

Using vmrun to Control Virtual Machines

For VMware Workstation 6.5, VMware Fusion 2.0, and VMware Server 2.0

Beta



Using vmrun to Control Virtual Machines

Revision: 20080724

Item: EN-000068-00

You can find the most up-to-date technical documentation on our 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

Beta

© 2008 VMware, Inc. All rights reserved. Protected by one or more U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022, 6,944,699, 6,961,806, 6,961,941, 7,069,413, 7,082,598, 7,089,377, 7,111,086, 7,111,145, 7,117,481, 7,149,843, 7,155,558, 7,222,221, 7,260,815, 7,260,820, 7,269,683, 7,275,136, 7,277,998, 7,277,999, 7,278,030, 7,281,102, 7,290,253, and 7,356,679; patents pending.

VMware, the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks 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

About This Book

This manual, *Using vmrun to Control Virtual Machines*, introduces the `vmrun` command, a convenient command to help you manage the collection of virtual machines on a 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
20080623	First draft of this manual for the VMware Server 2.0 RC1 and Workstation 6.5 Beta2 releases.
20080724	Second draft including support for VMware Fusion on Intel-based Macintosh OS X hosts.

VMware provides many different SDK products targeting different developer communities and platforms. For information about SDK products, go to http://www.vmware.com/support/pubs/sdk_pubs.html. This is also the place to find the most up-to-date documentation.

Intended Audience

This book is intended for developers and system administrators who want to control virtual machines on various platforms, including VMware Workstation, VMware Fusion, and VMware Server.

Document Feedback

VMware welcomes your suggestions for improving our documentation. Send your comments to:

docfeedback@vmware.com

Technical Support and Education Resources

The following sections describe the technical support resources available to you. You can access the most current versions of other VMware manuals by going to:

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

Online Support

You can submit questions or post comments to the Developer Community: SDKs and APIs forum, which the VMware technical support and product teams monitor. You can access the forum at:

<http://communities.vmware.com/community/developer>

Support Offerings

To find out how VMware support offerings can help meet your business needs, go to:

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

VMware Education Services

VMware courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. For more information about VMware Education Services, go to:

<http://mylearn1.vmware.com/mgrreg/index.cfm>.

Using vmrun to Control Virtual Machines

VMware provides an application called `vmrun` for controlling virtual machines, or teams of virtual machines, from the command line. The `vmrun` command is available on any VMware product that includes the VIX API libraries, or when the libraries are separately installed. This document contains the following sections:

- [“About the vmrun Command,”](#) below.
- [“Using vmrun on Windows”](#) on page 6
- [“Using vmrun on Linux”](#) on page 6.
- [“Using vmrun on Mac OS X”](#) on page 7
- [“Virtual Machine Run Flags”](#) on page 7.
- [“Virtual Machine Run Commands”](#) on page 7.
- [“Examples of Using vmrun”](#) on page 11.

About the vmrun Command

The `vmrun` command manipulates virtual machines and runs on any VMware platform where you can install the VIX libraries, including VMware Workstation, VMware Fusion, and VMware Server. For information about the facilities that `vmrun` controls, see the documentation for your product, such as the *User's Manual*.

This document organizes `vmrun` commands into the six following categories.

Power Commands

Virtual machine power operations give you these options: start (power on), stop (power off), reset (reboot), suspend (but allow local work to resume), pause (without interrupting), and unpause (continue).

On some products, you can group virtual machines into teams, so power operations apply to the whole team.

Snapshot Commands

A snapshot reproduces virtual machine state at the time of the snapshot, including all data on virtual disks. You can snapshot a virtual machine as you choose, in any power state, and revert to the snapshot at any time. Snapshots are useful for experimentation, and especially useful for backups.

These commands list existing snapshots of a virtual machine, create a new snapshot, delete a snapshot, and revert a virtual machine to its state as of a specific snapshot. VMware Server limits you to one snapshot.

Record and Replay Commands

Virtual machine events can be recorded for later replay. The recording is called a snapshot object, but is really more like a movie. At this time, only VMware Workstation supports record and replay.

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

Guest Operating System Commands

These are the most distinctive and useful interfaces in the `vmrun` command. You can:

- Run an executable program in the guest operating system, or run an interpreted script. These interfaces serve a similar purpose, but `runProgramInGuest` provides more fine-grained 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 (except on VMware Server).
- List the processes running in the guest operating system, or terminate any process (with permission).
- Read or write a variable into the guest operating system's environment or virtual machine state.

General Commands

This catch-all category includes commands to list all running virtual machines, upgrade the virtual machine hardware version, and install VMware Tools semi-automatically on the guest.

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

VMware Server allows you to register and unregister virtual machines.

VProbes Commands

These commands support VProbes, a facility for dynamically and statically instrumenting portions of VMware software and a running guest operating system. See the *VProbes Programming Reference* for details.

Using vmrun on Windows

To use the `vmrun` command on a Windows system:

- 1 Locate the `vmrun` program, which by default is installed here:

```
C:\Program Files\VMware\VMware VIX
```

- 2 Add the install location to your system path. On Windows XP for example, choose:

Computer > Properties > Advanced > Environment Variables > System variables > Path > Edit

Using the right arrow key, move the cursor to the end of line, add a semicolon, the software location path (as in step 1) and click OK several times.

If VMware Workstation is already in Path, this step is optional because another `vmrun` is installed there.

- 3 In a command window, type the `vmrun` command in the following form

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

- 4 Continue with ["Virtual Machine Run Flags"](#) on page 7.

Using vmrun on Linux

To use the `vmrun` command on a Unix system:

- 1 (Optional) As root or superuser, edit the `/etc/ld.so.conf` file, add a line with the library location default directory below, save the file, and run the `ldconfig` command.

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

- 2 In a command or terminal window, type the `vmrun` command in the following form

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

- 3 Continue with ["Virtual Machine Run Flags"](#) on page 7.

Using vmrun on Mac OS X

To use the `vmrun` command on an Intel-based Macintosh system:

- 1 In a Terminal window, add the VMware Fusion directory to your system path:

```
export PATH=$PATH:/Library/Application\ Support/VMware\ Fusion
```

- 2 Type the `vmrun` command in this form:

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

- 3 Continue with “[Virtual Machine Run Flags](#)” on page 7.

Virtual Machine Run Flags

On VMware Workstation and VMware Fusion, `vmrun` controls guest operating systems on the local host, so you do not need to specify a remote host name or port. Possibly no flags are required.

In the syntax examples below, flags enclosed in angle brackets indicate variables that you fill in.

For commands that require authentication by the guest operating system, the command description states “valid guest login” required, in which case you must use the following flags:

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

To set the host type for remote access to VMware Server 2.0, use the following flags in conjunction:

```
-T server  
-h <hostName>  
-P <portNumber>  
-u <adminLogin on Server>  
-p <adminPassword on Server>
```

For example, this command lists all running virtual machines on a remote server:

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

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

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

For VMware Server 1.0, use the `-T` flag as follows:

```
vmrun -T server1
```

NOTE On VMware Workstation and VMware Fusion, starting a virtual machine with the default `gui` option requires a window system to be running on the host. VMware Server does not impose this requirement.

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 if you want to overwrite your existing snapshot.

VMware Fusion does not support snapshot trees.

Virtual Machine Run Commands

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

- On Windows
 - path on VMware Workstation:


```
C:\Documents and Settings\<username>\My Documents\My Virtual Machines\Win XP\Win XP.vmx
```
 - data store on VMware Server:


```
[storage1] Win XP/Win XP.vmx
```

- On Linux
 - path on VMware Workstation:
/home/<username>/VirtualMachines/Ubuntu/Ubuntu.vmx
 - data store on VMware Server:
[storage1] Win XP/Win XP.vmx
- On Mac OS X
~/Documents/Virtual Machines.localized/Windows XP Home.vmwarevm/Windows XP Home.vmx

All the `vmrun` commands and parameters are grouped and documented in [Table 2](#). 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		
<code>start</code> (Teams not supported on VMware Server or on VMware Fusion.)	Start a virtual machine (.vmx file) or team (.vmtm file). The default <code>gui</code> option starts interactively, which is required to display a Workstation user interface. The <code>nogui</code> option suppresses the user interface, including startup dialog, to allow non-interactive scripting.	<path to .vmx or .vmtm file> [gui nogui]
<code>stop</code> (Teams not supported on VMware Server or on VMware Fusion.)	Stop 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 stop using the <code>powerType</code> specified in the .vmx file, if present.	<path to .vmx or .vmtm file> [hard soft]
<code>reset</code> (Teams not supported on VMware Server or on VMware Fusion.)	Reset 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 reset using the <code>powerType</code> specified in the .vmx file, if present.	<path to .vmx or .vmtm file> [hard soft]
<code>suspend</code> (Teams not supported on VMware Server or on VMware Fusion.)	Suspend 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, while on UNIX guests they suspend networking. The <code>hard</code> parameter suspends the guest without running these scripts. The default is to suspend using <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 and on Linux networking is restarted.	<path to .vmx or .vmtm file> [hard soft]
<code>pause</code>	Pause a virtual machine (.vmx file). This is used to pause replay, but can be used in other contexts.	<path to .vmx file>
<code>unpause</code>	Unpause a virtual machine (.vmx file) and resume operation where it paused.	<path to .vmx file>
Snapshot Commands		
<code>listSnapshots</code>	List all snapshots in a virtual machine (.vmx file).	<path to .vmx file>
<code>snapshot</code> (VMware Server does not support multiple snapshots. VMware Fusion does not support snapshot trees.)	Create 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 path names, VMware recommends that you avoid using the slash character when naming a snapshot, because that makes it difficult to specify the snapshot path precisely.	<path to .vmx file> <snapshot name>

Table 2. vmrun Commands and Parameters (Continued)

Command	Description	Parameters
<code>deleteSnapshot</code> (VMware Server always deletes the root snapshot.)	Remove 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 chain.	<path to .vmx file> <snapshot name>
<code>revertToSnapshot</code> (VMware Server always reverts to the root snapshot.)	Set the virtual machine state to a snapshot. 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 snapshot name (first group of parameters). If several snapshots have the same name, you can still specify the snapshot by including a "path name" for the snapshot name. A path name is a series of snapshot names, separated by forward slash characters (/). Each name specifies a different snapshot in the tree. For example, the path name <code>Snap1/Snap2</code> identifies a snapshot named <code>Snap2</code> that was taken from the state of a root snapshot named <code>Snap1</code> .	<path to .vmx file> <snapshot name> or <path to .vmx file> <snap1/snap2/snapN>
Record and Replay Commands		
<code>beginRecording</code> (Recording not supported on VMware Server or on VMware Fusion.)	Begin 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>]
<code>endRecording</code>	End the recording of a virtual machine (.vmx file) that is currently active, and close its snapshot object.	<path to .vmx file>
<code>beginReplay</code> (Replay not supported on VMware Server or on VMware Fusion.)	Begin replaying the recorded activity of a powered off virtual machine (.vmx file) from a snapshot object. Only one recording or replay can be active at a time.	<path to .vmx file> <snapshot object name>
<code>endReplay</code>	End the replaying of a virtual machine (.vmx file) that is currently underway.	<path to .vmx file>
Guest Operating System Commands		
<code>runProgramInGuest</code>	Run a program in the guest operating system. Provide the full path name of a program accessible to the guest. VMware Tools and valid guest login are required. Also provide full accessible path names for any files specified in the program arguments. The <code>-noWait</code> option means to return immediately after the program starts in the guest, rather than waiting for it to finish. This is useful for interactive programs. The <code>-activeWindow</code> flag ensures that the Windows GUI is visible, not minimized. It has no effect on Linux. The <code>-interactive</code> flag forces interactive guest login and is useful for Windows Vista guests, to make the program visible in the console window.	<path to .vmx file> [<code>-noWait</code> <code>-activeWindow</code> <code>-interactive</code>] <program name> [<program arguments>]
<code>fileExistsInGuest</code>	Check whether the specified file exists in the guest operating system. VMware Tools and valid guest login are required.	<path to .vmx file> <path to file in guest>
<code>setSharedFolderState</code> (VMware Server does not support shared folders.)	Modify the writeability 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> <code>writable</code> <code>readonly</code>

Table 2. vmrun Commands and Parameters (Continued)

Command	Description	Parameters
<code>addSharedFolder</code> (VMware Server does not support shared folders.)	Add 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.	<path to .vmx file> <share name> <path to folder on host>
<code>removeSharedFolder</code> (VMware Server does not support shared folders.)	Remove 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>listProcessesInGuest</code>	List all processes running in the guest operating system. VMware Tools and valid guest login are required.	<path to .vmx file>
<code>killProcessInGuest</code>	Stop the specified process in the guest operating system. VMware Tools and 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>	Run a command script in the guest operating system. VMware Tools and valid guest login are required. The interpreter path is the command that run the script. Give the complete text of the script, not a file name.	<path to .vmx file> <interpreter path> <script text>
<code>deleteFileInGuest</code>	Delete the specified file from the guest operating system. VMware Tools and valid guest login are required.	<path to .vmx file> <path to file on guest>
<code>createDirectoryInGuest</code>	Create specified directory in the guest operating system. VMware Tools and valid guest login are required.	<path to .vmx file> <directory path on guest>
<code>deleteDirectoryInGuest</code>	Delete a directory from the guest operating system. VMware Tools and valid guest login are required.	<path to .vmx file> <directory path on guest>
<code>listDirectoryInGuest</code>	List directory contents in the guest operating system. VMware Tools and valid guest login are required.	<path to .vmx file> <directory path on guest>
<code>copyFileFromHostToGuest</code>	Copy a file from the host to the guest operating system. VMware Tools and valid guest login are required. Specify source file (host) before destination file (guest).	<path to .vmx file> <file path on host> <file path in guest>
<code>copyFileFromGuestToHost</code>	Copy a file from the guest operating system to the host. VMware Tools and valid guest login are required. Specify source file (guest) before destination file (host).	<path to .vmx file> <file path in guest> <file path on host>
<code>renameFileInGuest</code>	Rename or move a file in the guest operating system. VMware Tools and valid guest login are required. Specify source name (original) before destination (new).	<path to .vmx file> <original file name> <new file name>
<code>captureScreen</code>	Capture 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>
<code>writeVariable</code>	Write a variable into 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 valid guest login. Provide the variable name and its value.	<path to .vmx file> [runtimeConfig guestEnv] <variable name> <variable value>
<code>readVariable</code>	Read 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 valid guest login.	<path to .vmx file> [runtimeConfig guestEnv] <variable name>
General Commands		
<code>list</code>	List all running virtual machines.	None
<code>upgradem</code>	Upgrade a virtual machine to the current version of virtual hardware. Has no effect if already current.	<path to .vmx file>

Table 2. vmrun Commands and Parameters (Continued)

Command	Description	Parameters
installtools	Prepare to install VMware Tools in the guest operating system. In Windows guests with autorun enabled, the VMware Tools installer starts automatically. 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 run automatically, so 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 on VMware Fusion.)	Register 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 data store) is typical.	<path to .vmx file>
unregister (Registration not supported on VMware Workstation or on VMware Fusion.)	Unregister 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 data store) is typical.	<path to .vmx file>
clone (Cloning not supported on VMware Server or on VMware Fusion.)	Create a copy of the virtual machine and guest operating system. Provide the source .vmx file path name, and the destination .vmx file path name. You can create either a normal full clones, or a linked clone. You may specify a snapshot name if you want to make the clone from this snapshot, rather than from current virtual machine state.	<path to .vmx file> <destination .vmx file path> full linked [<snapshot name>]
VProbes Commands		
(VMware Server does not support VProbes.)		
vprobeVersion	Show VProbes version.	<path to .vmx file>
vprobeLoad	Load VProbes script.	<path to .vmx file> <text of VProbes script>
vprobeReset	Disable all VProbes.	<path to .vmx file>
vprobeListProbes	List active VProbes.	<path to .vmx file>
vprobeListGlobals	List VProbes global variables.	<path to .vmx file>

Examples of Using vmrun

Most examples work on any either VMware Workstation or VMware Fusion. The `-T ws` and `-T fusion` options are synonymous. Derive the guest virtual machine type by distinguishing `/` for Linux and `\` for Windows.

New Examples

Start an X terminal on a Linux guest (requires `-display` option to appear on the console):

```
vmrun -gu <user> -gp <password> runProgramInGuest Ubuntu/Ubuntu.vmx /usr/bin/xterm "-display :0"
```

List processes in a Linux guest on Workstation, and kill the process numbered 20001:

```
vmrun -T ws -gu guestUser -gp guestPassword listProcessesInGuest Ubuntu/Ubuntu.vmx
vmrun -T ws -gu guestUser -gp guestPassword killProcessesInGuest Ubuntu/Ubuntu.vmx 20001
```

Run a batch script on a Windows guest using Workstation:

```
vmrun -T ws -gu guestUser -gp guestPassword runProgramInGuest WinXP.vmx "C:\Workarea\script.bat"
```

Run a command script on a Windows guest using Workstation:

```
vmrun -T ws -gu guestUser -gp guestPassword runProgramInGuest WinXP.vmx
C:\Windows\System32\cmd.exe "/c C:\Workarea\script.cmd"
```

Prepare to install VMware Tools on VMware Fusion:

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

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

```
vmrun -T ws -gu guestUser -gp guestPassword runScriptInGuest Linux.vmx "/usr/bin/perl -pi"
"open(DOS, '/tmp/dos.txt'); while (<DOS>) { s/\r\n/\n/ ; print <DOS>}"
```

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

```
vmrun -T ws -gu guestUser -gp guestPasswd runScriptInGuest WinXP.vmx "C:\cygwin\bin\perl.exe -pi"
"open(UNIX, 'C:/Temp/unix.txt'); while (<UNIX>) { s/\n/\r\n/ ; print <UNIX>}"
```

Examples From the Usage Message

Start a virtual machine with VMware Workstation or VMware Fusion on a Windows host:

```
vmrun -T ws start "c:\my VMS\myVM.vmx"
```

Stop a virtual machine with Server on a Linux host:

```
vmrun -T server -h https://myHost.com/sdk -u hostUser -p hostPasswd stop "[storage1] vm/myVM.vmx"
```

Run a program in a virtual machine on a Windows host with a Windows guest:

```
vmrun -T ws -gu guestUser -gp guestPassword runProgramInGuest "c:\my VMS\myVM.vmx"
"c:\Program Files\myProgram.exe"
```

Run a program in a virtual machine with VMware Server on a Linux host with a Linux guest:

```
vmrun -T server -h https://myHost.com/sdk -u hostUser -p hostPasswd -gu guestUser -gp guestPasswd
runProgramInGuest "[storage1] vm/myVM.vmx" /usr/bin/X11/xclock -display :0
```

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

```
vmrun -T ws snapshot "c:\my VMS\myVM.vmx" mySnapshot
```

Revert to a snapshot with VMware Workstation or VMware Fusion on a Windows host:

```
vmrun -T ws revertToSnapshot "c:\my VMS\myVM.vmx" mySnapshot
```

Delete a snapshot with VMware Workstation or VMware Fusion on a Windows host:

```
vmrun -T ws deleteSnapshot "c:\my VMS\myVM.vmx" mySnapshot "C:\Program Files\VMware\VMware
Workstation"
```

Examples from Workstation Manual

Reset a virtual machine on Linux:

```
vmrun reset /usr/local/VMS/<virtual_machine_name>.vmx soft
```

Reset a virtual machine on Windows:

```
vmrun reset C:\Virtual Machines\<virtual_machine_name>.vmx soft
```

Disabling Dialog Boxes

With virtual machines that require input through a VMware Workstation dialog box, the `vmrun` command might time out and fail. To disable Workstation dialog boxes, insert the following line into the configuration file (`.vmx`) for a virtual machine:

```
msg.autoAnswer = TRUE
```