



## VMware Workstation

# Development in Multitier Configurations with Workstation Teams

More and more, developers of software applications and services have come to rely on a multitiered configuration model as the most efficient way to stage and deploy their products. This configuration is by now familiar in its basic and most common form: a system consisting of a database, a server, and some number of clients, all networked, with applications and services provided via a Web browser.

This technical note discusses the challenges of developing software in this environment, and how VMware Workstation addresses those challenges through virtualization. More specifically, this note shows how you can use Workstation 5, with its teams feature, to set up an entire three-tier configuration for development and testing, virtually, on a single physical machine.

- [Advantages of Multitiered Configurations for Software Development on page 1](#)
- [Challenges of Multitiered Configurations for Software Development on page 2](#)
- [Virtualization: VMware Workstation Changes the Landscape on page 3](#)
- [Scenario: Setting Up a Virtual Environment for Multitier Application Development on page 4](#)

## Advantages of Multitiered Configurations for Software Development

With the rapid, and near universal acceptance and deployment of the Internet and Internet-related technology (Ethernet, IP, and browsers), the network linking the components of a multitiered configuration is increasingly likely to be the Internet itself (or an intranet, or some sort of tunnel through the Internet). This environment has encouraged software producers to develop browser-based methods of delivering applications and services, for several reasons.

Using the Internet (or some subset of it) as a delivery/distribution network offers some obvious economies of scale and distribution: customers all over the world have immediate access to a company's products, services, support, and marketing, all of which can be delivered very efficiently over an existing network.

To maximize these advantages, producers have turned to browser-based methods of delivering applications and services. Browser-based applications are easy to deploy — there is no software



to install on the user's machine. The application itself is hosted (and maintained, and upgraded) within the producer's system — all the user needs is a browser and a link.

## Challenges of Multitiered Configurations for Software Development

While this model (a multitiered configuration, 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 even the simplest sort of three-tiered application, developers and testers now need, minimally, access to a database server, a Web server, and a client machine. Until recently, software producers have resorted to a couple of strategies, neither particularly attractive, for providing developers and testers with the hardware they need.

### One Strategy: Development Lab

One 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.

At the same time, maintaining such a lab can become a nightmare for IT. 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.

### Another Strategy: Separate Machines for Each Developer

To avoid the IT costs and production bottlenecks associated with maintaining a development lab of pooled machines, some companies have chosen to provide each developer and tester with a separate hardware setup, with three machines (two servers and a client) 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 Changes the Landscape

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. Individual developers can load, on a single machine, a clean “just-installed” configuration they need, simply by downloading the appropriate virtual machines, and networking them virtually.

### Workstation 5: Teams Feature

To further streamline development for multitier configurations, Workstation 5 introduces the concept of virtual machine teams. Virtual machine teams provide great flexibility — developers can create various multitier configurations by simply adding virtual machines to a team. In addition, teams allow developers to simulate real-world conditions by setting network characteristics such as network bandwidth and packet loss rate on any network segment that connects virtual machines in the team.



## Scenario: Setting Up a Virtual Environment for Multitier Application Development

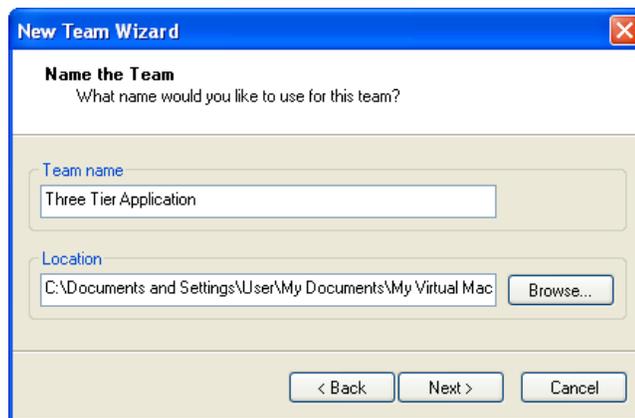
This section gives step-by-step instructions for setting up a basic three-tier configuration on a single Windows XP machine, using Workstation's team feature with three Windows XP SP2 virtual machines created in Workstation. These preconfigured virtual machines can be downloaded from a repository, as described in [Virtualization: VMware Workstation Changes the Landscape on page 3](#). (For detailed information about using Workstation's cloning and snapshot features to create a repository of preconfigured virtual machines for testing and development, see the VMware Technical Note *Using Clones to Streamline Development*.)

For this scenario, we will assume a three-tier configuration with a My SQL database server on Windows XP SP2, an Apache Web server on Windows XP SP2, and various browsers running on different operating systems.

Virtual Machine	Operating System	Application
Database Server	Windows XP SP2	MySQL
Web Server	Windows XP SP2	Apache
Client	Windows XP SP2	IE5, IE6, Mozilla

In Workstation 5:

1. Create a new team: select **File > New > Team**. A wizard guides you through the process.
2. Specify a name and location for the team.





3. Add virtual machines to the team. These are virtual machines that you have already created and loaded with the appropriate operating systems and applications. Select **Yes**, and click **Next**.

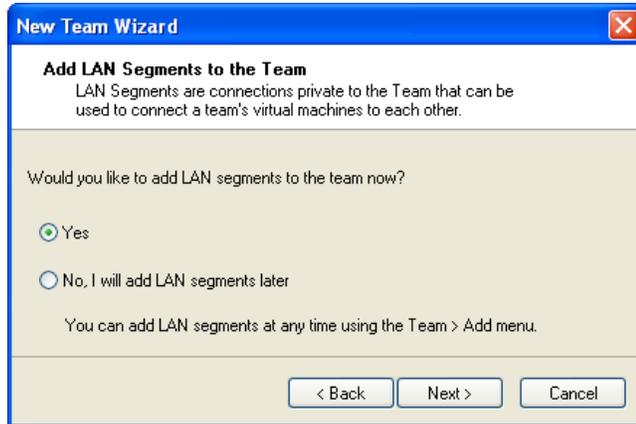


4. Select **Add > Existing Virtual Machine**.
5. Add virtual machines to the team. Browse to add the three Windows XP SP2 virtual machines.





6. Add LAN segments. Select **Yes** and click **Next**.



7. Add two LAN segments, LAN1 and LAN2.



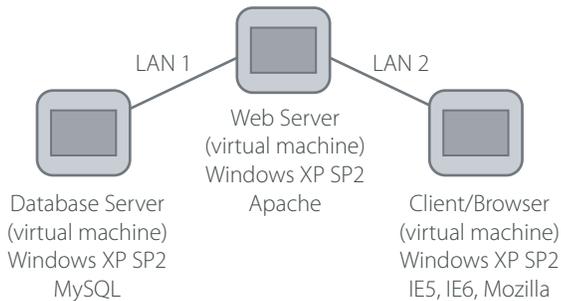


- You can now assign different virtual machines to each of the two LAN segments created above.



You have now created a team consisting of three virtual machines and two private LAN segments.

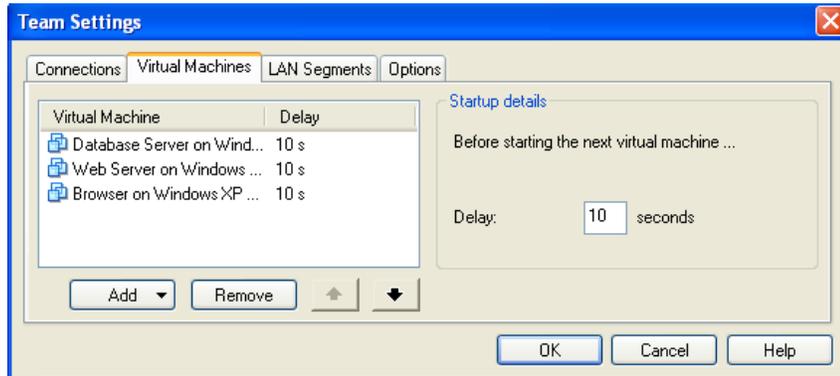
**Note:** The network you have configured in this scenario does not automatically have DHCP or DNS services. You will need to set up DCHP or DNS services on one of the virtual machines in the network, or use static IP addresses.



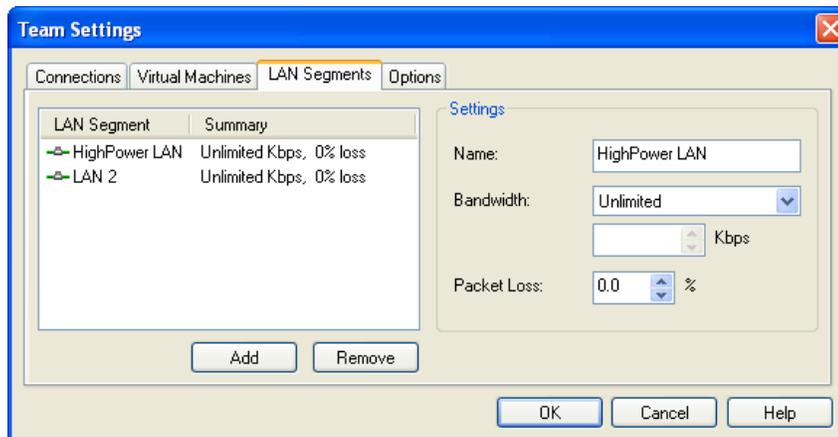
- Before powering the team, configure the startup order and network characteristics of the LAN segments. Select **Team > Settings**.



10. You need to ensure that the database server boots up before the Web server, and that the Web server boots up before the browser that connects to the Web server. In the **Virtual Machines** tab, specify the start up order for the virtual machines, and the delays between booting each virtual machine.



11. In the **LAN Segments** tab, you can specify the network bandwidth and packet loss rate for each LAN segment. This enables you to test application performance at various dial-up and broadband bandwidths, and to simulate adverse network conditions.



12. Select **Team > Power > Power On** to start the team. Notice that the virtual machines are powered on in the order specified. Furthermore, the virtual machines are automatically connected to the specified LAN segments.

**Note:** Once a virtual machine is added to a team, it can only be used within the team.