# Table of Contents

Introduction ........................................................................................................ 3  
Architecture ........................................................................................................ 3  
Management ......................................................................................................... 4  
  Automation ......................................................................................................... 5  
  Installation Destination ....................................................................................... 7  
  ESXi Scratch Partition ....................................................................................... 8  
  Deployment ........................................................................................................ 9  
  Hardware Monitoring ............................................................................................ 12  
  Firmware Upgrades ............................................................................................... 13  
  Systems Management and Backup ...................................................................... 14  
  Patching and Updating .......................................................................................... 14  
  Image Builder ...................................................................................................... 16  
  User Authentication ............................................................................................... 16  
  Logging ................................................................................................................ 17  
  ESXi Shell ............................................................................................................ 19  
  Local Access and Lockdown Mode ..................................................................... 20  
  ESXi Firewall ....................................................................................................... 22  
  ESXi Firewall CLI .................................................................................................. 23  
  Diagnostics and Troubleshooting ........................................................................ 24  
Summary .............................................................................................................. 27  
VMware ESXi Editions ............................................................................................ 28  
References ............................................................................................................ 28
Introduction

The hypervisor architecture of VMware vSphere® 5.0 plays a critical role in the management of the virtual infrastructure. The introduction of the bare-metal VMware ESX® architecture in 2001 significantly enhanced performance and reliability, which in turn enabled customers to extend the benefits of virtualization to their mission-critical applications. The introduction of the VMware ESXi™ architecture represents a similar leap forward in reliability and virtualization management. Less than 5 percent of the footprint of ESX, ESXi runs independently of a host operating system (OS) and improves hypervisor management in the areas of security, deployment and configuration, and ongoing administration. Yet none of this comes at the cost of functionality.

This paper describes the architecture of ESXi and then explains how various management tasks are performed in it. This information can be used to help plan a migration to the ESXi architecture from the legacy ESX framework and to improve or enhance day-to-day operations.

Architecture

In the original ESX architecture, the virtualization kernel (VMkernel) is augmented by a management partition known as the console operating system (COS) or service console. The primary purpose of the COS is to provide a management interface with the host. Various VMware® management agents are deployed in the COS, along with other infrastructure service agents (for example, name service, time service, logging, and so on). In this architecture, many customers deploy other agents from third parties to provide a particular functionality, such as hardware monitoring and systems management. Furthermore, individual administrative users log in to the COS to run configuration and diagnostic commands and scripts.

In the ESXi architecture, the COS has been removed, and all of the VMware agents run directly on the VMkernel. Infrastructure services are provided natively through modules included in the VMkernel. Other authorized third-party modules, such as hardware drivers and hardware monitoring components, can run in the VMkernel as well. Only modules that have been digitally signed by VMware are allowed on the system, creating a tightly locked-down architecture. Preventing arbitrary code from running on the ESXi host greatly improves the security and stability of the system.
Figure 2. Architecture of VMware ESXi

Management

The management functionality that was provided by agents in the ESX architecture is now exposed via APIs in the ESXi architecture. This enables an “agentless” approach to hardware monitoring and systems management. VMware also created remote command-line interface (CLI) tools, such as the VMware vSphere® Command-Line Interface (vSphere vCLI) and VMware vSphere® PowerCLI (vSphere PowerCLI), to provide command and scripting capabilities in a more controlled manner. These remote command-line sets include a variety of commands for configuration, diagnostics and troubleshooting. For low-level diagnostics and for initial configuration, a menu-driven and command-line interface is available on the local console of the server. In addition, a local version of the “esxcli” command set is accessible directly from the host’s local shell, referred to as the “ESXi Shell.” These are discussed in more detail in the following sections, which also discuss individual management topics and describe how tasks are performed in the ESXi architecture.
Automation

To automate the management of an ESXi deployment, VMware has created easy-to-use CLI tools. Users can employ them to write scripts that provide the same functionality as the VMware vSphere® Client™ to automate manual tasks, enabling efficient management and configuration of small- to large-scale environments. These CLI tools work well with both ESXi and ESX hosts, empowering users to administer mixed environments easily.

vSphere PowerCLI is a robust, Windows-based CLI tool for automating all aspects of vSphere management, including host, network, storage, virtual machine, guest OS and more. It is distributed as a Windows PowerShell snap-in. Windows PowerShell is a scripting tool written by Microsoft and designed for the systems administrator. vSphere PowerCLI includes more than 300 PowerShell cmdlets, along with built-in documentation and samples. It seamlessly blends the vSphere platform with Windows and .NET, which means users can utilize it by itself or within many different third-party tools.

vSphere vCLI is a separate set of CLI tools that, like vSphere PowerCLI, can be used to perform remote management of ESX and ESXi hosts. Whereas vSphere PowerCLI is very Windows-centric, vSphere vCLI has a more Linux-like “look and feel” and targets non-Windows users. VMware provides vCLI packages for installation on both Windows and Linux systems. vCLI is also packaged as part of the VMware vSphere® 5.0 Management Assistant (vMA), a Linux-based virtual appliance that packages the vCLI together with other tools to facilitate deployment and use of the vCLI.

Whereas vSphere PowerCLI is built on top of Windows PowerShell and is composed of a collection of PowerShell cmdlets, vSphere vCLI is a combination of the following separate command-line tools:

- “esxcli”
- “vmkfstools”
- “vmware-cmd”
- “resxtop”
- “vicfg-*”

Figure 3. New and Improved Paradigm for VMware ESX Management
In ESXi 5.0, most of the vCLI command-line tools are the same as in earlier ESX/ESXi releases, with the exception of the “esxcli” command. In ESXi 5.0, the “esxcli” command has been enhanced and is now available both locally from the ESXi Shell and remotely through the vCLI. The new “esxcli” command marks the beginning of VMware efforts to standardize on a single command-line interface for both local and remote administration.

![Figure 4. esxcli—a Standardized Command-Line Interface](image)

The improved “esxcli” command provides an intuitive interface that enables real-time discovery of command syntax. Whereas the command structure is similar in look-and-feel to its vSphere 4.x predecessor, the new “esxcli” command has an improved syntax that has been extended to include additional functionality not available in earlier versions, such as the ability to configure network policies and security policies, manage VIBs, and configure and manage the ESXi firewall.

The “vicfg-” family of commands, introduced in the vCLI in vSphere 4.0, is still available in vCLI 5.0, but most of these commands are played down in favor of the “esxcli” equivalent. The following “vicfg” commands do not have an “esxcli” equivalent in vSphere 5.0:

- “vicfg-authconfig”
- “vicfg-cfgbackup”
- “vicfg-hostops”
- “vicfg-ipsec”
- “vicfg-ntp”
- “vicfg-route”
- “vicfg-snmp”
- “vicfg-user”
- “vifs”

All other “vicfg” commands should be avoided in favor of their “esxcli” equivalents, which are listed in the vSphere Command-Line Interface Concepts and Examples document, part of the vSphere 5.0 documentation set.

As was previously mentioned, the “esxcli” command is available on each ESXi host via the ESXi Shell, in addition to being available as part of the optional vCLI package that can be installed on any supported Windows or Linux server (or through the vMA).

![Figure 5. esxcli Command Structure](image)
In addition to providing a consistent “look and feel” for both local and remote CLI administration, the new “esxcli” command provides the ability to format the command output. Using the “--formatter” option, administrators can choose to have the command output formatted as XML, a key-value pair or a list of comma-separated values. The “esxcli” formatter enhances users’ ability to parse command output, helping to simplify scripting and improve report generation.

Both vSphere PowerCLI and vSphere vCLI (including “esxcli”) are built on the same APIs as the vSphere Client. They can be connected directly to an ESXi host or to VMware vCenter™. When connected to a host, they can execute commands directly on an ESXi host, similarly to how a command in the COS of ESX operates on only that host. Local authentication is required in this case. Alternatively, when communicating through vCenter, the vSphere vCLI and vSphere PowerCLI commands benefit from the same authentication (for example, Microsoft Active Directory) roles and privileges and event logging as vSphere Client interactions. This provides for a much more secure and auditable management framework.

The following table contains different categories of operational procedures and the preferred tool for each category. We have rated each tool per task to classify the level of expertise required:

<table>
<thead>
<tr>
<th>TASK</th>
<th>vCLI/vMA</th>
<th>POWERCLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting</td>
<td>Normal</td>
<td>Easy</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Hard</td>
<td>Normal</td>
</tr>
<tr>
<td>Configuration</td>
<td>Easy</td>
<td>Easy</td>
</tr>
<tr>
<td>Automation</td>
<td>Normal</td>
<td>Easy</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>Easy</td>
<td>Hard</td>
</tr>
</tbody>
</table>

Table 1. Ease of Operational Tasks

Although each of the tools can be used to accomplish a given task, the preceding table can be used as an indication of which tools would best meet a user’s requirements.

**Installation Destination**

When planning the implementation of—or migration to—ESXi, one of the first decisions that must be made concerns the type of installation destination to be used. The form factor of ESXi enables it to be installed on multiple different installation destination types, including the following:

- Local disk (including SSD)
- Removable media
  - USB
  - SD
- Boot from SAN
  - FC
  - iSCSI
- Diskless PXE boot

Local disks are a popular installation destination. Local disk installations have an advantage over removable devices because users are able to provide a level of resiliency to protect against disk failure. Resiliency refers to the ability to run two local disks in RAID-1. Although ESXi runs in the host memory, it routinely writes its configuration to the boot disk (every 10 minutes by default) to ensure changes will persist following a host reboot. In the case of boot media failure, these updates might be at risk, possibly resulting in a loss of configuration changes. Having a resilient boot device helps eliminate the risk.
Removable devices such as USB and SD are also popular ESXi installation destinations due to the flexibility and cost factors associated with them. These devices typically have a shorter life span than hard disks and therefore impose a minor risk. However, hardware vendors have found a solution that increases resiliency by offering a “dual-SD module” configuration. And many customers have further mitigated the risk by using enterprise-grade USB/SD modules and keeping several of them on hand. VMware supports removable devices only under one or more of the following conditions:

• The server on which a user wants to install VMware ESXi 5.0 is in the VMware ESXi 5.0 Hardware Compatibility List (HCL).

• A user has purchased a server with VMware ESXi 5.0 embedded on the server from a certified vendor.

• A user has utilized a USB or SD Flash device that is approved by the server vendor for the particular server model on which they want to install VMware ESXi 5.0 on a USB or SD Flash storage device.

As of vSphere 4.1, support for boot from SAN—both FC and iSCSI—has also been included. Boot from SAN gives users resiliency and enables them to leverage the flexibility of a diskless server while still providing them with the option to do a scripted installation. Booting from SAN requires using a supported storage device and adapter. Consult the vSphere Storage/SAN Compatibility Guide for information on supported storage components.

Diskless PXE boot is introduced with vSphere 5.0 as part of VMware vSphere® Auto Deploy. Using Auto Deploy, experienced systems administrators can manage large deployments efficiently. Unlike the other installation options, Auto Deploy does not store the ESXi image, its configuration or its state on disk. Instead, state is managed through VMware vCenter Server™. The ESXi image profile is directly loaded into memory over the network, and the host is configured using host profiles. Auto Deploy enables a great deal of flexibility in changing the identity of a physical server. It also enables a very agile update management. With Auto Deploy, updating a hypervisor is as simple as updating the host’s image profile and rebooting the host. Each time the host reboots, a fresh ESXi image will be reloaded, which also helps eliminate configuration discrepancy between hosts. Auto Deploy does require an initial investment in terms of knowledge, architecture and implementation tasks.

Each type of installation media has its benefits. Depending on the environment, all media should be considered. Based on requirements and constraints regarding budget, licensing and array capabilities, a decision must be made on a per-case basis. Generally speaking, using “local disks” is the most compelling option because it provides improved resiliency in comparison to USB/SD, and it is relatively inexpensive in comparison to boot from SAN. For large environments (20+ hosts) we recommend testing Auto Deploy. Auto Deploy offers a highly flexible and agile solution and can reduce the amount of operational effort associated with managing and maintaining ESXi hosts.

ESXi Scratch Partition

An important consideration when choosing the type of boot device for ESXi hosts is the location of the ESXi scratch partition. The scratch partition is a 4GB region used by ESXi to store log files and core dumps, as well as a staging area for updates and other temporary files. During the ESXi installation, the installer will attempt to allocate a 4GB region of disk from a local disk for the scratch partition. If no local disk is available, the scratch partition will be created on a RAM disk in the host’s memory. It’s important to note that because the scratch partition is an area of heavy write I/O, placing it on a USB/SD device is not allowed. When installing on a host with no local datastores (i.e., boot from USB/SD), it’s important that following the ESXi installation, users manually reconfigure the scratch partition to reside on a persistent datastore.
Deployment

Various deployment methods are supported for ESXi, such as booting the installer off of a DVD or over PXE, and deploying the ESXi image onto a local disk over the network using a variety of protocols, including secure HTTP. VMware ESXi 5.0 enables users to do a scripted installation of the ESXi software onto the local disk of a server, analogous to the kick-start mechanism used for ESX architecture. The scripted installation configuration file (typically named “ks.cfg”) can also specify the following scripts to be executed during the installation:

- Preinstall
- Postinstall
- First-boot

These scripts are run locally on the ESXi host and can perform various tasks, such as configuring the host’s virtual networking and joining it to vCenter Server. These scripts can be written using either the ESXi Shell or Python.

In ESXi 5.0, boot from SAN is supported on Fibre Channel SAN, as well as iSCSI and FCoE for certain storage adapters that have been qualified for this capability.

VMware ESXi 5.0 is still available preinstalled on Flash drives on certain server models available from a number of hardware OEM vendors. (Consult the server HCL to determine which combinations of server and USB or Flash drive are supported.)

As stated, with vSphere 5.0, VMware has added scripted installation capabilities to ESXi. A basic scripted CD-ROM–based install entails the following procedure:

1. Boot from the ESXi CD-ROM.
2. Press “Tab” when the “VMware VMvisor Boot Menu” is displayed.
3. Edit the string so it includes the location of the script:
   
   ```
   > mboot.c32 -c boot.cfg ks=http://<ip-address>/ks.cfg
   ```
The string has changed compared to vSphere 4.1. The <ip-address> should be replaced with the ip-address of
the Web server hosting the configuration file. The ks.cfg configuration file can also be located on other types of
media such as CD-ROM or an FTP server. For more details, refer to the VMware vSphere 5.0 ESXi Installable and
vCenter Server Setup Guide.

It is also possible to PXE boot the VMware ESXi installer. This however requires a TFTP server that supports PXE
boot, gPXE and a modification to the DHCP server to allow the DHCP server to send the host the correct TFTP
and PXE information (DHCP options 66 and 67). For more details, refer to the VMware vSphere 5.0 ESXi
Installable and vCenter Server Setup Guide, where this procedure is fully documented.

When using a PXE mechanism to facilitate the installation or a CD-ROM, an answer script is required. The script
follows a standardized format to supply the installer with the correct parameters. The following example
includes an action on the first boot, to demonstrate the considerable capabilities the ESXi installer offers. Before
using a script in a production environment, it is recommended to extensively test and validate it in an isolated
environment. With vSphere 5.0, many of the scripted installation and upgrade commands either have been
replaced or deleted or are not supported anymore. For more details, refer to the VMware vSphere 5.0 ESXi
Installable and vCenter Server Setup Guide, where these commands are fully documented.

# Sample scripted installation file

# Accept the VMware End User License Agreement
vmaccepteula

# Set the root password for the DCUI and Tech Support Mode
rootpw mypassword

# Install on the first local disk available on machine
install --firstdisk --overwritevmfs

# Set the network to DHCP on the first network adapter, use the specified hostname and do
# not create a portgroup for the VMs
network --bootproto=dhcp --device=vmnic0 --addvmportgroup=0

# reboots the host after the scripted installation is completed
reboot

%firstboot --interpreter=busybox

# Add an extra nic to vSwitch0 (vmnic2)
esxcli network vswitch standard uplink add --uplink-name=vmnic2 --vswitch-name=vSwitch0

# Assign an IP-Address to the first VMkernel, this will be used for management
esxcli network ip interface ipv4 set --interface-name=vmk0 --ipv4=192.168.1.41
--netmask=255.255.255.0 --type=static

# Add vMotion Portgroup to vSwitch0, assign it VLAN ID 5 and create a VMkernel interface
esxcli network vswitch standard portgroup add --portgroup-name=vMotion --vswitch-name=vSwitch0
esxcli network vswitch standard portgroup set --portgroup-name=vMotion --vlan-id=5
esxcli network ip interface add --interface-name=vmk1 --portgroup-name=vMotion
esxcli network ip interface ipv4 set --interface-name=vmk1 --ipv4=192.168.2.41
--netmask=255.255.255.0 --type=static

# Enable vMotion on the newly created VMkernel vmk1
vim-cmd hostsvc/vmotion/vnic_set vmk1
# Add new vSwitch for VM traffic, assign uplinks, create a portgroup and assign a VLAN ID
esxcli network vswitch standard add --vswitch-name=vSwitch1
esxcli network vswitch standard uplink add --uplink-name=vmnic1 --vswitch-name=vSwitch1
esxcli network vswitch standard uplink add --uplink-name=vmnic3 --vswitch-name=vSwitch1
esxcli network vswitch standard portgroup add --portgroup-name=Production --vswitch-name=vSwitch1
esxcli network vswitch standard portgroup set --portgroup-name=Production --vlan-id=10

# Set DNS and hostname
esxcli system hostname set --fqdn=esxi5.localdomain
esxcli network ip dns search add --domain=localdomain
esxcli network ip dns server add --server=192.168.1.11
esxcli network ip dns server add --server=192.168.1.12

# Set the default PSP for EMC V-MAX to Round Robin as that is our preferred load balancing mechanism
esxcli storage nmp satp set --default-psp=VMW_PSP_RR --satp=VMW_SATP_SYMM

# Enable SSH and the ESXi Shell
vim-cmd hostsvc/enable_ssh
vim-cmd hostsvc/start_ssh
vim-cmd hostsvc/enable_esx_shell
vim-cmd hostsvc/start_esx_shell

This example script shows how to automate the installation of an ESXi host, including how to configure additional vSwitches, port groups including VLAN IDs and how to change the default path selection plugin (PSP).

The major change with ESXi 5.0 compared to ESXi 4.1 and prior is “esxcli.” The “esxcli” command has been enhanced with many new parameters (name spaces) and enables almost every configuration option available today. There are, however, still a few exceptions. In these cases, familiar commands such as “vicfg-*,” “vmware-cmd,” “vim-cmd” and “vmkfstools” can be used.

It is important to recognize the difference between the %pre, %post and %firstboot. In our example, we have used only %firstboot because that is most common when configuring ESXi hosts. It is executed during the first boot after the installer has completed. The following diagram depicts the process of a scripted installation where both the %post and %firstboot sections are used:

---

Figure 7. Scripted Installation Process
Both %pre and %post are most commonly used when there is a requirement to download driver packages or to make changes before the actual configuration. For instance, during the %post, a driver package might be downloaded to a local datastore, using "wget." This package might be installed during the %firstboot phase. In our example, the drivers must be available on local disk during the %firstboot phase to be able to install them. The following is an example of how to implement this:

%post
# download drivers to local volume
wget http://192.168.1.100/network.zip -O /vmfs/volumes/datastore1/network.zip

%firstboot
# install drivers that were downloaded
/sbin/esxcli software vib install --depot=/vmfs/volumes/datastore1/network.zip --vibname=<name of .VIB to install>

Depending on the scenario, it is also possible to download and install drivers during the %firstboot phase.

%firstboot
/sbin/esxcli software vib install --depot=https://192.168.1.100/network.zip --vibname=<name of .VIB to install>

As demonstrated, there are many ways to configure an ESXi host or to install additional drivers and packages when required. We have yet to face a problem that could not be resolved by the various command-line tools and APIs. We refer to the VMware VMTN Community for sample scripts.

Hardware Monitoring

The Common Information Model (CIM) is an open standard that defines a framework for agentless, standards-based monitoring of hardware resources for ESXi. This framework consists of a CIM object manager, often called a CIM broker, and a set of CIM providers.

CIM providers are the mechanisms that provide management access to device drivers and underlying hardware. Hardware vendors, including server manufacturers and specific hardware device vendors, can write providers to supply monitoring and management of their particular devices. VMware also writes providers that implement monitoring of server hardware, ESXi storage infrastructure and virtualization-specific resources. These providers run inside the ESXi system and are designed to be extremely lightweight and focused on specific management tasks. The CIM broker takes information from all CIM providers and presents it to the outside world via standard APIs, the most common one being WS-MAN. Any software tool compatible with one of these APIs, such as HP SIM or Dell OpenManage, can read this information, monitoring the hardware of the ESXi host.

One consumer of the CIM information is vCenter. Through a dedicated tab in the vSphere Client, users can view the hardware status of any ESXi host in their environment, providing a single view of the physical and virtual health of their systems. Users can also set vCenter alarms to be triggered on certain hardware events, such as temperature or power failure and warning states.
ESXi also exposes hardware status information via SNMP for other management tools that rely upon that standard. SNMP traps are available from both the ESXi host and vCenter. VMware ESXi 5.0 supports SNMPv2, and it can be configured using the vSphere vCLI command “vicfg-snmp” or using the “Set-VMHostSNMP” cmdlet in vSphere PowerCLI.

VMware ESXi 5.0 adds the capability to convert CIM indications to SNMP traps. Users should check with their hardware vendor to see whether its CIM provider supports this functionality. In addition, ESXi 5.0 now supports the Host Resources MIB (RFC 2790) and enables finer control over the types of traps sent by the SNMP agent.

**Firmware Upgrades**

Upgrading firmware on any platform is a cumbersome task. Historically, customers who have used the COS have upgraded the firmware with tools provided by the respective vendor. With ESXi, that approach will no longer work, due to the absence of the COS. Firmware upgrades, however, still must be applied periodically. The following options exist to solve this problem:

1. Hardware vendor vCenter plug-in or management application
2. Hardware vendor bootable upgrade CD-ROM/DVD
3. PXE boot of vendor’s upgrade CD-ROM/DVD
4. PXE boot of small Linux distribution

Several hardware vendors provide management plug-ins for vCenter Server that enable users to manage firmware upgrades from within the vSphere Client. These plug-ins frequently offer reporting capabilities that reduce the chances of inconsistency across the virtual infrastructure. Large environments typically use a centralized management application to manage hardware end to end, which also includes the capabilities to upgrade firmware.
Many vendors offer a bootable CD-ROM/DVD that contains all required drivers and firmware code. These are typically categorized per server model and can be used to boot a host from and manually upgrade the appropriate devices. This solution typically is used in environments of up to 10 hosts. For larger environments, we have found that using a PXE boot configuration in conjunction with the vendor-provided upgraded CD-ROM/DVD can be a flexible alternative. The overall transfer size of the total package might be a constraint.

Finding a unified solution to manage firmware and patches in an environment where multiple types of hardware from different vendors are used can be a challenge. Creating a custom, slimmed-down Linux appliance that identifies the hardware configuration and updates the firmware accordingly can solve this problem. Solutions such as these typically use a PXE boot configuration with a central repository for the different types of firmware for this environment. This does require extensive knowledge of the various components and a substantial effort with regard to development, but it ultimately leads to a highly flexible and scalable solution that enables users to update any of the hardware components.

We advise managing the firmware level consistently and following the hardware vendor’s recommendations, to avoid running into any interdependency issues. We also recommend that when users are acquiring new hardware, they look into the level of integration and the mechanisms that can be leveraged around managing their hardware. Especially in converged, shared platforms, availability and manageability are key to the success of an IT department.

**Systems Management and Backup**

Systems management and backup products integrate with ESXi via the VMware vSphere® APIs. The API-based partner integration model significantly reduces management overhead by eliminating the need to install and manage agents in the COS.

VMware has worked extensively with our ecosystem to transition all partner products to the API-based integration model of ESXi. As a result, BMC, CA, HP, IBM, EMC, NetIQ, Quest Software, Commvault, Vizioncore, Double-Take Software, SteelEye and Symantec are among the majority of systems management and backup vendors in the VMware ecosystem that have products that support ESXi today. Users employing an agent-based partner solution to integrate with ESX should check with their vendor to see if a newer version of the product supports ESXi.

VMware also includes backup capability with the vSphere product suite. VMware® Data Recovery is a robust, easy-to-deploy backup and recovery solution that businesses should consider to provide the first line of data protection for their virtual environment.

VMware Data Recovery enables:

- Full image backup of virtual machines
- Full and incremental recovery of virtual machines, plus recovery of individual files and directories

**Patching and Updating**

The patching and updating of ESXi enable flexibility and control. During the patching process, only the specific modules being updated are changed. The administrator can preserve any previous updates to other components. Whether installed on disk or in embedded Flash memory, ESXi employs a “dual-image” approach, with both the current and prior version present. When a patch is installed, the new image is constructed and overwrites the prior image. The current version becomes the prior version and the system boots off the newly written image. If there is a problem with the image, or if the administrator wants to revert to the prior one, the host is simply rebooted off the recent, good image.
VMware vCenter™ Update Manager is a vCenter plug-in patch-management solution for vSphere. It enables centralized, automated patch and version management for vSphere. It offers support for ESX and ESXi hosts, virtual machines and virtual appliances, enabling administrators to make their virtual infrastructure compliant with baselines they define. Updates that users specify can be applied to ESX and ESXi hosts, virtual machines and virtual appliances that can be scanned. With Update Manager, users can perform the following tasks:

- Scan for compliance and apply updates for guests, appliances and hosts
- Directly upgrade hosts, virtual machine hardware, VMware Tools and virtual appliances
- Install and update third-party software on hosts

Update Manager 5.0 enables users to apply offline bundle patches. These are patches that are downloaded manually from a VMware or third-party Web site, not hosted in an online depot. This is especially relevant to ESXi, because many important components, such as third-party driver updates and CIM provider updates, are often distributed only as offline bundles.

An alternative to Update Manager is the vCLI command “esxcli software.” This command applies software updates to ESX/ESXi images, and installs and updates ESX/ESXi extensions such as VMkernel modules, drivers and CIM providers. Unlike Update Manager, “esxcli software” works only on an individual host and does not monitor for compliance to baselines. However, it does not require vCenter Server to function.

vSphere PowerCLI also offers a similar solution to “esxcli software” with the “Install-VMHostPatch” cmdlet. This cmdlet can be used to install host patches located locally, from a Web location or in a host file system. It works only on an individual host and does not monitor for compliance to baselines. It also does not require vCenter Server to function.
Table 2 presents a summary of ESXi patching and updating options.

<table>
<thead>
<tr>
<th>PATCHING AND UPDATING TOOL</th>
<th>WHEN TO USE</th>
</tr>
</thead>
</table>
| VMware vCenter Update Manager | • Use when hosts are managed by vCenter Server. Update Manager is integrated with vCenter Server and provides a single pane of glass.  
• Use when monitoring for compliance against patching baselines is required.  
• Use when coordination with host maintenance mode is needed for vSphere Distributed Resource Scheduler (DRS) to perform an orderly evacuation of virtual machines from existing hosts. |
| “esxcli software” | • Use for one-off host upgrades.  
• Use in remote situations in which vCenter Server is not accessible.  
• Use when ESX and ESXi hosts are not managed by vCenter Server. |
| Install-VMHostPatch PowerCLI cmdlet | • Use for one-off host upgrades.  
• Use in remote situations in which vCenter Server is not accessible.  
• Use when ESX and ESXi hosts are not managed by vCenter Server.  
• Use as part of scripted solutions. |

Table 2. Considerations for Patching and Update Tool

**Image Builder**

vSphere 5.0 introduces the ESXi Image Builder CLI, a PowerShell snap-in (set of commands) that enables users to customize ESXi images. With the Image Builder CLI, users can create ESXi installation images with a customized set of updates, patches and drivers. The ESXi installation image comprises a series of separately packaged software components referred to as VMware Installation Bundles (VIBs). When an ESXi host is installed, the installer formats the boot device and extracts the VIBs off the installation media onto the boot device. After the VIBs have been extracted, the host boots and the hypervisor is loaded. There was a challenge with ESXi versions prior to 5.0 anytime an administrator needed to update the ESXi installation image to add or modify one of the VIB components (to add new device drivers for a new network adaptor, for example). In vSphere 5.0, the Image Builder CLI addresses this gap by providing users with the ability to customize their ESXi installation images.

Using the Image Builder CLI, customers place the ESXi VIBs into collections referred to as software depots. The administrator then uses the Image Builder PowerCLI to combine the VIBs from the separate depots together with the default ESXi installation image to create a custom image profile that can then be used to install their ESXi hosts. Multiple depots and image profiles can be maintained. For example, a separate image profile can be created for installing ESXi on rackmounted servers while another separate image profile is used for installing ESXi on blade servers.

**User Authentication**

Although day-to-day operations are done on vCenter, there are instances when users must work with ESXi directly, such as with configuration backup and log file access. To control access to the host, customers can have local users on an ESXi system. With ESXi 5.0, customers can configure the host to join an Active Directory domain, and any user trying to access the host will automatically be authenticated against the centralized user directory. Customers can also have local users defined and managed on a host-by-host basis and configured using the vSphere Client, vCLI or PowerCLI. This second method can be used in place of, or in addition to, the Active Directory integration.
Users can also create local roles, similar to vCenter roles, that define things that the user is authorized to do on the host. For instance, a user can be granted read-only access, which allows them only to view host information. Or they can be granted administrator access, which allows them both to view and to modify host configuration. If the host is integrated with Active Directory, local roles can also be granted to Active Directory users and groups. For example, an Active Directory group can be created to include users who should have an administrator role on a subset of ESXi servers. On those servers, the administrator role can be granted to that Active Directory group. For all other servers, those users would not have an administrator role. If an AD administrator creates a group with the name “VMware ESX Admins,” ESXi 5.0 automatically grants administrator access to this group, enabling the creation of a global administrators group. This operation can be overridden on individual ESXi hosts by assigning the “no access” role to the group “ESX Admins.”

The only user defined by default on the system is the root user. The initial root password is typically set using the direct console user interface (DCUI). It can be changed afterward using the vSphere Client, vCLI or PowerCLI. The root user is defined only locally. In other words, Active Directory does not manage the root password. It is possible to exclude the root user access by enabling Lockdown Mode. This is addressed in a later section of this paper.

Logging

As of vSphere 5.0, ESXi host logging is managed through the syslog facility, including vSphere High Availability (VMware HA) logs. Host logs can also be downloaded from the host by using the “Export Diagnostic Data” vSphere Client option.

The ESXi log file structure is different from that of ESX. Because there is no service console, there is also no need to have the same collection of files. With ESXi 5.0, multiple log files have been added. The following log files are most commonly used for troubleshooting purposes:

<table>
<thead>
<tr>
<th>PATH + LOG FILE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/log/vmkernel.log</td>
<td>All log entries are generated by the VMkernel.</td>
</tr>
<tr>
<td>/var/log/vmkwarning.log</td>
<td>A subset of the VMkernel logs that include only warnings and sysalerts events.</td>
</tr>
<tr>
<td>/var/log/hostd.log</td>
<td>Host management service (hostd = host daemon) log.</td>
</tr>
<tr>
<td>/var/log/sysboot.log</td>
<td>System boot log.</td>
</tr>
<tr>
<td>/var/log/fdm.log</td>
<td>VMware HA log file.</td>
</tr>
</tbody>
</table>

Table 3. Summary of Log Files

Proper log management is important for both troubleshooting and compliance. ESXi exposes logs from the host agent (hostd), vCenter agent (vpxa) and VMkernel (messages) by using a host syslog capability. Users can configure syslog to write logs to any accessible datastore via the following steps:

1. In the vSphere Client inventory, left-click the host.
2. Click the Configuration tab.
3. Click Advanced Settings under Software.
4. Select Syslog in the tree control.
5. In the Syslog.global.logDir text box, enter the datastore name and the path to the file where syslog will log messages, using the format “/[storage1]/<host>/logs/” Ensure that the directory is created beforehand.

You can optionally include the protocol and the port; for example, ss://hostname:514. UDP (default), TCP and SSL are supported.
By default, when a local boot disk is used, the ESXi host will write log files to the host’s scratch partition (/scratch/log). For USB/SD or boot-from-SAN installations, where no local datastore is available, it is recommended to use a shared 20GB VMware vSphere® VMFS volume with unique directories for each host. When using a shared VMFS volume, it’s important to monitor the disk space usage on this volume using the vCenter-provided alarm functionality.

**NOTE:** You might need to reboot the host for the changes to take effect. It is recommended to include the host name in the name of the folder.

Users can also configure syslog to forward log messages to a remote syslog server for enterprise central logging. Using a remote syslog server will simplify troubleshooting and ensure that log files are always accessible, even when an ESXi host has physically failed. Using a centralized syslog server also facilitates correlation between events on different hosts.

VMware offers two separate remote syslog solutions. The first solution is called the ESXi Syslog Collector, which can be installed on a supported Windows server using the vCenter Server installation media. The second solution is to use the syslog capabilities of the VMware vSphere® Management Assistant (vMA). After a remote syslog host has been set up, configuring the ESXi host to forward the logs is straightforward and can be done via the following seven simple steps:

1. In the vSphere Client inventory, left-click the host.
2. Click the **Configuration** tab.
3. Click **Advanced Settings** under **Software**.
4. Select **Syslog** in the tree control.
5. In the **Syslog.global.LogHost** text box, enter the name of the remote host where syslog data will be forwarded. If no value is specified, no data is forwarded.
6. Click **OK**.

You can configure ESXi host logging during installation when doing a scripted installation. Both the syslog advanced setting and the local datastore path setting can be configured through the use of “vim-cmd.” The following command is an example of how to set the path to a local datastore:

```
vim-cmd hostsvc/advopt/update Syslog.global.LogDir string “/[storage1] var/log/messages”
```

It is also possible to automatically create a unique directory using the name of the ESXi host under the specified “Syslog.global.Logdir” by enabling “Syslog.global.LogDirUnique” through the advanced setting in the vSphere Client or through “vim-cmd” using the following command:

```
vim-cmd hostsvc/advopt/update Syslog.global.LogDirUnique bool true
```

To correlate log events between hosts, it is very important to keep the date and time of your ESXi hosts in sync with an accurate time source. This is often required for compliance. It is also important when using the host to maintain accurate time on the guest virtual machines. VMware recommends synchronizing virtual machines with an NTP or w32tm server as described in VMware knowledge base article 1006427 and VMware knowledge base article 1318. ESXi has built-in capabilities for synchronizing with Network Time Protocol (NTP) time servers, which can be configured through the vSphere Client or through the shell, as shown in the automated installation script or through vSphere PowerCLI with the “Set-VMHostNTPServer” cmdlet.
ESXi Shell

ESXi Shell is a simple shell intended for advanced troubleshooting under the guidance of technical support. When remote command-line tools are not capable of addressing a particular issue, the ESXi Shell provides an alternative. Similarly to how the COS is used to execute diagnostic commands and fix certain low-level problems, the ESXi Shell enables users to view log and configuration files, as well as to run certain configuration and utility commands to diagnose and fix problems. ESXi Shell is not based on Linux. Rather, it is a limited-capability shell compiled especially for ESXi.

In addition to being available on the local console of a host, the ESXi Shell can be accessed remotely through SSH. Access to the ESXi Shell is controlled in the following ways:

- Both SSH and ESXi Shell can be enabled and disabled separately in both the DCUI and the vSphere Client or through vSphere PowerCLI.
- Any authorized user, not just root users, can use ESXi Shell. Users become authorized when they are granted the administrator role on a host (through Active Directory membership in a privileged group and through other methods).
- All commands issued in ESXi Shell are logged through syslog, providing a full audit trail. If a syslog server is configured, this audit trail is automatically included in the remote logging.
- A timeout can be configured for ESXi Shell (including SSH), so that after being enabled, it will automatically be disabled after the configured time. Changes to the SSH timeout will apply only to new sessions. Existing sessions will not be timed out, but any new session is prevented after the timeout period.

ESXi Shell is recommended for use primarily for support, troubleshooting and break-fix situations. It also can be used as part of a scripted installation, as described in a previous section. All other uses of ESXi Shell, including running custom scripts, are not recommended in most cases. Instead, users should use the vSphere vCLI or vSphere PowerCLI.
When ESXi Shell is enabled, the vSphere Client will display a warning sign on the ESXi host, as depicted in the following screenshot. If wanted, this warning can be disabled per host by completing the following procedure:

• Select the host.
• Click the **Configuration** tab.
• Click **Advanced Settings**.
• Go to **UserVars** and scroll to the bottom.
• Change the value of **UserVars.SuppressShellWarning** from 0 to 1.

*NOTE: This change impacts the warning for both local and remote (SSH) access to the ESXi Shell.*

![ESXi Shell Warning When Shell Is Enabled](image)

**Local Access and Lockdown Mode**

ESXi 5.0 provides the ability to fully control all direct access to the host via vCenter Server. After a host has been joined to vCenter Server, every direct communication interface with the host is configurable as an independent service in the Configuration tab for the host in the vSphere Client. This includes the following interfaces:

• DCUI
• ESXi Shell
• SSH

Each of these can be turned on and off individually.

![Local and Remote Access Services](image)
Access based on the vSphere API—for example, the vSphere Client, PowerCLI, vCLI, and so on—is normally governed by granting local privileges to specific users. The root user is the only one that has a permanent administrator role on the host. All other users must be explicitly granted a local role on the host in order to access it.

There are cases in which you would not want anyone to access the host directly at all, instead managing it exclusively through VMware vCenter Server. Lockdown Mode is a feature designed to provide this capability. When Lockdown Mode is enabled on the host, all direct remote access to the host is blocked, including

- Any vSphere API client
- ESXi Shell
- SSH

Even if Tech Support Mode is enabled, Lockdown Mode effectively overrides this by preventing any connection from succeeding. The only way to manage the host remotely is through vCenter Server. The interaction between the host and vCenter Server occurs through a special-purpose account called “vpxuser”; all other ordinary user accounts, including root, can no longer connect remotely.

For the special case of hardware monitoring through the CIM interface, monitoring software must obtain this hardware information directly from the host. To do this, the monitoring software must be programmed to obtain a special authentication ticket from vCenter Server. This ticket allows the software to obtain the information from the host through the vCenter Server “vpxuser” account on a one-time basis.

With Lockdown Mode enabled, the only direct access to the host that remains open is through the DCUI. This provides a way to perform limited administrative tasks outside of vCenter Server. The DCUI can also turn off Lockdown Mode, disabling it without going through vCenter Server. This might be useful if vCenter Server is down or otherwise unavailable and users want to revert to direct management of the host. To log in to the DCUI in Lockdown Mode, however, the root password is required. No other user can log in, even if they have been granted an administrator role.

In the extreme case, users might want to disable all direct access to the host. For example, they might want to prevent anyone with the root password from disabling Lockdown Mode and managing the host. In this case, they can take the additional step of disabling the DCUI for the host, through vCenter Server. After this is done, no direct interaction with the host, local or remote, is possible. It can be managed only through vCenter Server. If vCenter Server is down or otherwise unavailable, users cannot revert to direct management, because logging in to the DCUI is no longer possible. If the vCenter Server cannot be restored, the only way to revert to direct management is to reinstall the ESXi software on the host.

Lockdown Mode is not permanent. It can be disabled for any individual ESXi host at any time (provided that vCenter Server is running and able to connect to that host). The recommendation is that Lockdown Mode be used in ordinary, day-to-day operations but that it be disabled for a host if the need arises to interact with it directly. For example, if a troubleshooting situation is encountered and the tools provided by vCenter Server are not sufficient, Lockdown Mode should be disabled and more extensive diagnostics should be performed, using Tech Support Mode, for example.
Table 4 presents a summary of Lockdown Mode and its interaction with the various host access services.

<table>
<thead>
<tr>
<th>ACCESS MODE</th>
<th>NORMAL</th>
<th>LOCKDOWN</th>
<th>LOCKDOWN + DCUI DISABLED</th>
</tr>
</thead>
<tbody>
<tr>
<td>vSphere API (e.g., the vSphere Client, vSphere PowerCLI, vSphere vCLI, and so on)</td>
<td>Any user, based on local roles/privileges</td>
<td>None (except vCenter “vpuser”)</td>
<td>None (except vCenter “vpuser”)</td>
</tr>
<tr>
<td>CIM</td>
<td>Any user, based on local roles/privileges</td>
<td>None (except via vCenter ticket)</td>
<td>None (except via vCenter ticket)</td>
</tr>
<tr>
<td>DCUI</td>
<td>Root and users with administrator privileges</td>
<td>Root only</td>
<td>None</td>
</tr>
<tr>
<td>ESXi Shell</td>
<td>Root and users with administrator privileges</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>SSH</td>
<td>Root and users with administrator privileges</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 4. Summary of Lockdown Mode Effect on Local Access

**ESXi Firewall**

Although ESXi has offered security enhancements in Lockdown Mode, ESXi 5.0 introduces a firewall as well. Similar to the ESX firewall, the ESXi firewall can be managed from the host and cluster view of the vSphere Client. After selecting the host and choosing the **Configuration** tab, the VI administrator can check different services and firewall settings under **Security Profile**.
The following screenshot shows the security profile of a host, with details on available services and firewall rules. Administrators can start or stop any of these services and also provide access to these services through the firewall parameters.

**ESXi Firewall CLI**

For firewall configuration commands, a separate firewall namespace is provided. The “esxcli” command can be used to list all firewall rules. The “list” command (“esxcli network firewall ruleset list”) can be used to collect the information about the current firewall settings. The “set” command (“esxcli network firewall ruleset set”) enables users to configure firewall rules. Administrators can use this simple and intuitive command interface option to manage firewall rules.
vSphere PowerCLI can also be used to view and configure firewall rulesets using the following cmdlets:

- “Get-VMHostFirewallDefaultPolicy”
- “Get-VMHostFirewallException”
- “Set-VMHostFirewallDefaultPolicy”
- “Set-VMHostFirewallException”

Diagnostics and Troubleshooting

With ESXi 5.0, there are a variety of options for diagnosing problems with the server configuration or operation, as well as for fixing them. Different methods will be appropriate depending upon the situation. There are also VMware knowledge base articles with instructions on various issues.

The DCUI is the menu-driven interface available at the console of the physical server on which ESXi is installed or embedded. Its main purpose is to perform the initial configuration of the host (IP address, host name, root password) and diagnostics.

The DCUI has several diagnostic menu items:

- Restart all management agents, including:
  - hostd
  - vpxa
- Reset configuration settings, for example:
  - Fix a misconfigured switch
  - Reset all configurations to factory defaults
- Enable ESXi Shell, including:
  - ssh

Users can also point an ordinary Web browser to the host and view files, including:

- Log files
- Configuration files
- Virtual machine files

As an example, we will demonstrate how to view the log files of any given virtual machine. A user with an administrator role must provide credentials to use this feature. The procedure is as follows:

1. Open a browser and enter the URL http://<vCenter hostname>, where <vCenter hostname> is the IP or fully qualified domain name for the vCenter Server.
2. Click the Browse datastores in the VMware vCenter inventory link.
3. Provide administrative credentials when prompted.
4. Navigate the Web pages until you reach the appropriate datacenter, datastore and folder, as noted in step 1.
5. Click the link to the appropriate log file, and open it with your preferred editor.
In addition to the new “esxcli” command, a new “localcli” command has been added in vSphere 5.0. The “localcli” command is largely equivalent to the “esxcli” command, with a noted exception that it bypasses the local “hostd” process on the server. The “localcli” command is intended for situations where the ESXi host’s “hostd” daemon becomes unresponsive. It is recommended that you do not use the “localcli” command outside of the direction of VMware Global Support Services because it can result in host instability.

Other commands that have proven to be very valuable over time include

- “vscsiStats,” which provides detailed information on SCSI performance
- “nc,” which is based on the standard netcat utility
- “tcpdump-uw,” which is based on the standard tcpdump utility

Some commands that are used in troubleshooting scenarios are listed here for your convenience. This is not a comprehensive list. Rather, the following are just a few of the capabilities that the ESXi Shell offers:

- “vmkping -s 9000 <ipaddress>”

The command “vmkping” can be used to do basic network troubleshooting, but it is more often used to validate the operation of jumbo frames by adding the size of the packet, as shown in our example.

- “fdisk –l”

This lists all disk partitions and includes the type of the partition, where VMFS partitions are labeled as “fb.”

- “vim-cmd hostsvc/maintenance_mode_enter”

Maintenance Mode can be entered from the command line by using “vim-cmd.”

- “vim-cmd hostsvc/maintenance_mode_exit”

Maintenance Mode can be exited using this command.

- “esxcli vm process list”
- “esxcli vm process kill -world-id=<world-id> --type=<soft, hard, force>”

The first command provides a list of all the virtual machines currently registered on the host. The second command enables you to power off a virtual machine.

These commands are just examples of what is possible with the ESXi Shell. We recommend that you avoid enabling access to the ESXi Shell unless absolutely needed and that you disable access when it is no longer needed. In general, troubleshooting workflows are similar to those with VMware ESX, due to the feature set of ESXi Shell.

One thing that has changed in terms of diagnosing problems in vSphere 5.0 is the way core dumps can be collected. A core dump can be used to determine the reason for system failure. With ESX, the core dump often was placed on the local disk, which in the case of a total failure frequently made it impossible to do a root cause analysis. With ESXi, there are the following two options for managing core dumps:

- Create a diagnostic partition on SAN storage. Each host must have a diagnostic partition of 100MB. If multiple hosts share a SAN, configure a diagnostic partition with 100MB for each host.
- Install and configure ESXi Dump Collector. New in ESXi, the Dump Collector service enables you to send core dumps to a network server. It is especially useful for determining reasons for failure of ESXi hosts provisioned with Auto Deploy.
You can use “esxcli system coredump” to configure a remote or local core dump partition. You can also use the network configuration host profile to set up hosts to use ESXi Dump Collector. In either case, you can apply to other hosts the host profile of a host that uses ESXi Dump Collector.

Figure 15. Listing the Coredump Partition
**Summary**

The following table provides a summary of the tasks traditionally performed in the service console of VMware ESX and the functional equivalents for VMware ESXi.

<table>
<thead>
<tr>
<th>TASK</th>
<th>VMWARE ESX</th>
<th>VMWARE ESXi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access local files: VMFS</td>
<td>Console commands to browse datastores and virtual machine files</td>
<td>• Remote command-line interface commands to list and retrieves files</td>
</tr>
<tr>
<td>files, configuration files,</td>
<td></td>
<td>• vSphere client datastore browser for VMFS files downloads and uploads</td>
</tr>
<tr>
<td>log files</td>
<td></td>
<td>files</td>
</tr>
<tr>
<td>Manipulate virtual</td>
<td>• Advanced configuration done in the vSphere client</td>
<td>• Advanced configuration done in vSphere Client</td>
</tr>
<tr>
<td>machine files (for example,</td>
<td>• Console commands to modify virtual machine files</td>
<td>• Remote command-line interface commands to list and retrieves virtual</td>
</tr>
<tr>
<td>modify .vmx)</td>
<td></td>
<td>machine files</td>
</tr>
<tr>
<td>Backup</td>
<td>• Virtual machine backup: agents in service console, VMware Data</td>
<td>• Virtual machine backup: VMware</td>
</tr>
<tr>
<td></td>
<td>Recovery or third-party backup products</td>
<td>Data Recovery or third-party backup products</td>
</tr>
<tr>
<td></td>
<td>• VMware ESX backup: uses agents in the service console, creates archive</td>
<td>• ESXi backup: single small backup file created via vSphere vCLI command</td>
</tr>
<tr>
<td></td>
<td>of service console files or performs a scripted reinstall</td>
<td>“vicfg-cfgbackup” or vSphere PowerCLI cmdlet “Get-VMHostFirmware”</td>
</tr>
<tr>
<td>Hardware monitoring</td>
<td>• Agents in service console</td>
<td>• CIM-based framework</td>
</tr>
<tr>
<td></td>
<td>• SNMP</td>
<td>• SNMP</td>
</tr>
<tr>
<td>Patching and updating</td>
<td>• Update Manager</td>
<td>• Update Manager</td>
</tr>
<tr>
<td></td>
<td>• RPM-based third-party tools</td>
<td>• vCLI command “vhostupdate”</td>
</tr>
<tr>
<td>Automated deployment</td>
<td>Red Hat Kickstart</td>
<td>• ESXi scripted installation (analogous to Red Hat Kickstart)</td>
</tr>
<tr>
<td>Troubleshooting or</td>
<td>• “esxcli”</td>
<td>• Remote command-line interface commands</td>
</tr>
<tr>
<td>support</td>
<td>• “esxcfg-*” commands</td>
<td>ESXi Shell</td>
</tr>
<tr>
<td>Advanced configuration</td>
<td>Edits configuration files (for example, hostd.conf) directly</td>
<td>• Remote command-line interface commands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to list and retrieve VMware ESXi configuration files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Edits files in Tech Support Mode directly</td>
</tr>
<tr>
<td>Logging</td>
<td>Remote syslog in service console</td>
<td>Built-in remote syslog client</td>
</tr>
<tr>
<td>Performance monitoring</td>
<td>• vSphere client</td>
<td>• vSphere Client</td>
</tr>
<tr>
<td></td>
<td>• “esxtop” in service console</td>
<td>• vSphere vCLI command “resxtop”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“esxtop” in Tech Support Mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• vSphere PowerCLI cmdlet “Get-EsxTop”</td>
</tr>
<tr>
<td>Reporting and auditing</td>
<td>• Service console scripts</td>
<td>• Remote command-line interface commands</td>
</tr>
<tr>
<td></td>
<td>• Log files</td>
<td>to list and retrieve log files, configuration and settings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• vSphere Client option to export diagnostic data</td>
</tr>
</tbody>
</table>

*Table 5. Comparison of Management Capabilities in VMware ESX and VMware ESXi*
VMware ESXi Editions

VMware ESXi architecture is offered as a part of all vSphere product editions, with each successive edition offering greater functionality. At the entry level, VMware offers the vSphere Hypervisor, which is a free virtualization product. Certain ESXi features are limited in this edition, as outlined in Table 6. All other paid editions of vSphere lift these feature restrictions. However, even though the host-level features are not limited in all paid editions, many advanced features, such as VMware DRS and VMware HA, are still available only in higher-license versions.

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>vSPHERE HYPERVERSOR</th>
<th>VMWARE ESXi ENTERPRISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP monitoring</td>
<td>Not supported</td>
<td>Full functionality</td>
</tr>
<tr>
<td>VMware Consolidated Backup (VCB) and VMware Data Recovery (vDR) tool</td>
<td>Not available</td>
<td>Both applications are available</td>
</tr>
<tr>
<td>vSphere vCLI</td>
<td>Limited to read-only access</td>
<td>Full functionality</td>
</tr>
<tr>
<td>vSphere PowerCLI and VMware vSphere® SDK for Perl</td>
<td>Limited to read-only access</td>
<td>Full functionality</td>
</tr>
</tbody>
</table>

Table 6. Comparison of VMware ESXi Editions

An administrator who has deployed vSphere Hypervisor can enjoy the benefits of virtualization with VMware ESXi within the feature limits. However, the deployment can be upgraded to a more fully featured version of vSphere at any time without having to uninstall or reinstall the ESXi software. The additional capabilities are activated simply when the proper license key is provided, either in the host configuration or in VMware vCenter Server.

References

VMware ESXi Configuration Guide

VMware ESXi Installable and vCenter Server Setup Guide

VMware vSphere Command-Line Interface Installation and Scripting Guide
http://www.vmware.com/support/developer/vcli/

VMware vSphere Command-Line Interface Reference
http://www.vmware.com/support/developer/vcli/

VMware ESXi Upgrade Center
http://www.vmware.com/go/UpgradeToESXi

VMware ESXi Chronicles Blog
http://blogs.vmware.com/esxi/