false,
name = "Storage",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.NetService",
onParent = false,
name = "NetService",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,

```

```

    privId = "Host.Config.Memory",
    onParent = false,
    name = "Memory",
    privGroupName = "Host.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Config.Network",
    onParent = false,
    name = "Network",
    privGroupName = "Host.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Config.AdvancedConfig",
    onParent = false,
    name = "AdvancedConfig",
    privGroupName = "Host.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Config.Resources",
    onParent = false,
    name = "Resources",
    privGroupName = "Host.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Config.Snmp",
    onParent = false,
    privGroupName = "Host.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Config.DateTime",
    onParent = false,
    name = "DateTime",
    privGroupName = "Host.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Config.PciPassthru",
    onParent = false,
    name = "PciPassthru",
    privGroupName = "Host.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Host.Config.Settings",
    onParent = false,
    name = "Settings",
    privGroupName = "Host.Config",
  },
  (vim.AuthorizationManager.Privilege) {

```

```
dynamicType = <unset>,
privId = "Host.Config.Patch",
onParent = false,
name = "Patch",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.Firmware",
onParent = false,
name = "Firmware",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Config.Power",
onParent = false,
name = "Power",
privGroupName = "Host.Config",
},
(vim.AuthorizationManager.Privilege) {
privId = "Host.Local.InstallAgent",
onParent = false,
name = "InstallAgent",
privGroupName = "Host.Local",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Local.ManageUserGroups",
onParent = false,
name = "ManageUserGroups",
privGroupName = "Host.Local",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Local.CreateVM",
onParent = false,
name = "CreateVM",
privGroupName = "Host.Local",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Local.ReconfigVM",
onParent = false,
name = "ReconfigVM",
privGroupName = "Host.Local",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "Host.Local.DeleteVM",
onParent = false,
name = "DeleteVM",
privGroupName = "Host.Local",
},
},
```

```

(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Host.Cim.CimInteraction",
  onParent = false,
  name = "CimInteraction",
  privGroupName = "Host.Cim",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Inventory.Create",
  onParent = false,
  name = "Create",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Inventory.CreateFromExisting",
  onParent = false,
  name = "CreateFromExisting",
  privGroupName = "VirtualMachine.Inventory",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Inventory.Register",
  onParent = false,
  name = "Register",
  privGroupName = "VirtualMachine.Inventory",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Inventory.Delete",
  onParent = true,
  name = "Delete",
  privGroupName = "VirtualMachine.Inventory",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Inventory.Unregister",
  onParent = true,
  name = "Unregister",
  privGroupName = "VirtualMachine.Inventory",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Inventory.Move",
  onParent = false,
  name = "Move",
  privGroupName = "VirtualMachine.Inventory",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.PowerOn",
  onParent = false,
  name = "PowerOn",
  privGroupName = "VirtualMachine.Interact",
}

```

```

},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  onParent = false,
  name = "PowerOff",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.Suspend",
  onParent = false,
  name = "Suspend",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.Reset",
  onParent = false,
  name = "Reset",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.AnswerQuestion",
  onParent = false,
  name = "AnswerQuestion",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.ConsoleInteract",
  onParent = false,
  name = "ConsoleInteract",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.DeviceConnection",
  onParent = false,
  name = "DeviceConnection",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.SetCDMedia",
  onParent = false,
  name = "SetCDMedia",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.SetFloppyMedia",
  onParent = false,
  name = "SetFloppyMedia",
  privGroupName = "VirtualMachine.Interact",
}

```

```

},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.ToolsInstall",
  onParent = false,
  name = "ToolsInstall",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.GuestControl",
  onParent = false,
  name = "GuestControl",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.DefragmentAllDisks",
  onParent = false,
  name = "DefragmentAllDisks",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.CreateSecondary",
  onParent = false,
  name = "CreateSecondary",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.TurnOffFaultTolerance",
  onParent = false,
  name = "TurnOffFaultTolerance",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.MakePrimary",
  name = "MakePrimary",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.TerminateFaultTolerantVM",
  onParent = false,
  name = "TerminateFaultTolerantVM",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.DisableSecondary",
  onParent = false,
  name = "DisableSecondary",
}

```

```

    privGroupName = "VirtualMachine.Interact",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Interact.EnableSecondary",
    onParent = false,
    name = "EnableSecondary",
    privGroupName = "VirtualMachine.Interact",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Interact.Record",
    onParent = false,
    name = "Record",
    privGroupName = "VirtualMachine.Interact",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Interact.Replay",
    onParent = false,
    name = "Replay",
    privGroupName = "VirtualMachine.Interact",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Interact.Backup",
    onParent = false,
    name = "Backup",
    privGroupName = "VirtualMachine.Interact",
  },
  dynamicType = <unset>,
  privId = "VirtualMachine.Interact.CreateScreenshot",
  onParent = false,
  name = "CreateScreenshot",
  privGroupName = "VirtualMachine.Interact",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Config.Rename",
  onParent = false,
  name = "Rename",
  privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Config.AddExistingDisk",
  onParent = false,
  name = "AddExistingDisk",
  privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VirtualMachine.Config.AddNewDisk",
  onParent = false,

```

```

    name = "AddNewDisk",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.RemoveDisk",
    onParent = false,
    name = "RemoveDisk",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.RawDevice",
    onParent = false,
    name = "RawDevice",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.HostUSBDevice",
    onParent = false,
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.CPUCount",
    onParent = false,
    name = "CPUCount",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.Memory",
    onParent = false,
    name = "Memory",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.AddRemoveDevice",
    onParent = false,
    name = "AddRemoveDevice",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.EditDevice",
    onParent = false,
    name = "EditDevice",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.Settings",

```

```

onParent = false,
name = "Settings",
privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Config.Resource",
onParent = false,
name = "Resource",
privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
privId = "VirtualMachine.Config.UpgradeVirtualHardware",
onParent = false,
name = "UpgradeVirtualHardware",
privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Config.ResetGuestInfo",
onParent = false,
name = "ResetGuestInfo",
privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Config.AdvancedConfig",
onParent = false,
name = "AdvancedConfig",
privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Config.DiskLease",
onParent = false,
name = "DiskLease",
privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Config.SwapPlacement",
onParent = false,
name = "SwapPlacement",
privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Config.DiskExtend",
onParent = false,
name = "DiskExtend",
privGroupName = "VirtualMachine.Config",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,

```

```

    privId = "VirtualMachine.Config.ChangeTracking",
    onParent = false,
    name = "ChangeTracking",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.Unlock",
    onParent = false,
    name = "Unlock",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.QueryUnownedFiles",
    onParent = false,
    name = "QueryUnownedFiles",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.ReloadFromPath",
    onParent = false,
    name = "ReloadFromPath",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Config.QueryFTCompatibility",
    onParent = false,
    name = "QueryFTCompatibility",
    privGroupName = "VirtualMachine.Config",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.State.CreateSnapshot",
    onParent = false,
    name = "CreateSnapshot",
    privGroupName = "VirtualMachine.State",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.State.RevertToSnapshot",
    onParent = false,
    name = "RevertToSnapshot",
    privGroupName = "VirtualMachine.State",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    onParent = false,
    name = "RemoveSnapshot",
    privGroupName = "VirtualMachine.State",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,

```

```

    privId = "VirtualMachine.State.RenameSnapshot",
    onParent = false,
    name = "RenameSnapshot",
    privGroupName = "VirtualMachine.State",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Provisioning.Customize",
    onParent = false,
    name = "Customize",
    privGroupName = "VirtualMachine.Provisioning",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Provisioning.Clone",
    onParent = false,
    name = "Clone",
    privGroupName = "VirtualMachine.Provisioning",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Provisioning.PromoteDisks",
    onParent = false,
    name = "PromoteDisks",
    privGroupName = "VirtualMachine.Provisioning",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Provisioning.CreateTemplateFromVM",
    onParent = false,
    name = "CreateTemplateFromVM",
    privGroupName = "VirtualMachine.Provisioning",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Provisioning.DeployTemplate",
    onParent = false,
    name = "DeployTemplate",
    privGroupName = "VirtualMachine.Provisioning",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Provisioning.CloneTemplate",
    onParent = false,
    name = "CloneTemplate",
    privGroupName = "VirtualMachine.Provisioning",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VirtualMachine.Provisioning.MarkAsTemplate",
    onParent = false,
    name = "MarkAsTemplate",
    privGroupName = "VirtualMachine.Provisioning",
  },
  (vim.AuthorizationManager.Privilege) {

```

```

dynamicType = <unset>,
privId = "VirtualMachine.Provisioning.MarkAsVM",
onParent = false,
name = "MarkAsVM",
privGroupName = "VirtualMachine.Provisioning",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Provisioning.ReadCustSpecs",
onParent = false,
name = "ReadCustSpecs",
privGroupName = "VirtualMachine.Provisioning",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Provisioning.ModifyCustSpecs",
onParent = false,
name = "ModifyCustSpecs",
privGroupName = "VirtualMachine.Provisioning",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Provisioning.DiskRandomAccess",
onParent = false,
name = "DiskRandomAccess",
privGroupName = "VirtualMachine.Provisioning",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Provisioning.DiskRandomRead",
name = "DiskRandomRead",
privGroupName = "VirtualMachine.Provisioning",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Provisioning.GetVmFiles",
onParent = false,
name = "GetVmFiles",
privGroupName = "VirtualMachine.Provisioning",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VirtualMachine.Provisioning.PutVmFiles",
onParent = false,
name = "PutVmFiles",
privGroupName = "VirtualMachine.Provisioning",
},
(vim.AuthorizationManager.Privilege) {
dynamicType = <unset>,
privId = "VRMPolicy.Query",
onParent = false,
name = "Query",
privGroupName = "VRMPolicy",
},
}

```

```
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VRMPolicy.Update",
  onParent = false,
  name = "Update",
  privGroupName = "VRMPolicy",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.AssignVMToPool",
  onParent = false,
  name = "AssignVMToPool",
  privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.AssignVAppToPool",
  onParent = false,
  name = "AssignVAppToPool",
  privGroupName = "Resource",
},
dynamicType = <unset>,
privId = "Resource.ApplyRecommendation",
onParent = false,
name = "ApplyRecommendation",
privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.CreatePool",
  onParent = false,
  name = "CreatePool",
  privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.RenamePool",
  onParent = false,
  name = "RenamePool",
  privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.EditPool",
  onParent = true,
  name = "EditPool",
  privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.MovePool",
  onParent = false,
  name = "MovePool",
  privGroupName = "Resource",
}
```

```
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.DeletePool",
  onParent = true,
  name = "DeletePool",
  privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.HotMigrate",
  onParent = false,
  privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.ColdMigrate",
  onParent = false,
  name = "ColdMigrate",
  privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Resource.QueryVMotion",
  onParent = false,
  name = "QueryVMotion",
  privGroupName = "Resource",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Alarm.Create",
  onParent = false,
  name = "Create",
  privGroupName = "Alarm",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Alarm.Delete",
  onParent = false,
  name = "Delete",
  privGroupName = "Alarm",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Alarm.Edit",
  onParent = false,
  name = "Edit",
  privGroupName = "Alarm",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Alarm.Acknowledge",
  onParent = false,
  name = "Acknowledge",
}
```

```

    privGroupName = "Alarm",
  },
  (vim.AuthorizationManager.Privilege) {
    onParent = false,
    name = "SetStatus",
    privGroupName = "Alarm",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Alarm.DisableActions",
    onParent = false,
    name = "DisableActions",
    privGroupName = "Alarm",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Task.Create",
    onParent = false,
    name = "Create",
    privGroupName = "Task",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Task.Update",
    onParent = false,
    name = "Update",
    privGroupName = "Task",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "ScheduledTask.Create",
    onParent = false,
    name = "Create",
    privGroupName = "ScheduledTask",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "ScheduledTask.Delete",
    onParent = false,
    name = "Delete",
    privGroupName = "ScheduledTask",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "ScheduledTask.Run",
    onParent = false,
    name = "Run",
    privGroupName = "ScheduledTask",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "ScheduledTask.Edit",
    onParent = false,
    name = "Edit",
    privGroupName = "ScheduledTask",
  },

```

```

},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Sessions.TerminateSession",
  onParent = false,
  name = "TerminateSession",
  privGroupName = "Sessions",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Sessions.ValidateSession",
  onParent = false,
  name = "ValidateSession",
  privGroupName = "Sessions",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Sessions.GlobalMessage",
  onParent = false,
  name = "GlobalMessage",
  privGroupName = "Sessions",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Sessions.ImpersonateUser",
  onParent = false,
  name = "ImpersonateUser",
  privGroupName = "Sessions",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Performance.ModifyIntervals",
  onParent = false,
  name = "ModifyIntervals",
  privGroupName = "Performance",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Authorization.ModifyRoles",
  name = "ModifyRoles",
  privGroupName = "Authorization",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Authorization.ReassignRolePermissions",
  onParent = false,
  name = "ReassignRolePermissions",
  privGroupName = "Authorization",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "Authorization.ModifyPermissions",
  onParent = true,
  name = "ModifyPermissions",
}

```

```

    privGroupName = "Authorization",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Extension.Register",
    onParent = false,
    name = "Register",
    privGroupName = "Extension",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Extension.Update",
    onParent = false,
    name = "Update",
    privGroupName = "Extension",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Extension.Unregister",
    onParent = false,
    name = "Unregister",
    privGroupName = "Extension",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.ResourceConfig",
    onParent = true,
    name = "ResourceConfig",
    privGroupName = "VApp",
  },
  dynamicType = <unset>,
  privId = "VApp.InstanceConfig",
  onParent = false,
  name = "InstanceConfig",
  privGroupName = "VApp",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VApp.ApplicationConfig",
  onParent = false,
  name = "ApplicationConfig",
  privGroupName = "VApp",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VApp.Export",
  onParent = false,
  name = "Export",
  privGroupName = "VApp",
},
(vim.AuthorizationManager.Privilege) {
  dynamicType = <unset>,
  privId = "VApp.Import",
  onParent = false,

```

```

    name = "Import",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.ExtractOvfEnvironment",
    onParent = false,
    name = "ExtractOvfEnvironment",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.AssignVM",
    onParent = false,
    name = "AssignVM",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.AssignResourcePool",
    onParent = false,
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.AssignVApp",
    onParent = false,
    name = "AssignVApp",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.Clone",
    onParent = false,
    name = "Clone",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.Create",
    onParent = false,
    name = "Create",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.Delete",
    onParent = true,
    name = "Delete",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.Unregister",

```

```
    onParent = true,
    name = "Unregister",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.Move",
    onParent = false,
    name = "Move",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    privId = "VApp.PowerOn",
    onParent = false,
    name = "PowerOn",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.PowerOff",
    onParent = false,
    name = "PowerOff",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.Suspend",
    onParent = false,
    name = "Suspend",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "VApp.Rename",
    onParent = false,
    name = "Rename",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Profile.Create",
    onParent = false,
    name = "Create",
    privGroupName = "Profile",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Profile.Delete",
    onParent = false,
    name = "Delete",
    privGroupName = "Profile",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>
```

```
    privId = "Profile.Edit",
    onParent = false,
    name = "Edit",
    privGroupName = "VApp",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Profile.Create",
    onParent = false,
    name = "Create",
    privGroupName = "Profile",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Profile.Delete",
    onParent = false,
    name = "Delete",
    privGroupName = "Profile",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Profile.Edit",
    onParent = false,
    name = "Edit",
    privGroupName = "Profile",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Profile.View",
    onParent = false,
    name = "View",
    privGroupName = "Profile",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Profile.Clear",
    onParent = false,
    name = "Clear",
    privGroupName = "Profile",
  },
  (vim.AuthorizationManager.Privilege) {
    dynamicType = <unset>,
    privId = "Profile.Export",
    onParent = false,
    name = "Export",
    privGroupName = "Profile",
  }
}
```

vimsvc/auth/lockdown_is_possible

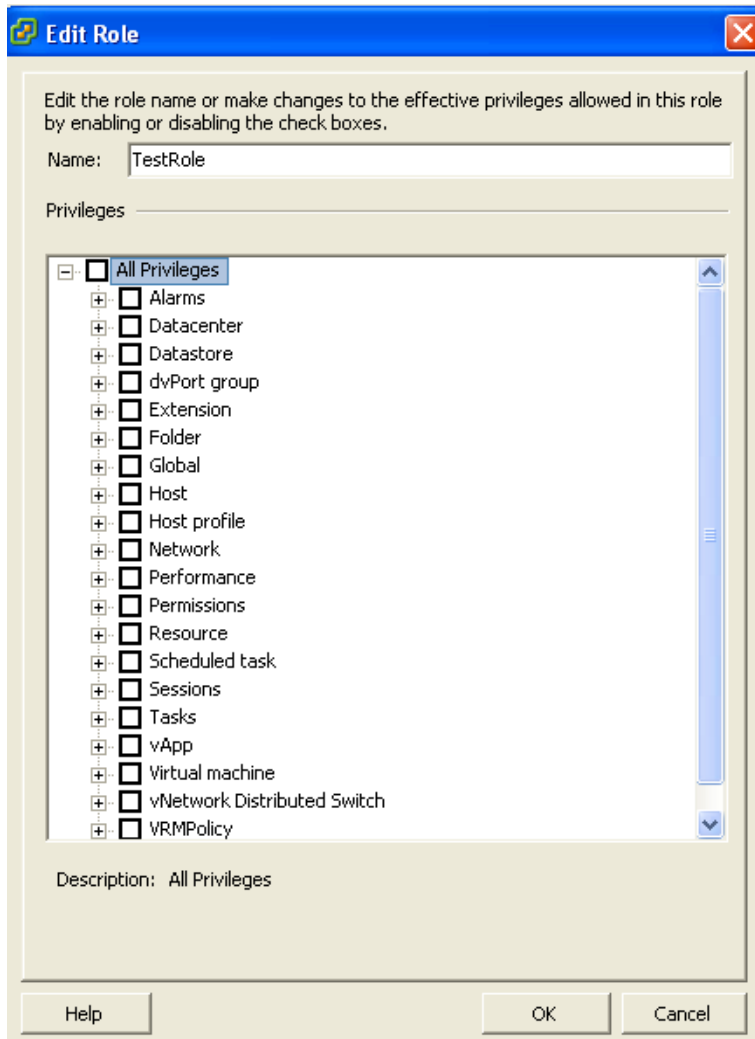
~ # vim-cmd vimsvc/auth/lockdown_is_possible

false

vimsvc/auth/role_add

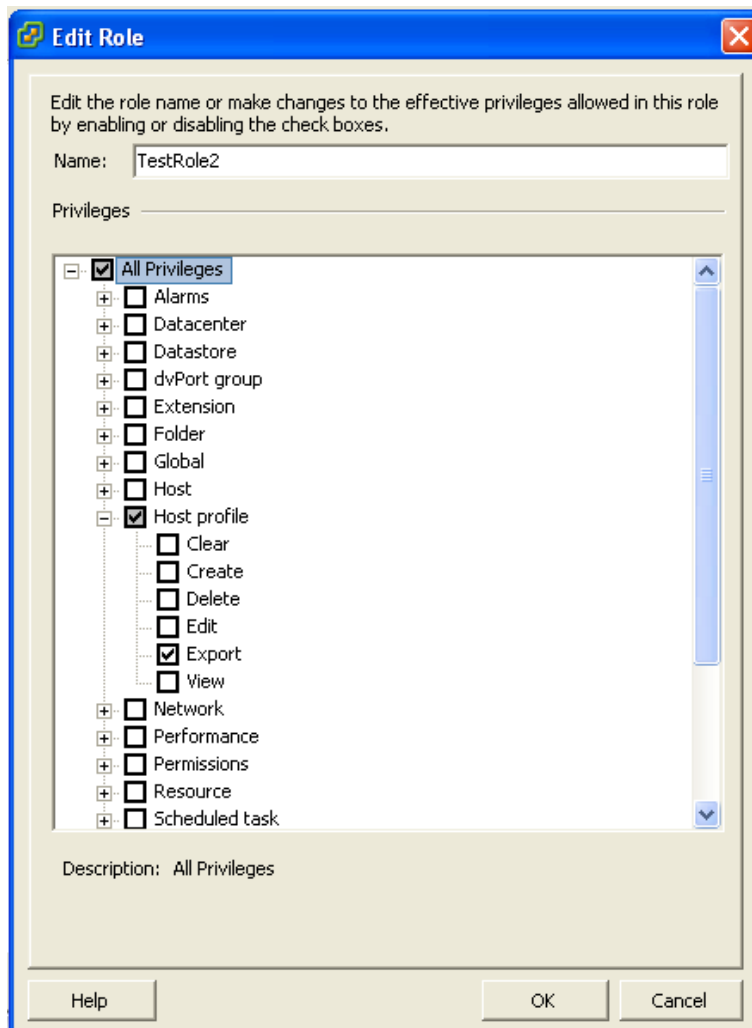
~ # vim-cmd vimsvc/auth/role_add TestRole

Role created: 12



```
~ # vim-cmd vimsvc/auth/role_add TestRole2 Profile.Export
```

Role created: 13



* **[vimsvc/login](#)**

* **[vimsvc/task_cancel](#)**

[vimsvc/task_list](#)

```
~ # vim-cmd vimsvc/task_list
```

```
(ManagedObjectReference) [
```

```
'vim.Task:haTask-ha-folder-vm-vim.Folder.registerVm-684'
```

```
]
```

* **[vimsvc/connect](#)**

* **[vimsvc/logout](#)**

vimsvc/task_description

~ # vim-cmd vimsvc/task_description

```
(vim.TaskDescription) {
  dynamicType = <unset>,
  methodInfo = (vim.ElementDescription) [
    (vim.ElementDescription) {
      dynamicType = <unset>,
      label = "Query options view",
      summary = "Returns nodes in the option hierarchy",
      key = "option.OptionManager.queryView",
    },
    (vim.ElementDescription) {
      dynamicType = <unset>,
      label = "Update option values",
      summary = "Updates one or more properties",
      key = "option.OptionManager.updateValues",
    },
    (vim.ElementDescription) {
      dynamicType = <unset>,
      label = "retrieveArgumentDescription",
      summary = "retrieveArgumentDescription",
      key = "event.EventManager.retrieveArgumentDescription",
    },
    (vim.ElementDescription) {
      dynamicType = <unset>,
      label = "Create event collector",
      summary = "Creates an event collector to retrieve all server events based on a filter",
      key = "event.EventManager.createCollector",
    },
    (vim.ElementDescription) {
      dynamicType = <unset>,
      label = "Log user event",
      summary = "Logs a user-defined event",
      key = "event.EventManager.logUserEvent",
    },
    (vim.ElementDescription) {
      dynamicType = <unset>,
      label = "Get events",
      summary = "Provides the events selected by the specified filter",
      key = "event.EventManager.QueryEvent",
    },
  ],
  ...
}
```

vimsvc/license

--show

Display license info.

~ # vim-cmd vimsvc/license --show

```
[200] Sending request for installed licenses...[200] Complete, result is:
serial: 00000-00000-00000-00000-00000
vmodl key: eval
name: Evaluation Mode
total: 1
used: 1
unit: host
Properties:
[ProductName] = VMware ESX Server
[ProductVersion] = 4.0
[evaluation] = License has not been set, evaluation Period in effect.
[expirationHours] = 1369
[expirationMinutes] = 48
[expirationDate] = 2010-10-10T02:38:55.816988Z
[system_time] = 2010-08-14T00:50:55.844015Z
[feature] = vsmp:8 ("Up to 8-way virtual SMP")
[feature] = esxHost ("vCenter agent for ESX Server")
[feature] = vstorage ("vStorage APIs")
[feature] = vmsafe ("VMsafe")
[feature] = dvfilter ("dvFilter")
[feature] = das ("VMware HA")
[feature] = hotplug ("Hot-Pluggable virtual HW")
[feature] = vmotion ("vMotion")
[feature] = ft ("VMware FT")
[feature] = dr ("Data Recovery")
[feature] = vshield ("vShield Zones")
[feature] = drs ("VMware DRS")
[feature] = svmotion ("Storage vMotion")
[feature] = mpio ("MPIO / Third-Party Multi-Pathing")
[feature] = dvs ("vNetwork Distributed Switch")
[feature] = hostprofile ("Host profiles")
[feature] = serialuri:2 ("Remote virtual Serial Port Concentrator")
[feature] = sioshares ("Storage I/O Control")
[feature] = dpvmotion ("dpvmotion")
[feature] = vaai ("vaai")
[FileVersion] = 4.1.0.8
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e1-4core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e1-ucore-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e1-vm-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e1-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e2-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e2-4core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e3-4core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e2-vm-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e3-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e4-vm-200803
```

```
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e3-vm-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e4-8core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e4-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e5-vm-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e5-4core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e5-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e6-4core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e7-8core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e6-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e6-vm-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e7-ucore-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e8-8core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e8-ucore-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e7-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e7-vm-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e8-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e8-vm-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e9-8core-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e9-ve-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-e9-vm-200803
[LicenseFilePath] = /usr/lib/vmware/licenses/site/license-esx-40-eval-200803
```

[200] End of report.

```
--set=string
```

Set license serial number and optionally append /sourceId.

vimsvc/property_dump

```
~ # vim-cmd vimsvc/property_dump
```

Found 60 managed objects:

1. vim.AuthorizationManager::ha-authmgr
2. vim.ComputeResource::ha-compute-res
3. vim.Datacenter::ha-datacenter
4. vim.Datastore::192.168.9.222:/media/dwl/public/ISO_Library
5. vim.Datastore::192.168.9.222:/media/dwl/public/NFS
6. vim.Datastore::4c5cc8e0-80fe1968-98ec-000c29829b7f
7. vim.Datastore::4c654f65-75b87896-ef8e-000c29829b75
8. vim.Datastore::4c65661f-1a2e18b5-783e-000c29829b75
9. vim.Datastore::4c656728-065ce6bc-aea3-000c29829b75
10. vim.DiagnosticManager::ha-diagnosticmgr
11. vim.EnvironmentBrowser::176-envmgr
12. vim.EnvironmentBrowser::192-envmgr
13. vim.EnvironmentBrowser::208-envmgr
14. vim.EnvironmentBrowser::224-envmgr
15. vim.EnvironmentBrowser::240-envmgr
16. vim.EnvironmentBrowser::256-envmgr
17. vim.EnvironmentBrowser::ha-env-browser-vmx-07
18. vim.Folder::ha-folder-host
19. vim.Folder::ha-folder-root
20. vim.Folder::ha-folder-vm
21. vim.HostSystem::ha-host
22. vim.LicenseManager::ha-license-manager

23. vim.Network::HaNetwork-TSpG
24. vim.Network::HaNetwork-VM-0
25. vim.Network::HaNetwork-VM-1
26. vim.Network::HaNetwork-VM-2
27. vim.Network::HaNetwork-pgVMKTest
28. vim.PerformanceManager::ha-perfmgr
29. vim.ResourcePool::ha-root-pool
30. vim.ResourcePool::pool1
31. vim.SearchIndex::ha-searchindex
32. vim.ServiceInstance::ServiceInstance
33. vim.SessionManager::ha-sessionmgr
34. vim.Task::haTask--vim.LicenseManager.updateLicense-757
35. vim.Task::haTask--vim.host.FirmwareSystem.syncConfiguration-691
36. vim.Task::haTask-ha-folder-vm-vim.Folder.registerVm-684
37. vim.TaskManager::ha-taskmgr
38. vim.UserDirectory::ha-user-directory
39. vim.VirtualMachine::176
40. vim.VirtualMachine::192
41. vim.VirtualMachine::208
42. vim.VirtualMachine::224
43. vim.VirtualMachine::240
44. vim.VirtualMachine::256
...

*** [vimsvc/task_info](#)**