

Using vmrun to Control Virtual Machines

VMware Workstation

VMware Fusion

VMware Server

VMware vSphere 4

EN-000222-1

vmware[®]

You can find the most up-to-date technical documentation on the VMware Web site at:

<http://www.vmware.com/support/>

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Copyright © 2008-2009 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About This Book	5
Using vmrun to Control Virtual Machines	7
About the vmrun Utility	7
Power Commands	7
Snapshot Commands	7
Record and Replay Commands	7
Guest Operating System Commands	8
Maintenance Commands	8
VProbes Commands	8
Limitations	8
Setting Up vmrun	8
Prerequisites	8
Linux Setup	8
Windows Setup	9
Mac OS X Setup	9
Specifying the VMware Product Platform	9
Guest Operations	9
Hosted Platforms Run Locally	9
VMware vSphere Run Remotely	10
VMware Server Run Remotely	10
Virtual Machine Run Syntax	10
Path to VMX File	10
Disabling Dialog Boxes	10
Syntax of vmrun Commands	11
Examples of Using vmrun	15
Reboot Commands	15
Power Commands	15
Snapshot Commands	16
Record and Replay Commands	16
Running Guest Applications	16
Guest-Host File Operations	17
Maintenance Commands	18
Index	19

About This Book

This manual, *Using vmrun to Control Virtual Machines*, documents the `vmrun` utility, which helps you manage the collection of virtual machines on a VMware® host.

Revision History

This book is revised with each release of the product or when necessary. A revised version can contain minor or major changes. [Table 1](#) summarizes the significant changes in each version of this guide.

Table 1. Revision History

Revision	Description
20090909	For VMware vSphere 4, VMware Workstation Technical Preview, and VMware Fusion Technical Preview.
20081231	Included information about null interpreter for <code>RunScriptInGuest</code> on Windows.
20080815	Revision with more examples for VMware Server 2.0 RC2 and Workstation 6.5 RC.
20080724	Initial release including support for VMware Fusion on Intel-based Macintosh OS X hosts.
20080623	Draft of this manual for the VMware Server 2.0 RC1 and Workstation 6.5 Beta2 releases.

Intended Audience

This book is intended for developers and system administrators who want to control virtual machines on various product platforms, including VMware vSphere™ (ESX™/ESXi hosts and VMware vCenter™ Server), VMware Workstation, VMware Fusion®, and VMware Server.

Document Feedback

VMware welcomes your suggestions for improving our documentation. Send your feedback to docfeedback@vmware.com.

Technical Support and Education Resources

The following sections describe the technical support resources available to you. To access the current versions of other VMware books, go to <http://www.vmware.com/support/pubs>.

Online and Telephone Support

To use online support to submit technical support requests, view your product and contract information, and register your products, go to <http://www.vmware.com/support>.

Support Offerings

To find out how VMware support offerings can help meet your business needs, go to <http://www.vmware.com/support/services>.

VMware Professional Services

VMware Education Services courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. Courses are available onsite, in the classroom, and live online. For onsite pilot programs and implementation best practices, VMware Consulting Services provides offerings to help you assess, plan, build, and manage your virtual environment. To access information about education classes, certification programs, and consulting services, go to <http://www.vmware.com/services>.

Using vmrun to Control Virtual Machines

You can use the `vmrun` command-line utility to control specific virtual machines, or teams of virtual machines. The `vmrun` utility is available on VMware products that include the VIX API libraries, or when the libraries are separately installed. This chapter contains the following sections:

- [“About the vmrun Utility”](#) on page 7
- [“Setting Up vmrun”](#) on page 8
- [“Specifying the VMware Product Platform”](#) on page 9
- [“Virtual Machine Run Syntax”](#) on page 10
- [“Examples of Using vmrun”](#) on page 15

About the vmrun Utility

The `vmrun` utility runs on most VMware product platforms, including VMware Workstation, VMware Fusion, ESX/ESXi hosts, vCenter Server, and VMware Server. For information about the facilities that `vmrun` controls, see the user documentation for your product.

You can use `vmrun` to control virtual machines and automate guest operations.

Power Commands

Power commands control these virtual machine operations: start (power on), stop (power off), reset (reboot), suspend (but allow local work to resume), pause (without interrupting), and unpause (continue).

VMware Workstation can group virtual machines in teams, and apply power operations to the whole team.

Snapshot Commands

A snapshot captures the state of a virtual machine at the time of the snapshot, including all data on virtual disks. You can take a snapshot of a virtual machine in its current power state and later revert to the snapshot. Snapshots are useful for experimentation and especially useful for backups.

Snapshot commands list existing snapshots of a virtual machine, create a snapshot, delete a snapshot, and revert a virtual machine to its state at the time of a snapshot. VMware Server limits you to one snapshot.

Record and Replay Commands

You can record virtual machine events for later replay. The recording is called a replay snapshot. It is similar to a movie. At this time, only VMware Workstation supports record and replay.

These commands begin or end the recording of events, and begin or end the replay of a recording.

Guest Operating System Commands

You can use the `vmrun` utility to interact with a guest operating system in the following ways:

- Run an executable program in the guest operating system, or run an interpreted script. These interfaces serve a similar purpose, but `runProgramInGuest` provides more control.
- Check if a file exists in the guest, delete a file, rename a file, list files, and create or delete a directory.
- Copy a file from the host to the guest, or copy a file from the guest to the host.
- Add a shared folder from the host, make a shared folder writable in the guest, remove a shared folder, or capture a screen image from the guest (VMware Workstation and VMware Fusion only).
- List the processes running in the guest operating system, or end a process (with permission).
- Read or write a variable in the guest operating system's environment or virtual machine state.

The timeout (wait for VMware Tools) is five minutes for all guest-related commands.

Maintenance Commands

This category includes commands to list all running virtual machines, upgrade the virtual machine hardware version, and install VMware Tools in the guest operating system.

Additionally, except on VMware Server, you can clone a virtual machine image to another virtual machine.

On VMware vSphere (ESX/ESXi hosts and vCenter Server) or VMware Server, you can register and unregister virtual machines.

VProbes Commands

On Workstation, `vmrun` can interact with VProbes, a facility for instrumenting a powered-on guest operating system, its processes, and the virtualization layer. See the *VProbes Programming Reference* for details.

Limitations

VMware Server does not support teams, shared folders, cloning, record and replay, or multiple snapshots. When you try to create a second snapshot, the UI asks you to overwrite your existing snapshot.

VMware Fusion does not support snapshot trees.

Setting Up vmrun

The procedure for setting up `vmrun` varies by operating system type of the client computer.

Prerequisites

The `vmrun` command installs with VMware Workstation and VMware Fusion.

For use with remote product platforms, you can obtain `vmrun` by installing the VIX standalone libraries, available free of charge on the VMware download site.

Linux Setup

To use the `vmrun` utility on Linux

- 1 In a command or terminal window, type `vmrun` to see command-line options.
- 2 If this fails on an old Linux distribution: as root or superuser, edit the `/etc/ld.so.conf` file, add the following line giving the VIX library's default location, save the file, and run the `ldconfig` command.


```
/usr/lib/vmware-vix/lib
```
- 3 Continue with [“Specifying the VMware Product Platform”](#) on page 9.

Windows Setup

To use the vmrun utility on Windows

- 1 Find the vmrun utility, which gets installed in this folder by default:
C:\Program Files\VMware\VMware VIX
- 2 Add the folder location to your system path. If VMware Workstation is already in your system Path, this step is unnecessary because a copy of vmrun is also installed there. On Windows XP, click:
Computer > Properties > Advanced > Environment Variables > System variables > Path > Edit
With the right arrow key, move the input pointer to the end of line, add a semicolon, add the full path of the folder where vmrun is located, and click **OK** several times.
- 3 In a command window, type vmrun to see command-line options.
- 4 Continue with [“Specifying the VMware Product Platform”](#) on page 9.

Mac OS X Setup

To use the vmrun utility on a Macintosh

- 1 In a Terminal window, add the VMware Fusion directory to your system path:
export PATH="\$PATH:/Library/Application Support/VMware Fusion"
- 2 In the Terminal window, type vmrun to see command-line options.
- 3 Continue with [“Specifying the VMware Product Platform”](#) on page 9.

Specifying the VMware Product Platform

The vmrun utility accepts option flags, commands, and parameters in this form:

```
vmrun <flags> <command> <parameters>
```

In the following syntax examples, options enclosed in angle brackets indicate variables that you supply.

Guest Operations

Guest operations require authentication by the guest operating system, so their command descriptions in [Table 2, “vmrun Commands and Parameters,”](#) on page 11 say that a “valid guest login” is required. Use the following flags to specify the guest login:

```
-gu <userName in guest OS>
-gp <userPassword in guest OS>
```

Hosted Platforms Run Locally

On VMware Workstation and VMware Fusion, vmrun controls guest operating systems on the local host. You do not need to specify a remote host name or port.

For VMware Workstation and VMware Fusion, use the -T flag as follows:

```
vmrun -T ws
vmrun -T fusion
```

NOTE On VMware Workstation, and VMware Fusion, starting (powering on) a virtual machine with the default gui option requires a window system (user interface) to be running on the host. VMware vSphere and VMware Server do not impose this requirement.

VMware vSphere Run Remotely

In VMware vSphere, use the `-T` flag as follows to connect to an ESX/ESXi 4.0 host:

```
vmrun -T esx
```

In VMware vSphere, use the `-T` flag as follows to connect to the vCenter Server:

```
vmrun -T vc
```

VMware vSphere requires more flags for connections to an ESX/ESXi host or the vCenter Server:

```
-h https://<hostName or IPAddr>/sdk
-P <portNumber>
-u <adminLogin on VMware Server>
-p <adminPassword on VMware Server>
```

The port number defaults to 443. You can also specify the port number in the `-h` option after the host name or IP address, separated by a colon, in standard URL syntax. For example, this command lists all running virtual machines on a remote server:

```
vmrun -T esx -h https://esx.example.com:8333/sdk -u root -p secretpw list
```

VMware Server Run Remotely

To set the host type for remote access to VMware Server 2.0, use the `-T` option with `-h` and other options:

```
vmrun -T server -h https://vm2.example.com:443/sdk -u root -p secretpw list
```

To set the host type for remote access to VMware Server 1.0.x, use the `-T` option, the `-h` option with host name instead of URL, and the `-P` option with port number.

```
vmrun -T server1 -h vm1.example.com -P 443 -u root -p secretpw list
```

Virtual Machine Run Syntax

This section documents the syntax of available commands in the `vmrun` utility.

Path to VMX File

VMware stores virtual machines as a package that includes the virtual machine settings file (`<vname>.vmx`) and the virtual disks. When required, you must provide the complete path to the `.vmx` file. Here are examples of where the `.vmx` file might be located:

- Datastore on an ESX/ESXi host:
[Storage1] Win XP/Win XP.vmx
- VMware Server datastore:
[standard] Win XP/Win XP.vmx
- VMware Workstation for Windows path:
C:\Documents and Settings\\My Documents\My Virtual Machines\Win XP\Win XP.vmx
- VMware Workstation for Linux path:
/home/<username>/VirtualMachines/Ubuntu/Ubuntu.vmx
- Fusion for Mac OS X path:
~/Documents/Virtual Machines.localized/Windows XP Home.vmwarevm/Windows XP Home.vmx

Disabling Dialog Boxes

With virtual machines that require input through a user-interface dialog box, the `vmrun` utility might time out and fail. To disable dialog boxes, insert the following line in the virtual machine configuration file (`.vmx`):

```
msg.autoAnswer = TRUE
```

Syntax of vmrun Commands

Table 2 lists vmrun commands and parameters according to their function. Parameters are listed one per line. Parameters enclosed in square brackets are optional. The vertical bar indicates a keyword choice.

Table 2. vmrun Commands and Parameters

Command	Description	Parameters
Power Commands		
start (Teams supported only on VMware Workstation.)	Starts a virtual machine (.vmx file) or team (.vmtm file). The default <code>gui</code> option starts the machine interactively, which is required to display a VMware user interface. The <code>nogui</code> option suppresses the user interface, including startup dialog box, to allow noninteractive scripting.	<path to .vmx or .vmtm file> [<code>gui</code> <code>nogui</code>]
stop (Teams supported only on VMware Workstation.)	Stops a virtual machine (.vmx file) or team (.vmtm file). Use the <code>soft</code> parameter to power off the guest after running shutdown scripts. Use the <code>hard</code> parameter to power off the guest without running scripts, as if you pressed the power button. The default is to use the <code>powerType</code> specified in the .vmx file, if present.	<path to .vmx or .vmtm file> [<code>hard</code> <code>soft</code>]
reset (Teams supported only on VMware Workstation.)	Resets a virtual machine (.vmx file) or team (.vmtm file). Use the <code>soft</code> parameter to run shutdown scripts before rebooting the guest. Use the <code>hard</code> parameter to reboot the guest without running scripts, as if you pressed the reset button. The default is to use the <code>powerType</code> specified in the .vmx file, if present.	<path to .vmx or .vmtm file> [<code>hard</code> <code>soft</code>]
suspend (Teams supported only on VMware Workstation.)	Suspends a virtual machine (.vmx file) or team (.vmtm) without shutting down, so local work can resume later. The <code>soft</code> parameter suspends the guest after running system scripts. On Windows guests, these scripts release the IP address. On Linux guests, the scripts suspend networking. The <code>hard</code> parameter suspends the guest without running the scripts. The default is to use the <code>powerType</code> specified in the .vmx file, if present. To resume virtual machine operation after <code>suspend</code> , use the <code>start</code> command. On Windows, the IP address is retrieved. On Linux, networking is restarted.	<path to .vmx or .vmtm file> [<code>hard</code> <code>soft</code>]
pause (Pause supported only on VMware Workstation)	Pauses a virtual machine (.vmx file). You can use this either to pause replay, or to pause normal operation.	<path to .vmx file>
unpause (Unpause supported only on VMware Workstation)	Resumes operation of a virtual machine (.vmx file) from where you paused replay or normal operation.	<path to .vmx file>
Snapshot Commands		
listSnapshots	Lists all snapshots in a virtual machine (.vmx file). The <code>showtree</code> option displays snapshots in tree format, with children indented under their parent.	<path to .vmx file> [<code>showtree</code>]
snapshot (VMware Server does not support multiple snapshots. VMware Fusion does not support snapshot trees.)	Creates a snapshot of a virtual machine (.vmx file). For products such as Workstation that support multiple snapshots, you must provide the snapshot name. Because the forward slash defines pathnames, do not use the slash character in a snapshot name, because that makes it difficult to specify the snapshot path later.	<path to .vmx file> <snapshot name>

Table 2. vmrun Commands and Parameters (Continued)

Command	Description	Parameters
deleteSnapshot (VMware Server always deletes the root snapshot.)	Removes a snapshot from a virtual machine (.vmx file). For products such as Workstation that support multiple snapshots, you must provide the snapshot name. The virtual machine must be powered off or suspended. If this snapshot has children, they become children of the deleted snapshot's parent, and subsequent snapshots continue as before from the end of the chain. The <code>andDeleteChildren</code> option deletes not only the specified snapshot, but also its children recursively. See <code>revertToSnapshot</code> for solutions to name conflicts.	<path to .vmx file> <snapshot name> [<code>andDeleteChildren</code>]
revertToSnapshot (VMware Server always reverts to the root snapshot.)	Sets the virtual machine to its state at snapshot time. However, if the virtual machine was powered on at the time of the snapshot, <code>vmrun</code> reverts it to suspended state. If a snapshot has a unique name within a virtual machine, revert to that snapshot by specifying the path to the virtual machine's configuration file and the unique snapshot name. If several snapshots have the same name, specify the snapshot by including a full pathname for the snapshot. A pathname is a series of snapshot names, separated by forward slash characters (/). Each name specifies a new snapshot in the tree. For example, the pathname <code>Snap1/Snap2</code> identifies a snapshot named <code>Snap2</code> that was taken from the state of a snapshot named <code>Snap1</code> .	<path to .vmx file> <snapshot name> or <path to .vmx file> <snap1/snap2/snapN>
Record and Replay Commands		
beginRecording (Recording supported only on VMware Workstation.)	Begins recording a running virtual machine (.vmx file), storing activity in the specified snapshot object, with optional description. Only one recording or replay can be active at a time.	<path to .vmx file> <snapshot object name> [<description>]
endRecording	Ends the recording of a virtual machine (.vmx file) that is in progress, and close its snapshot object.	<path to .vmx file>
beginReplay (Replay supported only on VMware Workstation.)	Begins replaying the recorded activity of a powered off virtual machine (.vmx file) from a snapshot object, powering off if necessary. Only one recording or replay can be active at a time. You can pause replay with the <code>pause</code> command, and resume replay with the <code>unpause</code> command.	<path to .vmx file> <snapshot object name>
endReplay	Ends replay of the recorded virtual machine (.vmx file) that is underway.	<path to .vmx file>
Guest Operating System Commands		
runProgramInGuest	Runs a program in the guest operating system. The <code>-noWait</code> option returns a prompt immediately after the program starts in the guest, rather than waiting for it to finish. This option is useful for interactive programs. The <code>-activeWindow</code> option ensures that the Windows GUI is visible, not minimized. It has no effect on Linux. The <code>-interactive</code> option forces interactive guest login. It is useful for Windows Vista guests to make the program visible in the console window. Provide the full pathname of a program accessible to the guest. VMware Tools and valid guest login are required. Also provide full accessible pathnames for any files specified in the program arguments, which are optional according to requirements of the named program.	<path to .vmx file> [<code>-noWait</code> <code>-activeWindow</code> <code>-interactive</code>] <program name> [<program arguments>]
fileExistsInGuest	Checks whether the specified file exists in the guest operating system. VMware Tools and a valid guest login are required.	<path to .vmx file> <path to file on guest>

Table 2. vmrun Commands and Parameters (Continued)

Command	Description	Parameters
<code>setSharedFolderState</code> (VMware vSphere and VMware Server do not support shared folders.)	Modifies the writability state of a folder shared between the host and a guest virtual machine (.vmx file). The share name is a mount point in the guest file system. The path to folder is the exported directory on the host. A shared folder can be made writable or read-only.	<path to .vmx file> <share name> <path to folder on host> writable readonly
<code>addSharedFolder</code> (VMware vSphere and VMware Server do not support shared folders.)	Adds a folder to be shared between the host and guest. The share name is a mount point in the guest file system. The path to folder is the exported directory on the host. On Windows guests, there might be a delay before shared folders are visible to the <code>InGuest</code> commands.	<path to .vmx file> <share name> <path to folder on host>
<code>removeSharedFolder</code> (VMware vSphere and VMware Server do not support shared folders.)	Removes a guest virtual machine's access to a shared folder on the host. The share name is a mount point in the guest file system.	<path to .vmx file> <share name>
<code>enableSharedFolders</code> (VMware vSphere and VMware Server do not support shared folders.)	Allows the guest virtual machine, specified by .vmx file, to share folders with its host. After enabling, run <code>addSharedFolder</code> to specify each host folder to share. The optional <code>runtime</code> argument means to share folders only until the virtual machine is powered off. Otherwise the setting persists at next power on.	<path to .vmx file> [runtime]
<code>disableSharedFolders</code> (VMware vSphere and VMware Server do not support shared folders.)	Stops the guest virtual machine, specified by .vmx file, from sharing folders with its host. The optional <code>runtime</code> argument means to stop sharing folders until the virtual machine is powered off. At the next power on, the previous setting persists.	<path to .vmx file> [runtime]
<code>listProcessesInGuest</code>	Lists all processes running in the guest operating system. VMware Tools and a valid guest login are required.	<path to .vmx file>
<code>killProcessInGuest</code>	Stops a specified process in the guest operating system. VMware Tools and a valid guest login are required. Take process ID from the number listed after <code>pid=</code> in the output of <code>listProcessesInGuest</code> .	<path to .vmx file> <process ID>
<code>runScriptInGuest</code>	Runs a command script in the guest operating system. VMware Tools and a valid guest login are required. The interpreter path is the command that runs the script. Provide the complete text of the script, not a filename.	<path to .vmx file> <interpreter path> <script text>
<code>deleteFileInGuest</code>	Deletes a specified file from the guest operating system. VMware Tools and a valid guest login are required. For Windows Vista restrictions on this command, see note in "Guest-Host File Operations" on page 17.	<path to .vmx file> <path to file on guest>
<code>createDirectoryInGuest</code>	Creates the specified directory in the guest operating system. VMware Tools and a valid guest login are required. For Windows Vista restrictions on this command, see note in "Guest-Host File Operations" on page 17.	<path to .vmx file> <directory path on guest>
<code>deleteDirectoryInGuest</code>	Deletes a directory from the guest operating system. VMware Tools and a valid guest login are required. For Windows Vista restrictions on this command, see note in "Guest-Host File Operations" on page 17.	<path to .vmx file> <directory path on guest>
<code>listDirectoryInGuest</code>	Lists directory contents in the guest operating system. VMware Tools and a valid guest login are required.	<path to .vmx file> <directory path on guest>

Table 2. vmrun Commands and Parameters (Continued)

Command	Description	Parameters
copyFileFromHostToGuest	Copies a file from the host to the guest operating system. VMware Tools and a valid guest login are required. Specify the source file (host) before the destination file (guest). For Windows Vista restrictions on this command, see note in “Guest-Host File Operations” on page 17.	<path to .vmx file> <file path on host> <file path in guest>
copyFileFromGuestToHost	Copies a file from the guest operating system to the host. VMware Tools and a valid guest login are required. Specify the source file (guest) before the destination file (host).	<path to .vmx file> <file path in guest> <file path on host>
renameFileInGuest	Renames or moves a file in the guest operating system. VMware Tools and a valid guest login are required. Specify the source name (original) before the destination (new).	<path to .vmx file> <original filename> <new filename>
captureScreen	Captures the screen of the virtual machine to a local file. The specified output file on the host is in PNG format. A valid guest login is required.	<path to .vmx file> <output path on host>
writeVariable	Writes a variable to the virtual machine state or guest. You can set either runtime configuration in the .vmx file, or environment variables in the guest operating system. The latter requires VMware Tools and a valid guest login (for Linux guests, setting guestEnv requires root login). Provide the variable name and its value.	<path to .vmx file> [runtimeConfig guestEnv] <variable name> <variable value>
readVariable	Reads a variable from the virtual machine state or guest. You can get either runtime configuration in the .vmx file, or environment variables in the guest operating system. The latter requires a valid guest login.	<path to .vmx file> [runtimeConfig guestEnv] <variable name>
Maintenance Commands		
list	Lists all running virtual machines.	None
upgradevm	Upgrades a virtual machine to the current version of virtual hardware. Has no effect if the virtual hardware version is the most recent supported.	<path to .vmx file>
installTools	Prepares to install VMware Tools in the guest operating system. In Windows guests with autorun enabled, the VMware Tools installer starts by itself. In Linux guests without autorun, this command connects the virtual CD-ROM drive to the VMware Tools ISO image suitable for the guest, but the installer does not start. You must complete the installation with additional manual steps, as described in your product documentation.	<path to .vmx file>
register (Registration not supported on VMware Workstation or VMware Fusion.)	Registers a virtual machine (.vmx file), adding it to the host’s inventory. Path format depends on the product. For VMware Server 2.0, “[storage1] vm/vm.vmx” (starting with the datastore) is typical.	<path to .vmx file>
unregister (Registration not supported on VMware Workstation or VMware Fusion.)	Unregisters a virtual machine (.vmx file), removing it from the host’s inventory. Path format depends on the product. For Server 2.0, “[storage1] vm/vm.vmx” (starting with the datastore) is typical.	<path to .vmx file>
listRegisteredVM	Lists all registered virtual machines.	None
deleteVM	Removes the virtual machine at source .vmx file path.	<path to .vmx file>

Table 2. vmrun Commands and Parameters (Continued)

Command	Description	Parameters
clone (Cloning not supported on VMware Server or on VMware Fusion.)	Creates a copy of the virtual machine and guest. Provide the source .vmx file pathname, and the destination .vmx file pathname. You can create either a normal full clone or a linked clone. To make the clone from a snapshot, rather than from the current virtual machine state, specify a snapshot name.	<path to .vmx file> <destination .vmx file path> full linked [<snapshot name>]
VProbes Commands		
(VProbes permitted only on VMware Workstation.)		
vprobeVersion	Shows the VProbes version on the virtual machine.	<path to .vmx file>
vprobeLoad	Loads a VP script onto the virtual machine.	<path to .vmx file> <text of the VP script>
vprobeLoadFile	Loads a VP script file onto the virtual machine.	<path to .vmx file> <path to VP script>
vprobeReset	Disables all VProbes on the virtual machine.	<path to .vmx file>
vprobeListProbes	Lists active VProbes on the virtual machine.	<path to .vmx file>
vprobeListGlobals	Lists VProbes global variables on the virtual machine.	<path to .vmx file>

Examples of Using vmrun

The following command-line examples work on VMware Workstation (-T ws), VMware Fusion (-T fusion), or VMware ESX/ESXi hosts (-T esx).

You can derive the guest operating system type in examples by distinguishing / for Linux and \ for Windows.

Reboot Commands

Reboot a virtual machine running on VMware Workstation for Linux:

```
vmrun -T ws reset /path/to/vm/RHEL4/RHEL4.vmx soft
```

Reboot a virtual machine running on VMware Workstation for Windows:

```
cd "C:\Documents and Settings\\My Documents\My Virtual Machines"
vmrun -T ws reset "WindowsXP\WindowsXP.vmx" soft
```

Reboot a virtual machine running on VMware Fusion:

```
vmrun -T fusion reset ~/Documents/VirtualMachines.localized/WindowsXP.vmwarevm/WindowsXP.vmx soft
```

Reboot a virtual machine running on an ESX/ESXi host:

```
vmrun -T esx -h https://10.0.1.8/sdk -u root -p <pass> reset "[storage1] WinXP/WinXP.vmx" soft
```

Power Commands

Power on a virtual machine with VMware Workstation on a Windows host:

```
vmrun start "C:\Documents and Settings\\My Documents\My Virtual Machines\WinXP\WinXP.vmx"
```

This error message indicates that the VIX package you installed does not support VMware Server:

```
vmrun -T server start "My Virtual Machines\WinXP\WinXP.vmx"
Error: The specified service provider was not found
```

Power off a virtual machine with VMware Workstation on a Windows host:

```
vmrun stop "C:\Documents and Settings\\My Documents\My Virtual Machines\WinXP\WinXP.vmx"
```

On the remote ESX/ESXi host with IP address 10.0.1.8, power on a virtual machine:

```
vmrun -T esx -h https://10.0.1.8/sdk -u root -p <pass> start "[storage1] WinXP/WinXP.vmx"
```

If HTTPS service is not configured on port 443, specify the appropriate port after a colon separator:

```
vmrun -T esx -h https://10.0.1.9:8333/sdk -u root -p <pass> start "[storage1] WinXP/WinXP.vmx"
```

On the remote ESX/ESXi host with IP address 10.0.1.8, power off the virtual machine:

```
vmrun -T esx -h https://10.0.1.8/sdk -u root -p <pass> start "[storage1] WinXP/WinXP.vmx"
```

For VMware Server installed on Windows hosts, the `-u` user is usually Administrator, not root.

```
vmrun -T server -h https://10.0.1.8/sdk -u Administrator -p <pass> start "[std] WinXP/WinXP.vmx"
```

Snapshot Commands

Create a snapshot of a virtual machine with VMware Workstation on a Linux host or VMware Fusion:

```
vmrun -T ws snapshot /path/to/vm/Ubuntu/Ubuntu.vmx mySnapshot
```

List snapshots on the virtual machine, showing the one made in the previous command:

```
vmrun -T ws listSnapshots /path/to/vm/Ubuntu/Ubuntu.vmx
Total snapshots: 1
mySnapshot
```

Revert to the snapshot you made, which suspends the virtual machine, and restart to resume operation:

```
vmrun -T ws revertToSnapshot /path/to/vm/Ubuntu/Ubuntu.vmx mySnapshot
vmrun -T ws start /path/to/vm/Ubuntu/Ubuntu.vmx
```

Delete the snapshot by specifying its name:

```
vmrun -T ws deleteSnapshot /path/to/vm/Ubuntu/Ubuntu.vmx mySnapshot
```

Record and Replay Commands

Start recording user events on a Windows guest, beginning with a snapshot of the virtual machine state:

```
vmrun -T ws -gu <user> -gp <pass> beginRecording WinXP\WinXP.vmx session1
```

When you stop recording, the user interface might ask you to name your session:

```
vmrun -T ws -gu <user> -gp <pass> endRecording WinXP\WinXP.vmx
```

Revert to the virtual machine snapshot and start replaying your recording of user events:

```
vmrun -T ws -gu <user> -gp <pass> beginReplay WinXP\WinXP.vmx session1
```

Pause replay, which you can also do from the user interface:

```
vmrun -T ws -gu <user> -gp <pass> pause WinXP\WinXP.vmx
```

Resume replay, which you can also do from the user interface:

```
vmrun -T ws -gu <user> -gp <pass> unpause WinXP\WinXP.vmx
```

End replay, which you can also do from the user interface:

```
vmrun -T ws -gu <user> -gp <pass> endReplay WinXP\WinXP.vmx
```

Running Guest Applications

Start the command tool on a Windows guest, minimized:

```
vmrun -T ws -gu <user> -gp <pass> runProgramInGuest WinXP\WinXP.vmx cmd.exe
```

Start the command tool on a Windows guest, as an active window on the desktop:

```
vmrun -T ws -gu <user> -gp <pass> runProgramInGuest WinXP\WinXP.vmx -activeWindow cmd.exe
```

Run a batch script on a Windows guest, with Perl as the script interpreter:

```
vmrun -T ws -gu <user> -gp <pass> runScriptInGuest Win2k\Win2k.vmx C:\perl\perl.exe C:\script.pl
```

Run a batch script and keep running afterwards. To use `cmd.exe` on Windows, specify interpreter as null:

```
vmrun -T ws -gu <user> -gp <pass> runScriptInGuest WindowsXP\WindowsProfessionalXP.vmx ""
      "cmd.exe /k \"C:\\Program Files\\Microsoft Visual Studio\\VC\\vcvarsall.bat\" x86"
```

Run a Bash shell script called `runit` on a Linux guest:

```
vmrun -T ws -gu <user> -gp <pass> runScriptInGuest Ubuntu/Ubuntu.vmx /bin/bash /home/<user>/runit
```

Start an X clock on a Linux guest (requires `-display` option to appear on the console). Do the same thing, but return control back to the console immediately.

```
vmrun -gu <user> -gp <pass> runProgramInGuest SUSE/SUSE.vmx /usr/bin/xclock -display :0
vmrun -gu <user> -gp <pass> runProgramInGuest SUSE/SUSE.vmx -noWait /usr/bin/xclock -display :0
```

For Linux applications that do not accept the `-display` command-line option, first set the guest environment for the `vmware-guestd` process, which requires root permission:

```
vmrun -T ws -gu root -gp <rootpass> writeVariable Ubuntu/Ubuntu.vmx guestEnv DISPLAY :0
vmrun -T ws -gu <user> -gp <userpass> runProgramInGuest Ubuntu/Ubuntu.vmx /usr/bin/firefox
```

List processes in a Linux guest, and end the process numbered 8192:

```
vmrun -T ws -gu <user> -gp <pass> listProcessesInGuest Ubuntu/Ubuntu.vmx
vmrun -T ws -gu <user> -gp <pass> killProcessInGuest Ubuntu/Ubuntu.vmx 8192
```

Run a Perl script on a Linux guest to remove DOS-style carriage returns from a file:

```
vmrun -T ws -gu <user> -gp <pass> runProgramInGuest Ubuntu/Ubuntu.vmx /usr/bin/perl -e
      "open(FILE, '>/tmp/unix.txt'); while (<>) { s/\r\n\n/ ; print FILE}" /tmp/dos.txt
```

Run a Perl script on a Windows guest to insert DOS-style carriage returns in a file:

```
vmrun -T ws -gu <user> -gp <pass> runProgramInGuest WinXP\WinXP.vmx C:\cygwin\bin\perl.exe -e
      "open(FILE, '>C:\dos.txt'); while (<>) { s/\n\r\n/ ; print FILE}" C:\unix.txt
```

Run a program in a Linux virtual machine on an ESX/ESXi host:

```
vmrun -T esx -h https://10.0.1.8/sdk -u root -p <pass> -gu <user> -gp <userpass>
      runProgramInGuest "[storage1] RHEL4/RHEL4.vmx" /usr/X11R6/bin/xclock -display :0
```

Guest-Host File Operations

To copy an arbitrary file from the host to a guest, the user must have write permission on the destination:

```
vmrun -gu <user> -gp <pass> copyFileFromHostToGuest Ubuntu\Ubuntu.vmx C:\Temp\img.db /tmp/img.db
```

To copy an arbitrary file from a guest to the host, the user must have read permission on the source file:

```
vmrun -gu <user> -gp <pass> copyFileFromGuestToHost Ubuntu\Ubuntu.vmx /home/<user>/addr addr.txt
```

Before sharing folders, you must enable them with the `enabledSharedFolders` command, or by selecting **VM > Settings > Options > Shared Folders > Enabled** in the user interface. On Linux guests the `/mnt/hgfs` directory is available for sharing, but you can use a different directory for shared folders.

To share a folder on a Windows host with a particular Linux guest:

```
vmrun -T ws addSharedFolder Ubuntu\Ubuntu.vmx <sharedFolderName> C:\Share
```

Shared folders are writable by default. To make a shared folder read-only, or to delete the shared folder:

```
vmrun -T ws setSharedFolderState Ubuntu\Ubuntu.vmx <sharedFolderName> C:\Share readonly
vmrun -T ws removeSharedFolder Ubuntu\Ubuntu.vmx <sharedFolderName>
```

NOTE Windows Vista requires Administrator for certain operations. The `copyFileFromHostToGuest` and `deleteFileInGuest` commands do not allow regular users, even those with administrator privileges, to modify `C:\` and system directories. The `createDirectoryInGuest` and `deleteDirectoryInGuest` commands do not allow regular users, even those with administrator privileges, to modify system directories. Only the Administrator account can perform these operations.

Maintenance Commands

List running virtual machines on VMware Workstation:

```
vmrun -T ws list  
Total running VMs: 2  
C:\Documents and Settings\user\My Documents\My Virtual Machines\Ubuntu\Ubuntu.vmx  
C:\Documents and Settings\user\My Documents\My Virtual Machines\WinXP\WinXP.vmx
```

Prepare to install VMware Tools on VMware Fusion:

```
vmrun -T fusion installTools RedHatEnt5/RedHatEnt5.vmx
```

Register a new virtual machine installed on an ESX/ESXi host:

```
vmrun -T esx -h https://10.0.1.5/sdk -u root -p <pass> register "[storage1] RHEL5/RHEL5.vmx"
```

Unregister an old virtual machine going out of service on an ESX/ESXi host:

```
vmrun -T esx -h https://10.0.1.5/sdk -u root -p <pass> unregister "[storage1] RHEL3/RHEL3.vmx"
```

On VMware vCenter, the `vmrun` command supports the same form of path-to-VMX specification as ESX/ESXi hosts. You cannot reach a virtual machine file through its resource pool or vApp.

```
vmrun -T vc -h https://10.0.1.9/sdk -u Administrator -p <pw> register "[storage1] RHEL5/RHEL5.vmx"
```

Index

A

addSharedFolder command 13, 17

B

beginRecording command 12, 16

beginReplay command 12, 16

C

captureScreen command 14

clone command 15

command-line flags 9

copyFileFromGuestToHost command 14, 17

copyFileFromHostToGuest command 14, 17

createDirectoryInGuest command 13

D

deleteDirectoryInGuest command 13

deleteFileInGuest command 13

deleteSnapshot command 12, 16

deleteVM command 14

dialog boxes, disabling 10

disableSharedFolders command 13

E

enableSharedFolders 13

endRecording command 12, 16

endReplay command 12, 16

ESX/ESXi hosts, flags for 10

F

fileExistsInGuest command 12

flags on command line 9

Fusion, flags for 9

G

guest operations 8

- addSharedFolder 13

- captureScreen 14

- copyFileFromGuestToHost 14

- copyFileFromHostToGuest 14

- createDirectoryInGuest 13

- deleteDirectoryInGuest 13

- deleteFileInGuest 13

- disableSharedFolders 13

- enableSharedFolders 13

- fileExistsInGuest 12

- killProcessInGuest 13

- listDirectoryInGuest 13

- listProcessesInGuest 13

- readVariable 14

- removeSharedFolder 13

- renameFileInGuest 14

- runProgramInGuest 12

- runScriptInGuest 13

- setSharedFolderState 13

- user name and password 9

- writeVariable 14

I

installTools command 14, 18

K

killProcessInGuest command 13, 17

L

Linux setup 8

list command 14, 18

listDirectoryInGuest command 13

listProcessesInGuest command 13, 17

listRegisteredVM command 14

listSnapshots command 11, 16

M

Mac OS setup 9

maintenance commands 8

- clone 15

- deleteVM 14

- installTools 14

- list 14

- listRegisteredVM 14

- register 14

- unregister 14

- upgradevm 14

P

path to VMX file 10

pause command 11, 16

power commands 7

- pause 11

- reset 11

- start 11

- stop 11

- suspend 11

- unpause 11

R

- record and replay commands **7**
 - beginRecording **12**
 - beginReplay **12**
 - endRecording **12**
 - endReplay **12**
- register command **14, 18**
- removeSharedFolder command **13, 17**
- renameFileInGuest command **14**
- reset command **11, 15**
- revertToSnapshot command **12, 16**
- runProgramInGuest command **12, 16**
- runScriptInGuest command **13, 16**

S

- Server, flags for **10**
- setSharedFolderState command **13, 17**
- setup on Linux **8**
- setup on Mac OS **9**
- setup on Windows **9**
- snapshot command **11, 16**
- snapshot commands **7**
 - deleteSnapshot **12**
 - listSnapshots **11**
 - revertToSnapshot **12**
 - snapshot **11**
- start command **11, 15**
- stop command **11, 15**
- suspend command **11**

T

- technical support resources **5**

U

- unpause command **11, 16**
- unregister command **14, 18**
- upgradevm command **14**

V

- vCenter Server, flags for **10**
- vmrun utility **7**
- VMware Fusion, flags for **9**
- VMware Server, flags for **10**
- VMware vSphere, flags for **10**
- VMware Workstation, flags for **9**
- VMX file, specifying **10**
- VProbes commands **8**
 - vprobeListGlobals **15**
 - vprobeListProbes **15**
 - vprobeLoad **15**
 - vprobeLoadFile **15**
 - vprobeReset **15**
 - vprobeVersion **15**

W

- Windows setup **9**
- Workstation, flags for **9**
- writeVariable command **14, 17**