

# Using vmrun to Control Virtual Machines

VMware Workstation 6.5 and VMware Server 2.0



## Using vmrun to Control Virtual Machines

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](mailto:docfeedback@vmware.com)

© 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, 7,356,679, 7,409,487, 7,412,492, and 7,412,702; 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](http://www.vmware.com)

# 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
20080623	First draft of this manual for the VMware Server 2.0 RC1 and Workstation 6.5 Beta2 releases.
20080815	Third draft of this manual for the VMware Server 2.0 RC2 and Workstation 6.5 RC releases.

## Intended Audience

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

## Documentation Resources

To access the current versions of VMware API and SDK documentation, go to:

[http://www.vmware.com/support/pubs/sdk\\_pubs.html](http://www.vmware.com/support/pubs/sdk_pubs.html)

To access the current versions of other VMware manuals, go to:

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

## Documentation Feedback

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

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

## Technical Support and Education Resources

The following sections describe the technical support and educational resources available to you.

### Online Support

You can submit questions or post comments to the Developer Community SDKs and APIs forum, which is monitored by VMware technical support and product teams. To access the forum, go to:

<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 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 any VMware product that includes the VIX API libraries, or when the libraries are separately installed. This chapter contains the following sections:

- [“About the vmrun Utility”](#) on page 5
- [“Setting Up vmrun on Windows”](#) on page 6
- [“Setting Up vmrun on Linux”](#) on page 6
- [“Flags to Specify Virtual Machine Type”](#) on page 7
- [“Virtual Machine Run Syntax”](#) on page 7
- [“Examples of Using vmrun”](#) on page 11

## About the vmrun Utility

The `vmrun` utility runs on any VMware platform with VIX libraries installed, including VMware Workstation and VMware Server. For information about the various facilities that `vmrun` controls, see the user documentation for your product.

You can use `vmrun` to perform various tasks on virtual machines, summarized below.

### 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, 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 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

You can record virtual machine events 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, 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 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.

## 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 any virtual machine image to another virtual machine.

VMware Server allows you to register and unregister virtual machines.

## VProbes Commands

You can use `vmrun` to script VProbes, a facility for transparently instrumenting a powered-on guest operating system, its running processes, and virtualization software. See the *VProbes Programming Reference* for details.

## Setting Up vmrun on Windows

### To use the `vmrun` utility on Windows

- 1 Locate the `vmrun` utility, which gets installed in this folder by default:

```
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**

If VMware Workstation is already in your system Path, this step is unnecessary because a copy of `vmrun` is also installed there.

Using 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 [“Flags to Specify Virtual Machine Type”](#) on page 7.

## Setting Up vmrun on Linux

### To use the `vmrun` utility on Linux

- 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 `vmrun` to see command-line options.
- 3 Continue with [“Flags to Specify Virtual Machine Type”](#) on page 7.

## Flags to Specify Virtual Machine Type

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

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

In syntax examples, options enclosed in angle brackets indicate variables that you fill in. On VMware Workstation, `vmrun` controls guest operating systems on the local host, so you need not specify a remote host name or port. Possibly no flags are required.

For commands that require authentication by the guest operating system, the command descriptions in [Table 2, “vmrun Commands and Parameters,”](#) on page 8 state “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:

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

If not specified, 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 server -h https://example.com:8333/sdk -u root -p secretpw list
```

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

```
vmrun -T ws
```

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

```
vmrun -T server1
```

---

**NOTE** On VMware Workstation, starting (powering on) a virtual machine with the default `gui` option requires a window system (user interface) 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.

## 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 some examples of where the `.vmx` file might be located:

- VMware Server datastore:  
[storage1] 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

## Using 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
<b>Power Commands</b>		
<b>start</b> (Teams not supported on VMware Server.)	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> ]
<b>stop</b> (Teams not supported on VMware Server.)	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> ]
<b>reset</b> (Teams not supported on VMware Server.)	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> ]
<b>suspend</b> (Teams not supported on VMware Server.)	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> ]
<b>pause</b>	Pauses a virtual machine (.vmx file). You can use this either to pause replay, or to pause normal operation.	<path to .vmx file>
<b>unpause</b>	Resumes operation of a virtual machine (.vmx file) from where you paused replay or normal operation.	<path to .vmx file>
<b>Snapshot Commands</b>		
<b>listSnapshots</b>	Lists all snapshots in a virtual machine (.vmx file).	<path to .vmx file>
<b>snapshot</b> (VMware Server does not support multiple snapshots.)	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 path names, 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>
<b>deleteSnapshot</b> (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.	<path to .vmx file> <snapshot name>



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

Command	Description	Parameters
revertToSnapshot (VMware Server always reverts to the root snapshot.)	Sets the virtual machine to its state at snapshot time. 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 path name for the snapshot. A path name is a series of snapshot names, separated by forward slash characters (/). Each name specifies a new snapshot in the tree. For example, the path name 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>
<b>Record and Replay Commands</b>		
beginRecording (Recording not supported on VMware Server.)	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 not supported on VMware Server.)	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 the replaying of a virtual machine (.vmx file) that is currently underway.	<path to .vmx file>
<b>Guest Operating System Commands</b>		
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 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, 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>]
runScriptInGuest	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>
setSharedFolderState (VMware Server does 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> <code>writable</code>   <code>readonly</code>
addSharedFolder (VMware Server does 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.	<path to .vmx file> <share name> <path to folder on host>

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

Command	Description	Parameters
removeSharedFolder (VMware Server does 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>
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. Take process ID from the 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>
deleteFileInGuest	Deletes a specified file from the guest operating system. VMware Tools and a valid guest login are required.	<path to .vmx file> <path to file on guest>
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>
createDirectoryInGuest	Creates 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 a directory from the guest operating system. VMware Tools and a valid guest login are required.	<path to .vmx file> <directory path on guest>
listDirectoryInGuest	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>
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).	<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>
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 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 a valid guest 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>
<b>Maintenance Commands</b>		
list	Lists all running virtual machines.	None
upgradevm	Upgrades 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	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.)	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.)	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>
clone (Cloning not supported on VMware Server.)	Creates 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 clone or a linked clone. If you want 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>]
<b>VProbes Commands</b>		
(VMware Server does not support VProbes.)		
vprobeVersion	Shows the VProbes version on the virtual machine.	<path to .vmx file>
vprobeLoad	Loads the VProbes script on the virtual machine.	<path to .vmx file> <text of VProbes script>
vprobeReset	Disables all VProbes on the virtual machine.	<path to .vmx file>
vprobeListProbes	Lists the active VProbes on the virtual machine.	<path to .vmx file>
vprobeListGlobals	Lists VProbes global variables on the virtual machine.	<path to .vmx file>

## Disabling Dialog Boxes

With virtual machines that require input through a VMware Workstation dialog box, the vmrun utility 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
```

## Examples of Using vmrun

Command-line examples below work either on VMware Workstation (-T ws) or VMware Server (-T server).

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 Server:

```
vmrun -T server -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 installed VIX package 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"
```

Using VIX to control a VMware Server with IP address 10.0.1.8, power on a virtual machine:

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

Using VIX to control a VMware Server with IP address 10.0.1.8, power off the virtual machine:

```
vmrun -T server -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.

## Snapshot Commands

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

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

List snapshots on the virtual machine, showing the one just made:

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

Revert to the snapshot just made, which suspends the virtual machine, and 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, specifying it by name:

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

## Record and Replay Commands

Start recording user events on a Windows guest, starting with a snapshot of 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, using `cmd.exe /c` as a command interpreter:

```
vmrun -T ws -gu <user> -gp <pass> runProgramInGuest WinXP\WinXP.vmx cmd.exe "/c C:\runit.bat"
```

With VMware Workstation, run a Bash shell script 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):

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

Same as above, but immediately return to command-line control:

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

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

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

Run a program in a Linux virtual machine on VMware Server:

```
vmrun -T server -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
```

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 into 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
```

## 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 using shared folders, you must enable them by selecting **VM > Settings > Options > Shared Folders > Enabled** in the user interface. To share a folder on a Windows host with a particular Linux guest:

```
vmrun -T ws addSharedFolder Ubuntu\Ubuntu.vmx /mnt/hgfs/Share C:\Share
```

Shared folders are writable by default. To make a shared folder read-only:

```
vmrun -T ws setSharedFolderState Ubuntu\Ubuntu.vmx /mnt/hgfs/Share C:\Share readonly
```

To stop sharing the folder:

```
vmrun -T ws removeSharedFolder Ubuntu\Ubuntu.vmx /mnt/hgfs/Share
```

## 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
```

Register a new virtual machine installed on VMware Server:

```
vmrun -T server -h https://10.0.1.8/sdk -u root -p <pass> register "[storage1] RHEL5/RHEL5.vmx"
```

Unregister an old virtual machine going out of service on VMware Server:

```
vmrun -T server -h https://10.0.1.8/sdk -u root -p <pass> unregister "[storage1] RHEL3/RHEL3.vmx"
```