Managing VMware ESXi

vmware®

Table of Contents

Managing VMware ESXi

Introduction

VMware® ESXi is the next-generation hypervisor, providing a new foundation for virtual infrastructure. This innovative architecture operates independently from any general-purpose operating system, offering improved security, increased reliability, and simplified management. This compact architecture is designed for integration directly into virtualization-optimized and certified server hardware, enabling rapid installation, configuration, and deployment.

Functionally, ESXi is equivalent to VMware ESX. However, the Linux-based service console has been removed, reducing the footprint to less than 32MB of memory. The functionally of the service console is replaced by remote command line interfaces and adherence to system management standards. In the simplest implementation, ESXi is embedded directly into the firmware of select server models from various vendors, allowing the server to boot directly into ESXi. (Go to *http://www.vmware. com* for the latest information on vendors and models.)

Because ESXi no longer includes a service console, many of the management activities performed on the ESX 3 platform — for example, configuring user access to the service console and administering management agents running in it — are no longer necessary. Other management tasks previously done in the service console are now performed in one of several ways:

- Using the Virtual Infrastructure Client (VI Client), which provides a Windows-based graphical user interface for interactive configuration of the platform. The VI Client has been enhanced to provide capabilities that were previously available only in the service console.

- Using the remote command line interfaces, new interfaces that enable scripting and command-line–based configuration of the platform from a Linux or Windows-based server, via an encrypted and authenticated communication channel.

- Using external agents that leverage well-defined APIs, such as the VIM API and the Common Information Model (CIM) management standard.

In addition, you can manage ESXi using VirtualCenter, just as you would any ESX 3 system. Distributed virtualization features, such as VMotion and VMware DRS, are designed to work exactly the same on ESXi — and also in a mixed environment of ESX 3 and ESXi systems. (Check the product documentation and release notes to see the latest status of support for particular features.) VirtualCenter presents both types of systems in the VI Client user interface in almost the same way; certain features unique to ESXi management appear for hosts equipped with that version.

In many respects, the functionality of an ESXi system is the same as for ESX 3. However, the architecture of ESXi points the way to a new management model for a virtualized infrastructure. The core aspects of this model are:

- Datacenter infrastructure based on stateless, interchangeable computing nodes

- Centralized management, including administration and policy

- Communication with the system using well-defined and standardized APIs instead of unstructured interactive sessions that are difficult to lock down and audit

In short, ESXi represents the continuation of a long term trend to move management functions out of the service console and into remote or central tools. The rest of this paper covers the management of an ESXi system and lays out the characteristics of this new management model.

Deployment

ESXi is designed for distribution in various formats, including directly embedded in the firmware of a server or as software to be installed on a server's boot disk. When you work with the embedded version, your experience when you power on an ESXi system is analogous to the way you configure a home networking router. The device uses detection, discovery, and sensible defaults to perform as much configuration as possible automatically. In other potential delivery form factors, such as preloaded onto the internal disk of a server, the startup process would be slightly different. This section focuses on the deployment experience when you are using ESXi integrated into the server hardware.

The ESX/ESXi platform — the hypervisor — runs immediately when the system boots and automatically detects network setting using DHCP. This process configures the following network properties:

• IP Address, netmask, and gateway associated with the management agent

• Host name and DNS server

If the primary network interface of the server is not on a network with DHCP services, the network properties remain unconfigured.

After the boot process, the console of the system displays a welcome screen with a log-in prompt. After logging in, you see a menu-driven configuration screen, shown in Figure 1, where you perform the initial configuration of the system. On this screen, you can perform these two basic configuration tasks:

• Configure networking properties, if not already done automatically.

• Set the administrator (root) password. By default, the administrator password is not set.

Because the administrator password is not set initially, it is important to make sure the server is on a trusted network when it is powered on for the first time. Alternatively, you can power on the server before it is connected to any network, set the administrator password, then connect to a network. You can then configure networking, either manually or by restarting the networking, thereby allowing automatic configuration.

After configuring the networking properties and setting the administrator password, you can perform all subsequent management activities remotely. The remaining menus of the console allow debugging activities, such as restarting the management agent and reviewing logs.

Large-Scale Standardized Deployment
In some large-scale deployments of ESX, administrators include post-install scripts that automatically perform detailed platform configuration when the installation of ESX is complete. Combined with PXE-boot–based installation or whole disk imaging, customers have used this for fully automated large-scale standardized deployments of ESX.   When ESXi is embedded in the firmware of the server, no additional hardware-level configuration is needed.

Once you have configured networking and the administrator passwords, you can use remote command line interfaces (RCLIs) scripts to configure the virtualization layer, including virtual networks and storage devices. Typical tasks performed in a traditional post-install script that you can implement using an RCLI script include:

• Create port groups

• Set up VMkernel network services to support VMotion or iSCSI

• Configure NIC teaming

• Configure HBA multipathing

• Set up NTP

The remote command line interfaces are described in greater detail in the following section.
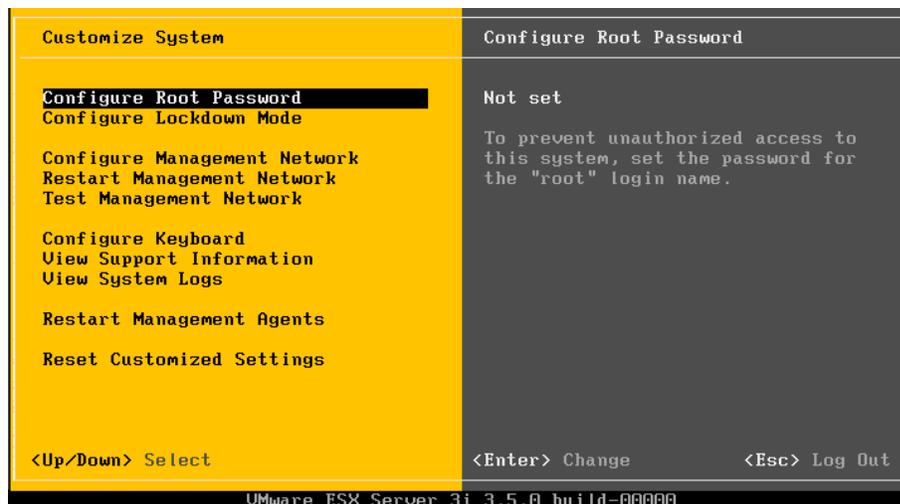
```
Customize System                    Configure Root Password

Configure Root Password             Not set
Configure Lockdown Mode
                                    To prevent unauthorized access to
Configure Management Network        this system, set the password for
Restart Management Network          the "root" login name.
Test Management Network

Configure Keyboard
View Support Information
View System Logs

Restart Management Agents

Reset Customized Settings




<Up/Down> Select            <Enter> Change          <Esc> Log Out
            VMware ESX Server 3i 3.5.0 build-00000
```

Figure 1: Welcome screen of the Direct Console User Interface

Interactive and Scripted Management

For management functionality, ESXi relies on remote tools instead of a service console. Depending on the situation, there are two methods for configuring an ESXi host:

• Using the VI Client, for graphical configuration

• Using the remote command line interfaces, for command-line–based and scripted configuration

These two methods provide almost equivalent functionality for most configuration actions. The choice of which to use is largely based upon experience and preference. Someone who is new to ESX/ESXi, or who requires a simpler, graphical interface, should use the VI Client. For those who are more experienced with ESX/ESXi and who prefer the speed and flexibility of command-line tools or scripting, the RCLIs are recommended.

You can connect the VI Client to VirtualCenter or directly to an ESXi host. Similarly, you can use a terminal or command prompt session on a remote computer to send commands either directly to a specified ESXi host or to VirtualCenter Server, which then executes the command on your behalf on the specified ESXi host under its management using a private protocol. If you use the VI Client or RCLI via VirtualCenter, you must be known to VirtualCenter as well as have the necessary privileges. If you connect directly to an ESXi system, your credentials must map to a local user on the system, who must also have the necessary privileges locally defined on that system to run each command.

The privilege model for ESX/ESXi is essentially a subset of the VirtualCenter privilege model. You can define custom roles on ESX/ESXi that allow individual tasks, such as managing virtual machines. The only difference is that tasks that apply only to

multiple ESX/ESXi hosts in the context of VirtualCenter, such as cluster administration, do not apply to ESX/ESXi. In either case, you can define custom roles that aggregate privileges in whatever manner is suitable for your needs.

The set of users and roles defined on VirtualCenter is totally independent of any scheme defined on the ESX/ESXi hosts, which in turn are totally independent of each other. You must perform any synchronization between the individual entities manually. Thus, it is generally better to use VirtualCenter because it makes it easy to centralize all user management. You can, however, still make use of local user accounts on the ESXi host, either for situations in which VirtualCenter is unavailable, or when you wish to manage a small environment with only a few hosts without using VirtualCenter.

VI Client

The VI Client is the primary user interface to VMware virtual infrastructure. It provides both a console to operate virtual machines and an administrative interface for ESX/ESXi hosts. The same tool also provides an interface for VirtualCenter, the VMware tool for centrally managing multiple ESX/ESXi hosts.

VI Client has been enhanced to provide some functionality that was previously available only via the service console. This added functionality includes:

• Time configuration, including setting the date and time, as well as configuring NTP (Network Time Protocol) servers to synchronize the time with an external source., as shown in Figure 2.
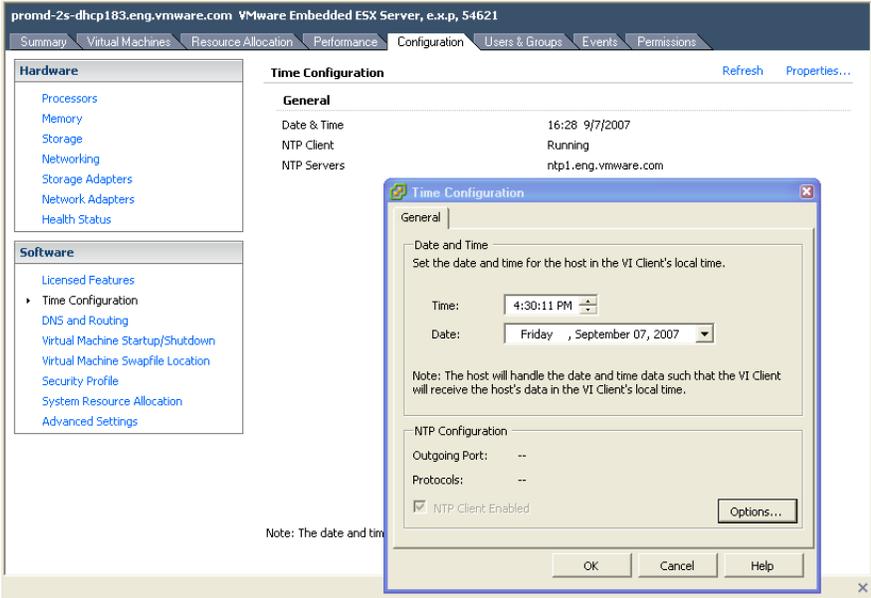

Figure 2: Configuring NTP using the VI Client

• Enhanced datastore browsing, which allows you to manage any VMFS datastore associated with this host, including operations such as uploading and downloading individual files to and from the local VI Client system, moving files between datastores, and deletion of files and folders.

All other the capabilities of VI Client are still available. For example, you can use Export Diagnostic Data to retrieve all the system log files from the ESXi host when you want to view these files. In particular, the user and group management capabilities, which allow you to create and manage local user accounts and user groups, have greater importance with ESXi, because you can no longer log in to the service console to create users and groups. You still associate these users with roles on managed objects, just as you normally would on a standalone ESX 3 system.

If you manage an ESXi system with VirtualCenter, all of the additional functionality described above is present in the VI Client interface for that host, with the exception of user and group management. You see this difference because all user management occurs on VirtualCenter itself, thus obviating the need for local user and group management on individual ESXi systems. Therefore, you can have the same level of management functionality you are accustomed to with ESX 3.

If you want to enforce central audited management through VirtualCenter, you can prevent root-level access by the VI Client. By enabling a feature called lockdown mode, you disable all remote root access. If you have defined no local users, the only way to manage the server is through VirtualCenter, which connects through the special user named vpxuser via an encrypted password. Only the VirtualCenter server can decrypt that password. If you have defined any local users, they can continue to connect directly to the host via the VI Client when lockdown mode is enabled. You can enable and disable lockdown mode only via VirtualCenter itself or by using the DCUI on the local console of the server.

The purpose of using lockdown mode is to prevent anonymous root-level interaction with the system. It effectively requires that all interaction occur either via a user known to VirtualCenter or via a named user account defined on the server locally. Thus all users must be granted explicit permissions to perform tasks, and their actions are fully audited.

Remote Command Line Interfaces

The remote command line interfaces enable scripting and command-line–based configuration of the platform from a Linux or Windows-based server. It provides a set of commands that allow you to perform administrative tasks over an encrypted and authenticated communication channel. The commands themselves are based on the `esxcfg-*` commands that are found in the service console of ESX. The syntax for using them is exactly the same as the service console equivalents, with the addition of options for authentication and for indicating on which server to perform the actions. The commands are available for both Linux and Windows; all you need to do is install the remote command line interfaces package on any system you will use to execute these commands.

As with the VI Client, the RCLI commands can be invoked directly on an ESXi system or through VirtualCenter, which then forwards the command request to the specified ESXi host under its management. Similarly, remote root access directly to the system through the RCLI is not possible when lockdown mode is enabled on the server, thus requiring that all RCLI commands go through VirtualCenter or are executed by a named account defined locally on the server.

Most of the `esxcfg-*` commands found in the service console are reproduced in the remote command line interfaces. The exceptions are commands used to manage services for the service console itself; because the service console has been eliminated, these commands are not relevant for an ESXi system.

Almost all RCLI commands are named with the initial prefix `vicfg-*`. However, for compatibility with previous scripts, all RCLI commands are aliased to equivalent commands with the prefix `esxcfg-*`, with the exception of those that are newly introduced with ESXi. The aliases are provided to ease the transition from COS-based management, allowing you to use existing scripts with a minimum of modification.

The remote command line interfaces commands can be divided roughly according to their purpose.

• Host configuration: `vicfg-advcfg`

• Storage configuration: `vicfg-nas`, `vicfg-swiscsi`, `vicfg-mpath`, `vicfg-rescan`, `vicfg-vmh-badevs`, `vmkfstools`

• Network configuration: `vicfg-vswitch`, `vicfg-vmknic`, `vicfg-route`, `vicfg-nics`

• Maintenance and patch: `vicfg-dumpart`

• Monitoring: `resxtop` (remote equivalent of `esxtop`)

In addition, several commands were created specifically to address functionality that previously was configurable only through the service console.

- `vicfg-ntp`: specify NTP servers
- `vicfg-snmp`: configure SNMP
- `vicfg-syslog`: specify a remote syslog server, for remote collection of log files
- `vicfg-cfgbackup`: a utility to back up and restore an embedded host's configuration, or reset the host, returning it to factory defaults

Backup is discussed in a later section.

For all the remote command line interfaces commands, information about the session must be specified. This includes:

- The user who is executing the command. If the user is not known to the server, or if the password is incorrect, the command is rejected.
- The server on which to authenticate the user, whether it is the VirtualCenter host or the individual target ESX/ESXi host. If connecting to VirtualCenter, you must also specify the target host on which you want to perform operations. If you are not connecting through VirtualCenter, the target host is assumed to be the same as the authenticating host.

Every command is associated with a specific privilege. Thus, even if the user is authenticated successfully, if the user does not have the proper permission to execute a particular action, the command is rejected. The same is true for file access; if the user is not authorized to view configuration files, for example, then any file access command is rejected. As mentioned earlier, these privileges can be defined either on VirtualCenter or on the individual ESX/ESXi hosts, and each set of privilege definitions is totally independent of the others.

To use a remote command line interfaces command, type the name of the command, and before typing the usual options for the command, type the remote host and authentication information. For example , to run a command directly on an ESXi host:

```
vicfg-nics --server=esx-host1
--username=joe_admin -list
```

where the strings in italics are those required for the session. After you enter the command, you are prompted for the password for the specified user. If you enter the correct password, the command executes.

To specify a VirtualCenter host as the authenticating server and the ESX/ESXi host as the target, enter the command as shown in the following example:

```
vicfg-nics --server=vc-host
--username=joe_admin
--vihost=esx-host1 -list
```

To facilitate scripting commands, there is an option to save authentication information to a session file. This session file encodes the user name and password information in a token, a method that avoids storing this information in plain text. By default, this token expires after 30 minutes. After authenticating once to generate the session file, you can run any remote command line interfaces command without providing the user name and password.

File Management

The ESXi system is designed with a simple in-memory file system used to hold configuration files, log files, and staged patches. For ease of administration, the structure of the file system is designed to be similar to that used in the service console. For example, ESX/ESXi configuration files are found in `/etc/vmware` and log files in `/var/log/vmware`. Files can also be saved in any VMFS datastore accessible by a host, either on a local disk or on a remote file server.

Files and directories are broadly classified into three groups:

- Datastore — Datastore files and directories accessible to a host
- Host — Host configuration files, which reside on the local file system of the server
- Temp — The `/tmp` directory and files inside that directory

You can manage these files using `vifs`, an RCLI utility for managing both local files on the ESXi system and files located in a datastore associated with that server, either on the local disk or on a shared file system.

For reasons of security, different capabilities are enabled for the different types of files.

- Datastore — You can upload and download datastore files and directories accessible to a host. You can also view directory listings. Additional functionality provided for VMFS datastores includes: `move`, `copy`, `mkdir`, `rmdir`, and `remove`.
- Host — You may download host configuration files for local manipulation, then uploaded them again. To avoid corrupting files that are not user-modifiable, there is a fixed list of files that you can upload and download. Any host file that is not on this list cannot be accessed. Furthermore, all files are specified without their full path, regardless of their actual location

in the local file system hierarchy. It is not possible to view the file system hierarchy.

• Temp — The `/tmp` directory is meant only to hold image files uploaded prior to flashing the firmware with the updated image. Thus, only upload of files to this directory is supported.

Remote Command Line Interfaces and ESX 3

The remote command line interfaces also work with ESX 3 hosts, meaning that you have the choice of staying with your current service console scripts or moving to the new management model of RCLI-based scripts. The advantage of migrating to the newer model is that the RCLI automatically imposes a fine-grained, auditable, and enforced privilege model, particularly if used through VirtualCenter. By using the RCLI with ESX/ESXi, you can reduce or eliminate the need for most personnel to have access to the service console, thus greatly improving security by reducing exposure to risk. In a mixed environment of ESX and ESXi, you can develop a set of scripts and procedures based on the RCLI, and use it across both types of systems.

Third-Party Management Applications

Because the service console provides access to low-level hardware metrics as well as higher-level status information, systems management software vendors created agents that ran in the service console in order to collect information on a host and forward it to a central management server. In the absence of a service console, this functionality is replaced by several different remote APIs. VMware is working actively with third-party companies to help them port their existing ESX management agents to work with this new management model.

Common Information Model

The Common Information Model (CIM) is an open standard of the Desktop Management Task Force that defines how computing resources can be represented and managed. It enables a framework for agentless, standards-based monitoring of hardware resources for ESXi. This framework consists of a CIM object manager (CIMOM), often called a CIM broker, and a set of CIM providers.

CIM providers are used as the mechanism to provide management access to device drivers and underlying hardware. Hardware vendors can write providers to provide monitoring and management for their devices. VMware is also writing providers that implement monitoring of server hardware, the underlying storage infrastructure used to support virtual machines, and virtualization-specific resources. These providers run inside the ESXi system, and hence 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, including WS-MAN. Figure 3 shows a diagram of the CIM management model.

VI API

The VMware Virtual Infrastructure API (VI API) provides a powerful interface for developing applications to integrate with VMware Infrastructure. The VI API enables your program or framework to invoke VirtualCenter Web Service interface functions on VirtualCenter to manage and control ESX/ESXi. It allows developers to provide tools that:

• Configure virtual machines and hosts

• Provide discovery and inventory functions in virtual environments.

• Access performance data

• Control virtual machine operations, such as configuration changes and migrations or virtual machine state

The VI API is fully synchronized with management tools via the remote command line interfaces, VI Client, and VirtualCenter.
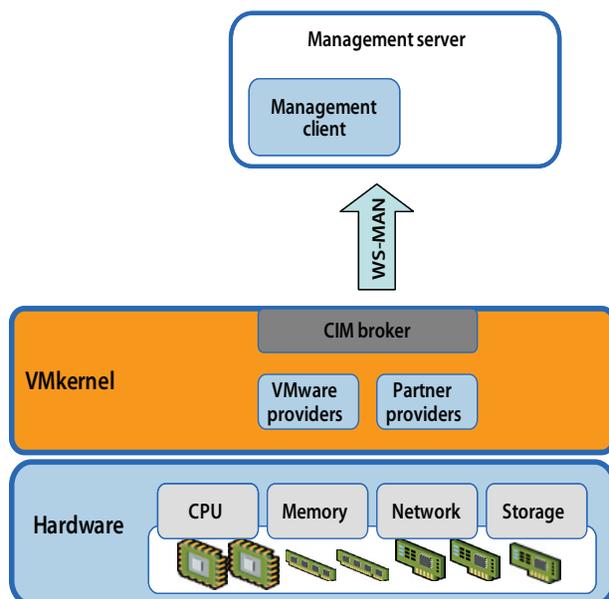

Figure 3: CIM Management model

Indeed, all three of these are built on the VI API. Furthermore, this API works for VirtualCenter as well as directly with ESX/ESXi. The only difference is that certain functions that pertain to multiple hosts, such as VMotion, are implemented only in VirtualCenter. Figure 4 illustrates how the VI API fits into the VMware Infrastructure management framework.

The third-party management products that can integrate with ESXi through the VI API include tools for:

- Monitoring
- Performance management
- Process automation
- Provisioning, change management, and configuration management

The VI SDK provides developers with a full environment for creating applications that interact with ESXi in a variety of programming languages. Resources for developers who want to use the VI API are available on the VMware Web site at *http://www.vmware.com/support/developer/vc-sdk/*.

SNMP

Simple Network Management Protocol (SNMP) allows management programs to monitor and control a variety of networked devices. Managed devices run SNMP agents, which can provide information to a management program in at least one of the following ways:

- In response to a `get` operation, which is a specific request for information from the management system.
- By sending a notification, which is an alert sent by an SNMP agent to notify the management system of a particular event or condition. VirtualCenter Server, ESXi, and ESX 3 each has an SNMP agent. The agents provided with each product have differing capabilities.

Even with the introduction of the CIM framework, hardware monitoring through SNMP continues to be supported by ESXi, and any third-party management application that supports SNMP can be used to monitor it. For example, Dell OpenManage IT Assistant (version 8.1 or later) has ESXi MIBs pre-compiled and integrated, allowing basic inventory of the server and making it possible to monitor hardware alerts such as a failed power supply. SNMP also lets you monitor aspects of the state of the VMkernel, such as resource usage, as well as the state of virtual machines.

ESXi ships with an SNMP management agent different from the one that runs in the service console of ESX 3. Currently, the ESXi SNMP agent supports only SNMP traps, not `get`s. SNMP write operations are not supported by ESXi. This agent is off by default. To use this agent, you must enable the SNMP service, specify at least one community, and configure a trap destination using the remote command line command `vicfg-snmp`.
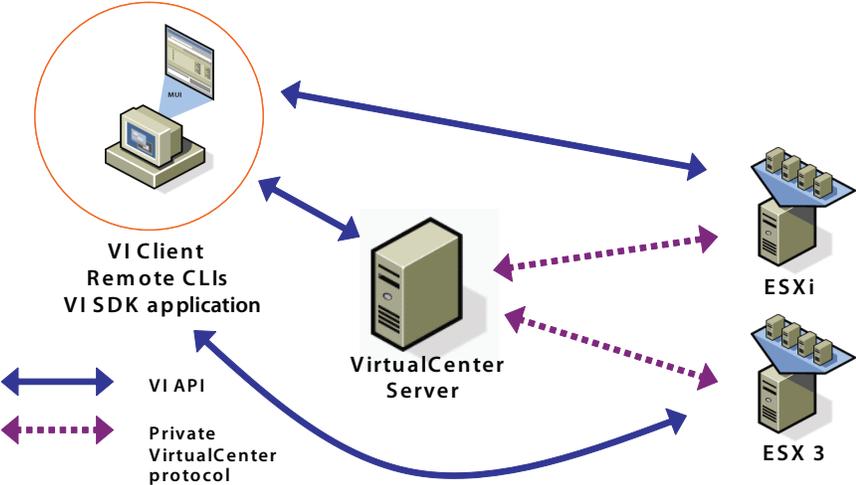

Figure 4: The VI API in the VMware Infrastructure management framework

9

System Image Design

Figure 5 shows a diagram of the contents of the ESXi system image. Regardless of whether the image exists on flash memory or on the hard drive of a computer, the same components are present:

• A 4MB bootloader partition, which runs upon system boot up.

• A 48MB boot bank, which contains the 32MB core hypervisor code, along with a second alternate boot bank of the same size. The reason for two boot banks is explained below.

• A 540MB store partition, which holds various utilities, such as the VI Client and VMware Tools images.

• A 110MB core dump partition, which is normally empty but which can hold diagnostic information in case of a system problem.

The ESXi system has two independent banks of memory, each of which stores a full system image, as a fail-safe for applying updates. When you upgrade the system, the new version is loaded into the inactive bank of memory, and the system is set to use the updated bank when it reboots. If any problem is detected during the boot process, the system automatically boots from the previously used bank of memory. You can also intervene manually at boot time to choose which image to use for that boot, so you can back out of an update if necessary.

At any given time, there are typically two versions of VI Client and two versions of VMware Tools in the store partition, corresponding to the hypervisor versions in the two boot banks.
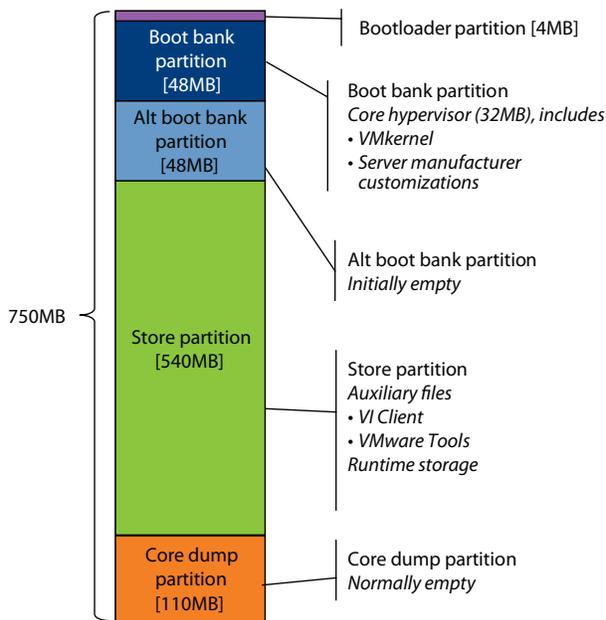

Figure 5: Contents of the ESXi system image

The specific version to use is determined by which boot bank is currently active.

The core hypervisor code also can contain custom code provided by server vendors (OEMs) that provides additional functionality, such as hardware monitoring and support information. These customizations would be present, for example, if ESXi had been obtained in embedded form from the server manufacturer or if a custom version of ESXi was installed onto the hard drive. Any update to an existing ESXi installation automatically incorporates the proper update to this custom code. The process is described in greater detail in the following section.

Patching and Upgrading

The process for patching and upgrading systems is significantly simplified in ESXi. This section describes the process for the embedded version of ESXi.

With ESX 3, patches are fine-grained, and you use `esx-update` to install multiple individual ESX patches.

With ESXi, when patches or updates are needed, you replace the whole system image. Because the total code base is so small, replacing a whole image is more efficient and simpler than managing many individual patches. This is analogous to the process of managing the BIOS or other firmware of a server.

As mentioned previously, the new version of the hypervisor code is loaded into the inactive boot bank. In addition to the core hypervisor code, each ESXi patch includes updated versions of VI Client and VMware Tools. This is the reason the patch bundles themselves are much larger than the 48MB size of the boot bank. The updated versions of these two components are placed in the store partition. Each ESXi patch also includes updates to server vendor customizations. Current updates to all existing customizations from every server vendor are included with each patch. When the patch is installed, the system is queried to determine which (if any) customization is appropriate for the specific system being updated, and the corresponding customization is installed. The customizations that are not relevant to the server are not installed.

There are a number of tools you can use to install patches onto ESXi hosts. You can use either the remote command line interfaces or the VMware Update service to install the new image directly on the system or use VMware Update Manager from VirtualCenter to manage updates across a set of systems.

You can use the `vihostupdate` RCLI command for maintenance of your ESXi hosts. The command can install software updates, enforce software update policies, and track installed software. Before you can update the firmware of an ESXi host, the update bundle must be locally accessible on the machine where you run the `vihostupdate` command. If the update

bundle is not accessible, the update process first pushes the update bundle to the host, then performs the update. If the update bundle is already accessible — for example, because it is installed locally or available on a remote datastore that the host can access — the update process does not have to push the update bundle to the host. Each update consists of a descriptor file and a set of packages. The descriptor controls the installation process and checks that requirements are met. For example, you might be required to power off all virtual machines running on the server you are preparing to update, or you might need to reboot the server after the update.

The new VMware Infrastructure Update service is a graphical, Windows-based tool that is an optional plug-in for the VI Client. VMware Update service allows you to learn about, download, and install maintenance and patch releases. The VMware Update Service provides security, stability, and feature enhancements for VMware Infrastructure. The VMware Update service periodically checks for new updates that are applicable to ESXi hosts connected to the VI Client. If it discovers new updates, VMware Update downloads the firmware and the companion software in the background and caches the downloaded updates in a local repository. You can then apply the updates or schedule them for automatic installation at a later time.

The previous two methods allow you to update ESXi hosts one by one. For bulk upgrades of multiple systems, VMware Update Manager provides an automated patch management solution for online ESX/ESXi hosts and online as well as offline Microsoft and Linux virtual machines. VMware Update Manager downloads information about the latest patches and updates for ESX/ESXi directly from VMware. It scans the state of the physical VMware ESX/ESXi hosts as well as select guest operating systems, compares them with baselines set by the administrator, then applies updates and patches to enforce compliance to baselines. VMware Update Manager supports either manual or scheduled patching of noncompliant machines. If a reboot is required on a manual patch or update, you have the option of rebooting immediately or delaying system restart by up to 60 minutes.

When used to update ESX/ESXi hosts, VMware Update Manager enables nondisruptive host patching in conjunction with VMware DRS. VMware Update Manager puts hosts in maintenance mode, one by one, and live migrates virtual machines to other hosts while patching. When a host is patched, VMware Update Manager migrates the virtual machines back to that original host, then moves on to patch the next host in the cluster, thus allowing for fully automated rolling upgrades.

Backup

The state of an ESXi system is fully described by a handful of configuration files. These files control such functions as configuration of virtual networking and storage, SSL keys, server network settings, and local user information Although these configuration files are all found in the in-memory file system, they are also periodically copied to persistent storage. For example, in the embedded version of ESXi, there is a small part of the server firmware that is designated as read-write. In case of sudden power loss, you can reboot the server and it is restored to the exact configuration of the last copy.

You can also download a backup file that contains all the state information using the `vicfg-cfgbackup` RCLI command. This allows you to replicate the state of an ESXi system onto another similar system. You can create backups of your server configuration, and if a server fails catastrophically, you can easily replace it with an identical unit, then bring that new unit to the same state by restoring the backup file.

For backing up virtual machines running on ESXi, VMware Consolidated Backup is the recommended approach. Consolidated Backup enables off-host backup of virtual machines running any supported operating system from a centralized backup server using existing backup software already in the environment. It includes seamless integration with most major backup providers and frees the LAN from backup traffic.

You create a backup job for each virtual machine and dispatch that job to the backup server. Consolidated Backup takes a virtual machine snapshot and mounts the snapshot on the backup server directly from the SAN. As part of this process, Consolidated Backup quiesces the file system in the virtual machine to ensure that all that the entire state of the virtual machine is captured at the time the snapshot is created (this option is available only for Windows guest operating systems. The backup agent, already in place on the backup server, then backs up the entire contents of the virtual machine's virtual disk as a point-in-time snapshot (Linux or Windows guest operating systems) or, optionally, selected files within the guest operating system's file system (Windows only). Finally, Consolidated Backup unmounts the snapshot and takes the virtual disk out of snapshot mode.

The latest version of VMware Consolidated Backup enables more flexibility in terms of the storage and backup architectures in use. Consolidated Backup now supports the latest backup tools on the market and works with iSCSI, NAS, and locally attached storage in addition to Fibre Channel attached storage. The backup server can also function within a virtual machine.

VMware Consolidated Backup enables you to create full and incremental file backups of virtual machines for recovery of individual files and directories as well as full image backup of virtual machines for disaster recovery. It has built-in integrations with

most major backup providers, allowing you to leverage existing investment in backup agents to move virtual machine data from the backup server to tape or disk. Consolidated Backup is supported with backup software from Symantec, CommVault, VizionCore, and many more vendors.

Summary

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

Task	What Is Done with ESX 3	What to Do in ESXi
User management	• Create or manage users on service console, `sudo`, etc. • Create or manage users via the VI Client — either local ESX users or VirtualCenter users	Create or manage users via VI Client: — either local ESXi users or VirtualCenter users
Access local files: VMFS files, configuration files, log files	Console commands to browse datastores and virtual machine files; `vmkfstools` command	Remote command line interfaces commands to list and retrieve files, VI Client datastore browser for VMFS files (download and upload file)
Manipulate virtual machine files (for example, modify `.vmx`)	• Console commands to modify virtual machine files • Advanced configuration in the VI Client	Advanced configuration in VI Client; remote command line interfaces commands to list and retrieve virtual machine files
Backup	Virtual machine backup: agents in service console or Consolidated Backup ESX backup: use agents in the service console, create archive of service console files, or perform a scripted reinstall	Virtual machine backup: Consolidated Backup ESXi backup: single small backup file via RCLI
Hardware monitoring	Agents in service console, SNMP	CIM-based framework, SNMP
Troubleshooting or support	Local `esxcfg-*` commands	Remote command line interfaces `vicfg-*` commands
Advanced configuration	Edit configuration files (for example, `hostd.conf`) directly	Remote command line interfaces commands to list and retrieve ESXi configuration files
Logging	Remote syslog in service console	Built-in remote syslog client
Performance monitoring	VI Client, `esxtop`	VI Client, remote `resxtop`
Reporting and auditing	Service console scripts, log files	Remote command line interfaces commands to list and retrieve files: configuration and settings, etc.; export diagnostic data

About the Author

Charu Chaubal is Technical Marketing Manager at VMware, where he specializes in enterprise datacenter management with a focus on security. Previously, he worked at Sun Microsystems, where he had over 7 years experience with designing and developing distributed resource management and grid infrastructure software solutions. Charu received a Bachelor of Science in Engineering from the University of Pennsylvania, and a Ph.D. from the University of California at Santa Barbara, where he studied the numerical modeling of complex fluids. He is the author of numerous publications and several patents in the fields of datacenter automation and numerical price optimization.

vmware®