



VMware Workstation

Using Clones to Streamline Development

In recent years, the trend in software development has been toward applications that are configured on multitier platforms, and delivered via Web browser across the Internet or an intranet. While this model, with browser-based delivery of applications and services, offers irresistible cost savings in delivering and supporting products, it also greatly increases the complexity involved in creating those products. To produce applications for this environment, software developers and testers now require a multitiered platform under their control — deployed, maintained, and readily available to Development and QA.

Browser-based multitier applications must be developed and tested against an ever wider variety of hardware platforms, operating systems, and browsers — all with multiple versions. Many software companies report that the overhead involved in IT setup and teardown of lab machines consumes 25 to 30 percent of the development budget, a staggering cost.

This technical note shows one way to address this challenge through virtualization, by using Workstation 5, with its linked clones and multiple snapshots features, to set up a repository of clean development environments in a series of virtual machines from which developers can quickly clone and download a desired setup in minutes:

- [Traditional Approaches to Providing Development Environments on page 1](#)
- [Virtualization: VMware Workstation Provides an Efficient Alternative on page 2](#)
- [Workstation 5 Cloning and Multiple Snapshot Features on page 2](#)
- [Scenario — Creating and Using a Repository of Test Environments in Virtual Machines on page 3](#)
- [Cloning a Virtual Machine from the Repository on page 7](#)

Traditional Approaches to Providing Development Environments

One traditional way to create a development and test environment is to set up a lab, with a pool of machines shared by Development and QA and maintained by an IT department. This saves money in hardware, but at a great cost in efficiency. Typically in this arrangement, a developer must request access to a particular configuration of hardware, operating system, and browser, which in turn must be set up by IT. In essence, the software engineer must make an appointment to gain access to the necessary development environment, and IT personnel must set up and tear down, again and again, every possible configuration needed by each developer and tester. For the developer and tester, this arrangement is frustrating and inefficient: they must contend and wait for restricted access to a limited number of machines. Inevitably, this results in dead time for the engineer, delays in production, and an increase in the time to market for the product.

Alternatively, some companies have chosen to provide each developer and tester with a separate hardware setup, with servers and client machines on every desk. However, while this strategy guarantees developers access to the machines they need, the hardware costs can quickly become prohibitive.



Perhaps even more significantly, this strategy does little to reduce the huge overhead associated with setting up and tearing down development and test configurations. The burden on IT may be reduced somewhat, but only because the setup and teardown work has been assumed by the individual developers and testers. The overhead remains; it has merely been distributed over a wider range of personnel.

Virtualization: VMware Workstation Provides an Efficient Alternative

In recent years, virtualization has changed this landscape considerably. VMware Workstation enables developers to store any number of standard x86-based configurations as virtual machines, already loaded with any of a wide range of Windows and Linux operating systems, browsers, and other applications. Engineers can use these virtual machines, instead of physical hardware, to develop and test multitiered applications on a single piece of hardware.

Because virtual machines are software entities, they can be easily copied, cloned, reset, and reused with a few mouse clicks. A company can create a repository of pre-loaded virtual machines in every desired configuration. Each machine requires just a single setup. From these template virtual machines, individual developers can download a clean, "just-installed" configuration they need, simply by cloning the appropriate virtual machine.

Workstation 5 Cloning and Multiple Snapshot Features

The new Workstation 5 linked clone and multiple snapshot features make it easy to set up a repository of development environments in virtual machines. For example, you can create a virtual machine, install an operating system, and load various service packs, saving each as a separate snapshot (a scenario discussed in detail in this tech note). A developer can then open the virtual machine from your repository, go to the appropriate snapshot, and clone it to a local drive in a few minutes.

Using linked clones makes it even faster to download a clean environment into a virtual machine. When you make a linked clone from a virtual machine, the clone uses the virtual disk of the parent virtual machine it was cloned from. Since virtual machines can be large, this dramatically reduces the time it takes to clone a virtual machine, and the amount of disk space the clone uses.

By making it so easy to quickly create a lightweight clone of an existing environment, linked clones also facilitate collaboration among developers and testers. For example, suppose a developer is working on a bug and wants help from other developers or testers. The entire setup containing the bug — including the operating system and all applications involved — can be captured in a snapshot, or a series of snapshots, and the virtual machine can be stored on the network. Then, any developers with network access can download a lightweight linked clone in minutes. When the problem is solved, teardown is as simple as deleting the clone.

Of course, if the clone will be used in a laptop, or some other setting where it will not have access to the parent virtual machine, you can make a full clone, which is independent of the parent machine. You might also choose to make a full clone if performance is more important than conserving disk space or speeding the cloning process. Because a full clone's virtual disk is local (and not shared with the parent virtual machine across a network), it can perform better.



Scenario — Creating and Using a Repository of Test Environments in Virtual Machines

One way you can use clones to make your development environment more efficient is by creating a repository of test environments. You can store a range of environments as a series of snapshots in virtual machines. Then your engineers can clone any of those snapshots as a separate virtual machine that they can download and use.

For this scenario, suppose that you want to establish a repository of Windows 2000 environments at each Service Pack level that you develop and test in.

Create a Virtual Machine and Snapshots

In Workstation 5:

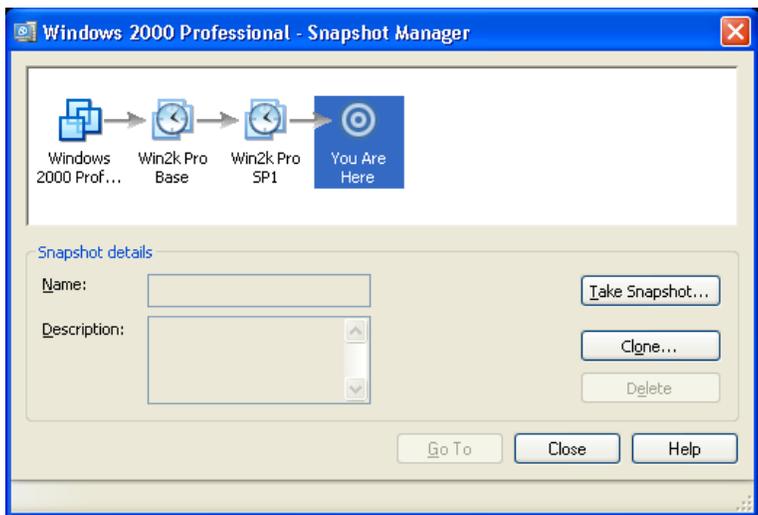
1. Begin by creating a new virtual machine. Install Windows 2000 as the guest operating system, and install VMware Tools, as described in the Workstation 5 User's Manual.
2. Power off the virtual machine. You are about take a snapshot of the virtual machine in its current state: a clean Windows 2000 installation with no Service Packs loaded. Since users will be making clones from the snapshot, you want the snapshot of the machine in a powered-off state: you cannot clone a snapshot that is taken while the virtual machine is powered on. (Of course, a user could always restore the desired snapshot, power off the virtual machine, and then clone it. But by powering off before you take the snapshot, you save your users from going through the extra steps of restoring and powering off the virtual machine every time they want to clone it.)
3. Take a snapshot (**VM > Snapshot > Take Snapshot...**), and give it an appropriate name and description, such as "Win2k Pro Base."



4. Power on the machine, and install SP1. Power off the virtual machine, and take a snapshot, naming it "Win2k Pro SP1."



- 5. Power on the machine, and open the Snapshot Manager (VM > Snapshot > Snapshot Manager).

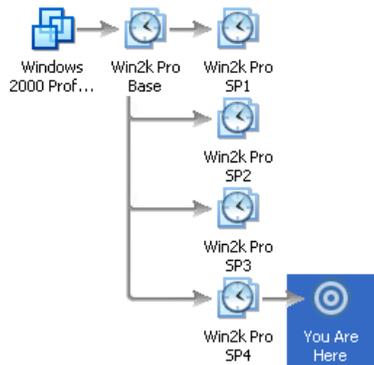


Planning Snapshots for Best Performance

The next step is to load SP2 and take another snapshot. We could do that from the current **You Are Here** position, building the SP2 snapshot on the SP1 snapshot, and then the SP3 snapshot on the SP2 snapshot, and so on, producing a chain of snapshots like this:



However, for this scenario, a linear chain of snapshots is not the best strategy for performance. As the chain of snapshots gets longer, each succeeding snapshot takes longer for Workstation to work with, since it must read all the prior images in the chain to construct the current snapshot. As you proceed farther along the chain, performance can degrade. In some situations, this may be an acceptable tradeoff. If you want to preserve a historical trail of a particular sequence of actions, a chain of snapshots may be necessary. For our current task, however, to improve performance, we want to minimize the number of preceding snapshots that any snapshot is based on. To do this, we need a snapshot tree that looks like this:





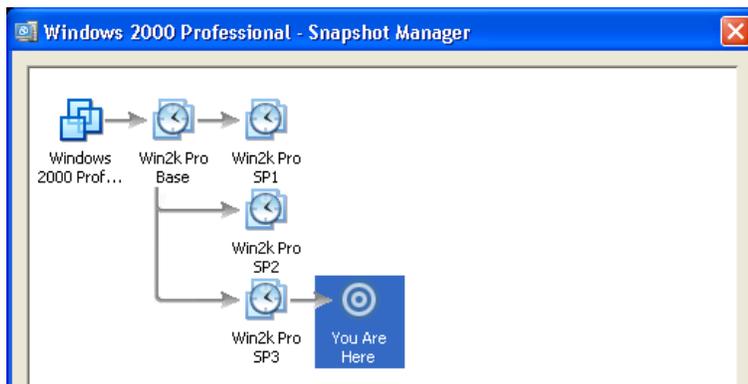
1. Select the Win2k Pro Base snapshot, and select **Go To**. We want to base the next snapshot image, for SP2, on the Base image instead of the SP1 image. Reopen the Snapshot Manager (which closed when you selected **Go To**). Note the **You Are Here** position. Now, changes you make will be based on the Base snapshot.



2. Close the Snapshot Manager, power on the VM, and install SP1 and SP2. Power off the virtual machine, and take a snapshot, naming it "Win2k Pro SP2."



3. Open the Snapshot Manager, select the Win2k Pro Base snapshot, and select **Go To**. Again, we want to base the next snapshot, for SP3, on the Base snapshot, not on the SP2 snapshot, to minimize the length of the snapshot chain and maximize performance.
4. Power on the virtual machine, and install SP1, SP2, and SP3. Power off the virtual machine, and take a snapshot, naming it "Win2k Pro SP3."



5. Continue in this way until you have snapshots of each Service Pack level you want in the repository.



The virtual machine now contains a clean installation of Windows 2000 at each SP level. You could use the same procedure to create virtual machines containing any number of standard development environments, with any of the operating systems supported by Workstation. You could also create virtual machines with applications — various Web browsers, for example — already installed.

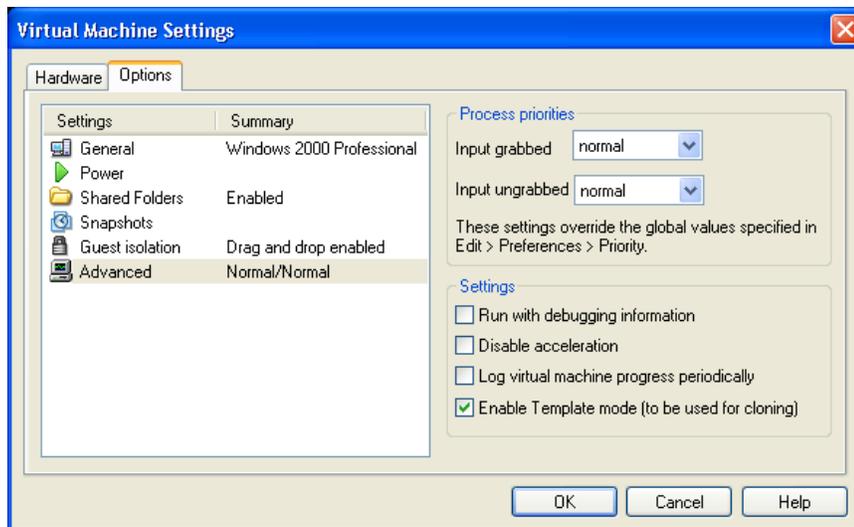
Once you have created a repository of virtual machines, any developer and tester with network access can set up a clean installation of any of the environments stored in the repository in minutes, simply by cloning a virtual machine, as described in [Cloning a Virtual Machine from the Repository](#) on page 7.

Setting the Virtual Machine to Template Mode

Next, set the virtual machine to template mode. This ensures that users will not change the virtual machine. This is especially important because users will often make linked clones from the virtual machine. Since linked clones share and depend upon the virtual disk of the parent virtual machine, it is especially important that virtual machines in the repository are not changed. When the virtual machine is in template mode, it cannot be deleted, and none of the template's snapshots can be deleted.

Note: The system administrator should make sure that users do not have write permissions for the virtual machine's files. This precaution prevents users from disabling template mode for a virtual machine, or inadvertently deleting a virtual machine through the file system.

Set the virtual machine to template mode. Select **VM > Settings**, and select the **Options** tab. Select **Advanced**, and select **Enable Template mode (to be used for cloning)**.

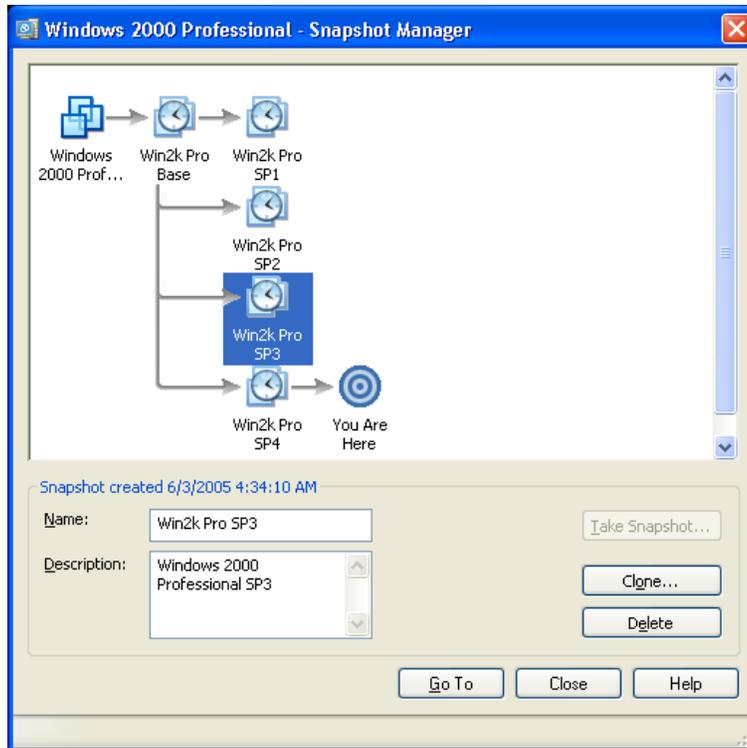




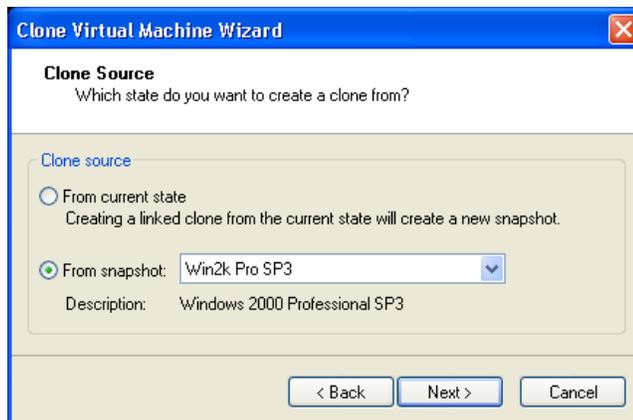
Cloning a Virtual Machine from the Repository

To access and use the repository, open the virtual machine you want, and clone the appropriate snapshot to the local test or development machine:

1. Open the virtual machine with the environment you want. In the Snapshot Manager (Snapshot > Snapshot Manager), select the snapshot with the specific operating system version, service pack level, or applications you want, and select **Clone...**



2. The Clone Virtual Machine Wizard opens with the snapshot you selected as the clone source. Click **Next**.





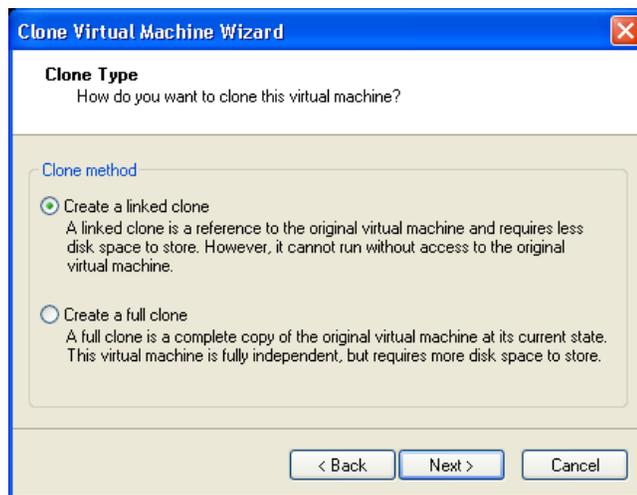
Selecting the Type of Clone: Full or Linked

You can create a full clone or a linked clone.

- A *linked clone* is a copy of a virtual machine that shares virtual disks with the parent virtual machine. This conserves disk space and allows multiple virtual machines to use the same software installation. A linked clone must be able to access the parent virtual machine.
- A *full clone* is an independent copy of a virtual machine, and does not need access to the parent virtual machine. Creating a full clone is an appropriate choice if you want to create a totally independent clone — to use on a laptop, for example. Also, because a full clone accesses its virtual disk locally (instead of across a network, to the parent virtual machine), it may be preferable to a linked clone when performance is more important than conserving disk space.

Follow these steps to create a clone:

1. Select the type of clone you want, and click **Next**.





2. Name the clone, and save it to your local machine. Remember, if the clone is linked, the clone must be able to access the parent virtual machine.



3. Click **Finish**. If you are creating a full clone, the cloning process may take a few minutes.

The virtual machine is ready to use. One thing to remember: the clone you have just made is from a single snapshot of the parent machine. The clone does not retain the multiple snapshots of the parent. Before working with the clone, you should take a snapshot, so you can restore the virtual machine to its clean, just-installed state at any time.