Using vmrun to Control Virtual Machines

VMware Fusion 5

VMware Workstation 9 and Player 5

VMware vSphere 5.x

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN-001006-00

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

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

Contents

About This Book

This manual, *Using vmrun to Control Virtual Machines*, documents the `vmrun` utility, which helps you manage a 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
2012-10-04	For the VIX API 1.12 release. Added createTempFileInGuest command. Changed Fusion install location.
2011-04-21	For the VIX API 1.11 release, which supported vSphere 5.0, Workstation 8.0, and Fusion 4.0.
2010-07-19	For the VIX API 1.10 release, which supported vSphere 4.1, Workstation 7.1, and Fusion 3.1.
2009-10-20	For the VIX API 1.8 release, which supported VMware Workstation 7.0, Player 3.0, and VMware Fusion 3.0.
2009-09-09	For the VIX API 1.7 release, which provided support for ESX/ESXi hosts and VMware vSphere 4.
2008-12-31	New information about null interpreter for `RunScriptInGuest` on Windows.
2008-08-15	More examples for VMware Server 2.0 RC2 and Workstation 6.5 RC.
2008-07-24	Initial release, including support for VMware Fusion on Intel-based Macintosh OS X hosts.
2008-06-23	Initial draft 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 guest virtual machines on various VMware product platforms. Supported platforms include VMware Workstation, VMware Player, VMware Fusion®, and VMware vSphere™ (ESX™/ESXi hosts with vCenter™ Server).

The `vmrun` utility is associated with the VIX API. On some VMware platform products, you must install the VIX libraries before you can use `vmrun`.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation go to http://www.vmware.com/support/pubs.

Document Feedback

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

Using vmrun to Control Virtual Machines

This document contains the following sections:

About the vmrun Utility

You can use the `vmrun` command-line utility to control virtual machines and automate guest operations on VMware virtual machines. The `vmrun` utility is included with the VIX API libraries.

The `vmrun` utility runs on most VMware product platforms, including Workstation, Player, VMware Fusion, and VMware vSphere (ESX/ESXi hosts managed by vCenter Server). On these platforms, the VIX API libraries and the `vmrun` utility are often the best way to automate guest operations.

Capabilities of `vmrun` are summarized in the sections below.

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).

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 then use the snapshot to revert the virtual machine to its previous state. Snapshots are useful for data backup, and as a placeholder for development and testing.

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. Revert to snapshot does not resume running a virtual machine, even if it was running at the time of a snapshot.

VMware Fusion supports snapshots in a line only. VMware Server (discontinued) limits each virtual machine to one snapshot.

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 that you provide.

- Check if a file or directory exists in the guest, delete a file or directory, rename a file, list files, or create a new directory.
- Copy a file from the host to the guest, or from the guest to the host.
- Create a temporary file in the guest operating system.
- Add a shared folder from the host, make a shared folder writable in the guest, or remove a shared folder.
- Capture a screen image from the guest (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 (discontinued), you can clone a virtual machine image to another virtual machine.

On VMware vSphere and VMware Server (discontinued), you can register and unregister virtual machines.

VProbes Commands

On Workstation and Fusion, `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 Fusion does not support snapshot trees.

VMware Player does not support pause and unpause, snapshot operations, virtual machine cloning, or virtual hardware upgrade.

VMware Server (discontinued) did not support teams, shared folders, virtual machine cloning, or multiple snapshots. When you tried to create a second snapshot, the UI asked to overwrite your existing snapshot.

The record and replay commands were deprecated for Workstation 7.1 and removed in Workstation 8.

Setting Up vmrun

The procedure for setting up `vmrun` varies based on the operating system of the client computer.

How to Get vmrun

The `vmrun` utility installs with 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 with the default location of the VIX library, save the file, and run the `ldconfig` command.

   ```
   /usr/lib/vmware-vix/lib
   ```

Windows Setup

To use the `vmrun` utility on Windows

1. If you are running VMware Workstation, a copy of `vmrun` is included in the folder where you installed Workstation. If you install the VIX API, a possibly newer `vmrun` is placed in one of these folders:

   32-bit systems – `C:\Program Files\VMware\VMware VIX`
   64-bit systems – `C:\Program Files (x86)\VMware\VMware VIX`

2. (Optional) Most users find it convenient to add the `vmrun` install location, either `VMware Workstation` or `VMware VIX`, to the system path. On Windows 7:

   (right-click) **Computer >** (click) **Properties > Advanced System Settings > Environment Variables >** (in list of) **System Variables >** (select) **Path >** (click) **Edit**

   With the right arrow key, move the insertion point to the end of line, add a semicolon, add the full path of the folder where `vmrun` is located, and click **OK** three times.

3. In a command window, type `vmrun` to see command-line options.

Mac OS X Setup

To use the `vmrun` utility on Mac OS

1. In a Terminal window, add the VMware Fusion (4.0 and later) directory to your system path.

```
export PATH="$PATH:/Applications/VMware\ Fusion.app/Contents/Library"
```

2. Type `vmrun` to see command-line options.

Specifying the VMware Product Platform

The `vmrun` utility accepts option flags, commands, and parameters in the following syntax:

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

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

Encrypted Virtual Machines

Encrypted virtual machines require a password for most operations.

```
-vp <password for encrypted virtual machine>
```

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>
```

As of the VIX 1.10 release, you no longer need to specify a guest user name and password, or an encryption password. If `vmrun` needs authentication information, it prompts for it.

Running Hosted Platforms Locally

On 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. Use the –T flag for Workstation, Player, and VMware Fusion:

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

NOTE On VMware Fusion, Workstation, and Player, powering on a virtual machine with the default `gui` option requires a window system (user interface) to be running on the host. ESX/ESXi hosts, vCenter Server, and VMware Server (discontinued) do not impose this requirement.

Running VMware vSphere Remotely

In VMware vSphere, use the `–T` flag to connect to an ESX/ESXi host or to the vCenter Server.

```
vmrun –T esx
vmrun –T vc
```

VMware vSphere requires additional flags for connections to an ESX/ESXi host or to vCenter Server (VC):

```
–h <hostName or IPaddr>
–P <portNumber>
–u <adminLogin on ESX/ESXi or VC>
–p <adminPassword on ESX/ESXi or VC>
```

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, using standard URL syntax. For example, the following command lists all running virtual machines on a remote ESX/ESXi host:

```
vmrun –T esx –h esx.example.com:8333 –u root –p secretpw list
```

As of the VIX 1.10 release, if you do not specify administrator login name and password, `vmrun` prompts you for authentication information. Also as of the VIX 1.10 release, you do not need to specify leading `https://` and trailing `/sdk` with the `–h` option.

Running VMware Server 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 vm2.example.com:443 –u root –p secretpw list
```

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

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

Virtual Machine Run Reference

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

Path to VMX File

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

- Datastore on an ESX/ESXi host

  ```
  [datastore1] Win XP/Win XP.vmx
  ```

- Workstation for Windows path

  ```
  C:\Documents and Settings\<username>\My Documents\My Virtual Machines\Win XP\Win XP.vmx
  ```

- Workstation for Linux path

  ```
  /home/<username>/VirtualMachines/Ubuntu/Ubuntu.vmx
  ```

- VMware Fusion for Mac OS X path

  ```
  ~/Documents/Virtual Machines.localized/Windows XP Home.vmwarevm/Windows XP Home.vmx
  ```

IMPORTANT For the `vmrun` commands that require VMware Tools, you must install the latest VMware Tools package and, especially after operating system updates, keep VMware Tools up-to-date.

Disabling Dialog Boxes

With virtual machines that require user input through a 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 Workstation.)	Starts a virtual machine (`.vmx` file) or team (`.vmtm` file). The default `gui` option starts the machine interactively, which is required to display a VMware user interface. The `nogui` option suppresses the user interface, including the startup dialog box, to allow noninteractive scripting.	<path to `.vmx` or `.vmtm` file> `[ gui | nogui ]`
`stop` (Teams supported only on Workstation.)	Stops a virtual machine (`.vmx` file) or team (`.vmtm` file). Use the `soft` option to power off the guest after running shutdown scripts. Use the `hard` option to power off the guest without running scripts, as if you pressed the power button. The default is to use the `powerType` value specified in the `.vmx` file, if present.	<path to `.vmx` or `.vmtm` file> `[ hard | soft ]`
`reset` (Teams supported only on Workstation.)	Resets a virtual machine (`.vmx` file) or team (`.vmtm` file). Use the `soft` option to run shutdown scripts before rebooting the guest. Use the `hard` option to reboot the guest without running scripts, as if you pressed the reset button. The default is to use the `powerType` value specified in the `.vmx` file, if present.	<path to `.vmx` or `.vmtm` file> `[ hard | soft ]`
`suspend` (Teams supported only on Workstation.)	Suspends a virtual machine (`.vmx` file) or team (`.vmtm`) without shutting down, so local work can resume later. The `soft` option suspends the guest after running system scripts. On Windows guests, these scripts release the IP address. On Linux guests, the scripts suspend networking. The `hard` option suspends the guest without running the scripts. The default is to use the `powerType` value specified in the `.vmx` file, if present. To resume virtual machine operation after `suspend`, use the `start` command. On Windows, the IP address is retrieved. On Linux, networking is restarted.	<path to `.vmx` or `.vmtm` file> `[ hard | soft ]`
`pause` (Pause supported only on Workstation.)	Pauses a virtual machine (`.vmx` file). You can use this to suspend virtual machine operation.	<path to `.vmx` file>
`unpause` (Unpause supported only on Workstation.)	Resumes operation of a virtual machine (`.vmx` file) from where you paused normal operation.	<path to `.vmx` file>
Snapshot Commands		
`listSnapshots`	Lists all snapshots in a virtual machine (`.vmx` file). The `showtree` option displays snapshots in tree format, with children indented under their parent.	<path to `.vmx` file> `[ showtree ]`
`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 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 that support multiple snapshots, you must provide the snapshot name. The virtual machine must be powered off or suspended. If the 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 `andDeleteChildren` option deletes the specified snapshot and its children recursively. See `revertToSnapshot` for solutions to name conflicts.	<path to `.vmx` file> <snapshot name> `[ andDeleteChildren ]`
`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, `vmrun` reverts it to suspended state, but does not resume running the virtual machine. 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 `Snap1/Snap2` identifies a snapshot named `Snap2` that was taken from the state of a snapshot named `Snap1`.	<path to `.vmx` file> <snapshot name> or <path to `.vmx` file> <snap1/snap2/snapN>
Guest Operating System Commands		
	The timeout (wait for VMware Tools) is five minutes for all guest-related commands.	
`runProgramInGuest`	Runs a specified program in the guest operating system. The `-noWait` 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 `-activeWindow` option ensures that the Windows GUI is visible, not minimized. It has no effect on Linux. The `-interactive` option forces interactive guest login. It is useful for Vista and Windows 7 guests to make the program visible in the console window. You must provide the full pathname of a program accessible to the guest. Also provide full accessible pathnames for any files specified in the program arguments, according to requirements of the program. VMware Tools and a valid guest login are required.	<path to `.vmx` file> `[ -noWait | -activeWindow | -interactive ]` <program name> [<program arguments>]
`runScriptInGuest`	Runs the specified command script in the guest operating system. See `runProgramInGuest` for an explanation of options. The interpreter path is the command that runs the script. Provide the complete text of the script, not a filename. VMware Tools and a valid guest login are required.	<path to `.vmx` file> `[ -noWait | -activeWindow | -interactive ]` <interpreter path> <script text>
`listProcessesInGuest`	Lists all processes running in the guest operating system. VMware Tools and a valid guest login are required.	<path to `.vmx` file>
`killProcessInGuest`	Stops a specified process in the guest operating system. VMware Tools and a valid guest login are required. The process ID can be any number listed after `pid=` in the output of `listProcessesInGuest`.	<path to `.vmx` file> <process ID>
`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
`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>
`deleteFileInGuest`	Deletes the give file from the guest operating system. VMware Tools and a valid guest login are required. For Vista and Windows 7 restrictions on this command, see note in "Guest to Host File Operations" on page 17.	<path to `.vmx` file> <path to file on guest>
`directoryExistsInGuest`	Checks whether the specified directory exists in the guest operating system. VMware Tools and a valid guest login are required.	<path to `.vmx` file> <directory path on guest>
`createDirectoryInGuest`	Creates the specified directory in the guest operating system. VMware Tools and a valid guest login are required. For Vista and Windows 7 restrictions on this command, see note in "Guest to Host File Operations" on page 17.	<path to `.vmx` file> <directory path on guest>
`listDirectoryInGuest`	Lists contents of the specified directory in the guest operating system. VMware Tools and a valid guest login are required.	<path to `.vmx` file> <directory path on guest>
`deleteDirectoryInGuest`	Deletes the specified directory from the guest operating system. VMware Tools and a valid guest login are required. For Vista and Windows 7 restrictions on this command, see note in "Guest to Host File Operations" on page 17.	<path to `.vmx` file> <directory path on guest>
`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 Vista and Windows 7 restrictions on this command, see note in "Guest to 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>
`createTempfileInGuest`	Creates a temporary file in the guest operating system, and returns the pathname of the temporary file created. Pathname varies according to the operating system. You can run `deleteFileInGuest` to remove the file. VMware Tools and a valid guest login are required.	<path to `.vmx` file>
`enableSharedFolders` (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 `addSharedFolder` to specify each host folder to share. The optional `runtime` 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]`
`disableSharedFolders` (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 `runtime` argument means to stop sharing folders only until the virtual machine is powered off. Otherwise, the setting persists at next power on.	<path to `.vmx` file> `[runtime]`
`addSharedFolder` (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 `InGuest` commands.	<path to `.vmx` file> <share name> <path to folder on host>

Table 2. `vmrun` Commands and Parameters (Continued)

Command	Description	Parameters
`removeSharedFolder` (VMware vSphere and VMware Server do not support shared folders.)	Removes the 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>
`setSharedFolderState` (VMware vSphere and VMware Server do not support shared folders.)	Modifies the writability state of a specified 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`
`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 a non-persistent guest variable (`guestVar`), a runtime configuration parameter as stored in the `.vmx` file, or an environment variable (`guestEnv`) in the guest operating system. A guest variable is a runtime-only value that provides a simple way to pass runtime values in and out of the guest. Environment variables require VMware Tools and valid guest login. With Linux, setting the guest environment also requires root login. Provide the variable name and its value.	<path to `.vmx` file> `[ guestVar | runtimeConfig | guestEnv ]` <variable name> <variable value>
`readVariable`	Reads a variable from the virtual machine state or guest. You can get a guest variable, a runtime configuration as stored in the `.vmx` file, or environment variables in the guest operating system. Reading the `guestEnv` requires a valid guest login. See above for a description of variable types.	<path to `.vmx` file> `[ guestVar | 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 the product documentation.	<path to `.vmx` file>
`register` (Registration not supported on Workstation or on VMware Fusion.)	Registers the specified virtual machine, adding it to the host's inventory. Path format depends on the product. For ESX/ESXi hosts, `"[datastore1] vm/vm.vmx"` (starting with the datastore) is typical.	<path to `.vmx` file>
`unregister` (Registration not supported on Workstation or on VMware Fusion.)	Unregisters the specified virtual machine, removing it from the host's inventory. Path format depends on the product. For ESX/ESXi, `"[datastore1] vm/vm.vmx"` (starting with the datastore) is typical.	<path to `.vmx` file>
`listRegisteredVM`	Lists all registered virtual machines.	None
`deleteVM`	Removes the specified virtual machine.	<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 both the source and the destination .vmx file pathname. You can create either a full clone or a linked clone. To create 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 Workstation and VMware Fusion.)		
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 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 Workstation for Linux:

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

Reboot a virtual machine running on Workstation for Windows:

```
cd "C:\Documents and Settings\<user>\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 10.0.1.8 –u root reset "[datastore1] WinXP/WinXP.vmx" soft
Host password: <pass>
```

Power Commands

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

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

This error message following this command 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 Workstation on a Windows host:

```
vmrun stop "C:\Documents and Settings\<user>\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 10.0.1.8 -u root -p <pass> start "[datastore1] WinXP/WinXP.vmx"
```

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

```
vmrun -T esx -h 10.0.1.9:8333 -u root -p <pass> start "[datastore1] 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 10.0.1.8 -u root -p <pass> start "[datastore1] WinXP/WinXP.vmx"
```

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

```
vmrun -T vc -h 10.0.1.8 -u Administrator -p <pass> start "[datastore1] WinXP/WinXP.vmx"
```

Snapshot Commands

Create a snapshot of a virtual machine with 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
```

Running Guest Applications

Most vmrun guest operations require VMware Tools to be installed on the guest operating system.

Start the command tool, minimized, on a Windows guest:

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

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

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

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

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

Run a batch script and keep running afterwards. To use cmd.exe on Windows, you must specify the script 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 (this requires the -display option to appear on the console).

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

Run the same X clock command, but return control back to the console immediately:

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

New versions of Firefox support the --display flag from X11, so the option looks a bit different:

```
vmrun -gu <user> -gp <pass> runProgramInGuest Ubuntu/Ubuntu.vmx /usr/bin/firefox --display=:0
```

For commands that require an environment variable, you can set it first, or export it for multiple commands:

```
vmrun -gu <user> -gp <pass> runProgramInGuest SUSE/SUSE.vmx "DISPLAY=:0 /usr/bin/salesgui"
vmrun -gu <user> -gp <pass> runProgramInGuest SUSE/SUSE.vmx "export REL=7; /opt/cmd1; /opt/cmd2"
```

Setting the guest environment with `guestEnv` requires root permission on Linux, because the change affects subsequent commands issued by other users:

```
vmrun -T ws -gu root -gp <rootpass> writeVariable SUSE/SUSE.vmx guestEnv SRC tmp.example.com:1666
```

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 10.0.1.8 -u root -p <pass> -gu <user> -gp <userpass> runProgramInGuest
                "[datastore1] RHEL4/RHEL4.vmx" /usr/X11R6/bin/xclock -display :0
```

Guest to Host File Operations

To copy a 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 a 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 On Windows 7 and Vista, only the Administrator account can use `copyFileFromHostToGuest` and `deleteFileInGuest` to write and delete files in `C:\` and system folders, or use `createDirectoryInGuest` and `deleteDirectoryInGuest` to modify system directories. Regular users, even those with administrator privilege, cannot perform these operations.

Guest Variables and Environment

From the host, set a guest variable on two virtual machines:

```
vmrun writeVariable WindowsXP/WindowsXP.vmx guestVar vmstartdate 21April2011
vmrun writeVariable Ubuntu10/Ubuntu10.vmx guestVar vmstartdate 21April2011
```

On the guest operating systems, read the guest variable that you just set:

```
> vmware-rpctool.exe "info-get vmstartdate"
$ vmware-rpctool "info-get vmstartdate"
```

From the host, set a guest environment variable on a Linux virtual machines, and verify by writing the environment variables into a temporary file:

```
vmrun writeVariable Ubuntu10/Ubuntu10.vmx guestEnv LD_LIBRARY_PATH /usr/local/lib
Guest user: root
Guest password:
vmrun runScriptInGuest Ubuntu10/Ubuntu10.vmx /bin/bash "usr/bin/env > /tmp/env.out"
Guest user: root
Guest password:
```

No output comes to the host from `runScriptInGuest`, so look at `/tmp/env.out` on the guest.

On a Linux guest, determine the IP address and set it in a guest variable:

```
$ ipaddr=‘ifconfig eth0 | grep inet.addr‘
$ vmware-rpctool "info-set guestinfo.ip $ipaddr"
```

From the host, retrieve the IP address that was just set on the guest:

```
vmrun readVariable Ubuntu10/Ubuntu10.vmx guestVar guestinfo.ip
```

Maintenance Commands

List running virtual machines on 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\Windows7\Windows7.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 10.0.1.5 -u root -p <pass> register "[datastore1] RHEL5/RHEL5.vmx"
```

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

```
vmrun -T esx -h 10.0.1.5 -u root -p <pass> unregister "[datastore1] RHEL3/RHEL3.vmx"
```

On vCenter Server, the `vmrun` utility supports the same form of path-to-VMX specification as ESX/ESXi hosts.

```
vmrun -T vc -h 10.0.1.9 -u Administrator -p <pw> register "[datastore1] RHEL5/RHEL5.vmx"
```

You cannot reach a virtual machine file through its resource pool or vApp.

Index