

# **Deploying Applications to Locked Down Desktops**

*How Application Virtualization speeds deployment, reduces support costs,  
and eliminates regression testing*

*A Thinstall White Paper*



**Thinstall**  
22 Battery Street, Suite 703  
San Francisco, CA 94111  
415.274.2558  
[www.thinstall.com](http://www.thinstall.com)

## Table of Contents

---

Executive Summary.....	3
Locked Down Desktops are Key to Streamlined IT Operations.....	3
Limitations of Locked Down Desktop Environments .....	4
Alternatives to Deploying Applications on Locked-down Desktops.....	5
Application Virtualization.....	6
Thinstall Application Virtualization Platform.....	6
Thinstall Application Virtualization in a Locked-Down Desktop Environment.....	9
Conclusions .....	11
About Thinstall .....	11

## Executive Summary

---

Today's businesses require ever more efficient and cost effective methods to manage the cost of their IT operations. One of the proven ways to streamline IT operations is by standardizing the end user Windows desktop. Generally, this means that application updating and changes to the installed software is limited to IT rather than the end user doing so, so that the desktop configuration is controlled. This enables easier troubleshooting of individual problems, faster delivery of new applications to standard configurations, and helps protect the machine from viruses.

However, a locked down desktop environment can be limiting to an organization that needs to deploy custom business applications or meet disparate needs of the different lines of business. The challenge for IT remains how to keep desktop deployment efficient for operations yet agile enough to meet ever-evolving business software needs?

A new *application virtualization* technology has been introduced which allows custom and commercial software to be easily deployed to the locked down desktop environment. Application virtualization provides unprecedented efficiency in managing application delivery and access while maintaining integrity and control of desktops. With this technology, applications can be updated, centrally managed, and delivered faster.

Additionally, this technology greatly accelerates new software development projects by enabling developers to use the latest runtimes, frameworks, and components rather than being limited by the lowest common denominator mandated by a central IT department. Because applications are isolated and don't modify the file system or registry, they are able to run without risk of damaging the host OS. Users benefit by experiencing the full richness of client applications while administrators experience reduced time and pain associated with resolving application conflicts.

## Locked Down Desktops are Key to Streamlined IT Operations

---

Managing corporate desktops is one of the most important functions of IT yet it is also the most costly in terms of time to maintain, user productivity, and support tools needed. To optimize desktop operations, there are many products that assist with configuration, deployment, and updates for the PC system and software. Best practices have emerged for how to streamline and lower the cost of the PC platform maintenance and clearly locked down desktops is one of the solutions applied across many organizations. Locked down desktops provide many benefits and cost savings:

### **Ease of maintenance**

Single desktop images are ideal for containing costs. This means that IT has to maintain a limited set of applications and configurations. Troubleshooting becomes less onerous with a known configuration; updating becomes easier because all PC's are updated consistently.

### **System Stability**

By testing all changes against a standard, not only can testing variations be kept to a minimum but any changes introduced are assured that they will not affect system performance or stability.

### **Reduced administrative and maintenance costs**

Administration of PC's, either manually or via automated systems introduces costs. The less changes done on a one-off basis means more control, less times a computer has to be 'touched'.

### **Rapid update**

Rapid updates are enabled when desktop standards exist. This is key when changes have to be made for enhancements to applications; there are fewer variations to test against.

### **Security**

Rapid response to security updates is key to limiting the exposure to Windows threats. Another aspect of security is that users can't add software or make system changes because their rights to change the system are limited (if they don't have Administrative control). End users can also be limited as to the data that they can remove from the PC, another important security aspect in many corporate environments.

Current solutions for locked down desktops include 'locking' desktop images with standard software and hardware, but also simplifying application access through the use of Terminal Servers and Citrix. These solutions limit the

desktop to act as a 'terminal' that accesses applications running on a central server. In this case, multiple users can use the PC for application access, but don't have permissions to alter the integrity of a standard machine.

## Limitations of Locked Down Desktop Environments

---

Though locked down desktops provide a range of cost savings benefits, they also can introduce a challenge to IT to quickly respond to the need to deliver business applications. If locked down desktops provide a multitude of benefits, how can applications be deployed more optimally in such an environment? What are some of the inherent challenges presented by a locked down standard?

### Multiple applications developed in varied programming languages and versions

One of the limitations of locked down desktops is that it is difficult to deploy applications that need to be updated often or to deploy a wide range of applications simultaneously which require different, conflicting versions of underlying code components. These factors often impede deployment responsiveness, as well as limit options available to internal developers or prevent use of some external packages.

### Applications often require administrative rights and access to shared directories

Custom and commercial packaged applications often require access to the administrative rights (by a user or a process) on a PC to configure and access system resources. Applications need to write registry keys, write to the file system, and install components such as DLLs to the shared directories of the computer - all of which is not possible on locked down desktops with multiple users and no administrative rights. Microsoft's Application Isolation technology can help address these issues, however only in limited circumstances.

### Timely deployment and the need for multiple application regression testing

Although a desktop may be locked down, multi-application deployment presents complexity into even the most stable locked down environment. Regression testing on sets of applications to ensure they do not cause each other to fail is often required. For example, installing an application that uses the Microsoft .NET Framework version 2.0 requires the .NET Framework version 2.0 to be installed on the desktop first. If .NET 2.0 is not 100% backwards compatible with older version of the .NET Framework then older applications that have been deployed with the .NET Framework 1.0 or 1.1 may fail.

Without full regression testing, it is difficult to verify there will not be any inter-application failures. Applications that use shared components or runtimes are prone to failure as the number of applications installed on a user's PC increase. Especially troublesome are various shared runtimes such as Java, .NET, VB, Crystal Reports, MDAC, and COM/ActiveX controls where installation requires placing files in system global locations and modifying the system registry.

Because of the extensive amount of testing required to roll out new shared components such as .NET 2.0, developers are forced to use the older technologies currently available on the end user desktop. This problem is magnified because developers must decide on their technology choice during the on-set of a new project where projects may take several months or years to complete.

To further complicate multi-application deployment, companies need to respond quickly to even small changes requested by individual users. However fast the development 'enhancement' may be, it may take weeks or months to test against all other applications installed on a user's desktop and then to upgrade all the locked down desktops. Getting around the locked down desktop with case-by-case user installations may result in lost productivity because of the potential of breaking other applications. Moreover, maintenance becomes more difficult when non-standard desktops start to require individual, time consuming support and 'office visits' to fix a unique application problem.

### Deploying multiple versions of the same application

Transitioning from older versions of an application to newer versions is not always instantaneous. Often new versions of an application provide critical functionality that translates to an essential competitive advantage for the organization; however there may be a required migration period where the older version is still required. Data files may need to be migrated, new versions of the application may have dropped essential older features, the newer version may have bugs not present in older versions, or the results of the new application need to be tested against the results of older applications before it is accepted. Most applications are not designed to permit multiple versions to be installed simultaneously which means end users must repeatedly install and uninstall the version they require,

or they must have two desktops (real or virtual). In a managed IT environment, the user will not have permission to install and uninstall older versions of the application so their only option is to use two desktops.

### Deploying applications on Terminal Server

To obtain even greater control over application deployment and to permit immediate remote access from a wide variety of devices, many companies use Microsoft Terminal Server and Citrix MetaFrame to deploy applications. In these environments, it is essential that all users run in Restricted Accounts without access to write file-system components or registry keys to the shared system. Furthermore, many applications are not designed to run in a multi-user environment and attempt to write to shared file system locations or registry keys, making Terminal Server or Citrix deployment impossible. The need to run concurrent versions of the same application on Terminal Server, or run two applications that are incompatible translates into a requirement of additional Terminal / Citrix Servers – which means additional costs.

However cost effective locked down desktops may be, it leads to limitations in the flexibility and speed at which applications are able to be deployed. This also might imply that variations in desktop profiles are needed to accommodate varying business needs of different users groups. So instead of changing the desktop environment, which adds cost, why not *lower the cost of and simplify application deployment* to better accommodate a streamlined, locked down environment?

## Alternatives to Deploying Applications on Locked-down Desktops

The challenge for IT organizations in deploying applications in a locked down – or any environment - is providing a flexible platform to allow rapid software updates while maintaining the highest level of stability and user productivity. Specifically:

- How are applications best deployed for Windows Desktops, Terminal Server, and Citrix MetaFrame in a locked down environment?
- How can multiple versions of an application be in use concurrently?
- How can changes and updates be done as rapidly as possible, without affecting system stability or other applications or requiring weeks of regression testing?

Some solutions have been deployed to make software deployment more cost effective while making application access simpler for the end user. All have advantages and disadvantages, as compared in the Table below.

*Application delivery technology*

	Technology	Advantages	Disadvantages
Traditional client applications	<i>Install and run an application locally</i>	<ul style="list-style-type: none"> <li>▪ Performance</li> <li>▪ Control</li> <li>▪ Able to exploit PC power and functions</li> </ul>	<ul style="list-style-type: none"> <li>▪ User has to be local to machine</li> <li>▪ Changes across an organization can be difficult</li> <li>▪ If machine not configured consistently, maintenance costs can be higher</li> </ul>
Web applications	<i>Web server and browser delivery</i>	<ul style="list-style-type: none"> <li>▪ Easy to update and manage distributed applications</li> <li>▪ Access from anywhere</li> <li>▪ Almost no impact on other applications or software on system</li> </ul>	<ul style="list-style-type: none"> <li>▪ User Interface limitations</li> <li>▪ Lower productivity than interactive applications</li> <li>▪ Lack of offline usage</li> </ul>
VMWare	<i>Emulates instructions and a new machine</i>	<ul style="list-style-type: none"> <li>▪ Easy to deploy new images</li> <li>▪ Full backup and disaster recovery abilities</li> <li>▪ Higher degree of portability than traditional desktops</li> </ul>	<ul style="list-style-type: none"> <li>▪ Additional Windows license cost</li> <li>▪ Additional machines to manage</li> <li>▪ Additional complexity for user</li> <li>▪ Performance hit</li> <li>▪ Large memory footprint</li> </ul>
Terminal Server with Citrix	<i>Run applications on centralized server</i>	<ul style="list-style-type: none"> <li>▪ Applications are centrally managed</li> <li>▪ Flexible license usage</li> <li>▪ Less powerful client machines required</li> <li>▪ Access from any machine</li> </ul>	<ul style="list-style-type: none"> <li>▪ Can't run multiple versions</li> <li>▪ Display performance issues</li> <li>▪ Lack of offline usage</li> <li>▪ High hardware, license, and admin cost</li> </ul>

	Technology	Advantages	Disadvantages
Software as a Service (ASP)	<i>Buy software as a service from vendor</i>	<ul style="list-style-type: none"> <li>▪ Access from anywhere</li> <li>▪ No maintenance or development costs</li> <li>▪ Low entry cost</li> </ul>	<ul style="list-style-type: none"> <li>▪ Security of data</li> <li>▪ Reliability of service</li> <li>▪ Transferability of data</li> <li>▪ Ongoing fees</li> <li>▪ Lack of control and ability to customize</li> </ul>

How an application is delivered is dependent on the profile of the user, the complexity of the application, the level of interactivity needed, and the type of application. For many business critical situations, where maximum worker productivity is essential, desktop applications offer the best and most cost-effective solution by providing a rich level of interactivity, data viewing, and data manipulation. 52% of Enterprises (companies with more than 1,000 employees) and 79% of SMBs provide desktops to more than three quarters of their users.<sup>1</sup> Clearly, applications and desktops running them are the business driven mode of operation.

## Application Virtualization

*Application Virtualization provides the ability to deploy software without modifying the local operating system.*

Application Virtualization works completely in user space, rather than as a device driver, which means it can run on all versions of Windows 95+, without requiring Administrator rights or a system reboot. It works in Guest and Restricted User accounts without any prior installation or permissions required.

Using application virtualization, companies can address the challenges of deploying and accessing software that are described above. Some of the benefits include:

- The ability to deploy custom and licensed software across mixed Windows and locked down corporate desktop environments without installation conflicts, system changes, or impact on stability. This type of deployment makes software delivery and access more fail-proof, easier, and cost effective.
- Shorten the time to deliver application updates because regression testing can be reduced to hours instead of days when testing needs to be done against only a single application.
- Take advantage of new development frameworks sooner (such as .NET) without the need for the user to install the entire framework or to require different versions of the framework for various applications.
- Enables companies to fully utilize their investment in Citrix® MetaFrame® Presentation Server and Microsoft Terminal Server.
  - Allows concurrent execution of applications that were not designed to run multiple instances on the same system. Many applications store configuration data in c:\program files or global registry / INI files. Application virtualization enables selective files and registry keys to be transparently remapped to user-local storage locations such as c:\documents and settings\username\...
  - Allows incompatible applications to be installed and run simultaneously. DLLs, Configuration Files, and Registry values can be isolated on an item-by-item basis for specific applications. This enables multiple applications to be installed and run on a single Citrix® or Terminal Server where previously multiple servers would be required. Each application will have and use private copies of its runtime and configuration files. Software packages can be isolated to eliminate effects on other applications on the same machine.
  - Application Virtualization also solves problems of backward and forward deployment compatibility between multiple versions of the same application.

## Thinstall Application Virtualization Platform

*Thinstall Application Virtualization Platform* represents next generation technology. It has been used by hundreds of independent software developers for several years, running on millions of desktops worldwide. The Thinstall Application Virtualization Platform can be used by companies deploying both custom and commercial applications on locked down and managed corporate desktops. *This section overviews the Thinstall product components.*

<sup>1</sup> TechTarget IT Briefing, SearchWin2000.com, *Back to the Future with Thin Clients*

Thinstall is a virtual machine technology similar to VMWare and VirtualPC, however Thinstall acts at the layer between the application and the OS, rather than emulating instructions and an entirely new machine. For this reason deploying an application with Thinstall does not require additional Windows license procurement. Thinstall runs all application instructions natively, so they do not have any significant performance differences from the original version. Thinstall disk and memory footprints are a fraction of the size of other VM technologies. For example, Thinstall typically consumes 1 MB of RAM, whereas VMWare requires setting aside enough to host another entire copy of Windows and the application to be run (~256MB). Thinstall is packaged directly with the application being deployed to form a single EXE. The executable can be launched by the user and run without any installation. The Thinstall Virtual Machine (VM) occupies no more than 300K of disk space. Traditional VM technologies can require a gigabyte or more of disk space because they duplicate an installation of Windows.

**The Thinstall Thin Virtual Machine (VM)**

Thinstall acts as a *Thin Virtual Machine* that transparently merges a virtual system environment with the real system environment. Thinstall's *Virtual Machine Technology*, with *Thinstall Studio* is the development environment that allows developers to package entire applications into a single file that can be run without an installation process, and without modifying the resident operating system. The **Thinstall Virtual Machine (VM)** performs:

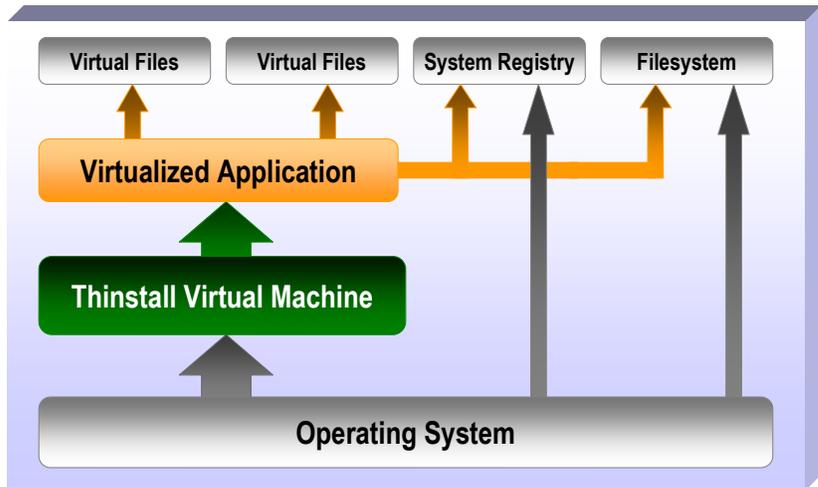
- **Process Loading.** The VM loads a starting EXE file from the Virtual File System and allows it to execute any other EXE directly from the Virtual File System or from the normal file system.
- **DLL Loading.** The VM loads any DLL dependencies the EXE/DLL/OCX files may have directly from archive when instructed. The VM can import and use DLLs located on the normal file system as well as the Virtual File System. The VM provides full control over which DLLs are used, making it possible to entirely eliminate "DLL Hell" problems without relying on partial solutions such as Windows XP "Side-by-side" loading.
- **Thread & Process Management.** The VM is responsible for keeping track of all processes and threads created inside the virtual machine including out-of-process COM and utility applications. The VM manages thread-local storage, notifies DLLs about new process threads, and manages memory for thread stacks.
- The VM runs on all versions of 32bit Windows platforms (95/98/ME/NT/2k/XP/Longhorn) without installation, drivers, reboots, or administrator access.

The *Thinstall Virtual Machine*, only 100-300k in size on disk, consists of 3 core technologies, the [Virtual Operating System \(VOS\)](#), [Virtual File system \(VFS\)](#) , and [Virtual Registry \(VREG\)](#). *Thinstall Studio* packages the VOS, VFS, and VREG along with the application EXE, DLLs, data files, and registry keys into a single compressed EXE. When the application is running, virtual files and registry keys will override system files and registry keys where they might collide, otherwise the application can access any system files and registry keys it normally could.

**The Virtual Operating System (VOS)**

The VOS allows the application to load and run COM, DLL, and EXE files directly from the created package file, bypassing Windows normal loading procedures for DLL loading and COM object creation.

Thinstall's Virtual Machine VM is responsible for loading virtualized applications into memory and managing the virtualized file system and registry.



**The Virtual File System (VFS)**

The VFS is a compressed file system that is transparently merged with the real file system at runtime. The VFS remains embedded in the initial EXE distribution package and never extracts to disk. Individual package files are decompressed into memory in small blocks as required by the application. The Virtual File System is only visible to applications running under the Virtual Machine. At runtime the application can access both virtual files and real system files without needing to know the difference between the two.

To the Thinstall packaged application, a virtual file is indistinguishable from a normal file; however it does not actually exist on the hard drive. Thinstall makes it appear as though all virtual files have been extracted to the hard drive. Because virtual and regular files may simultaneously exist at the same path location, Thinstall allows the creator to control which version will be used when a file is read, or where the data will be placed when data is written to a file.

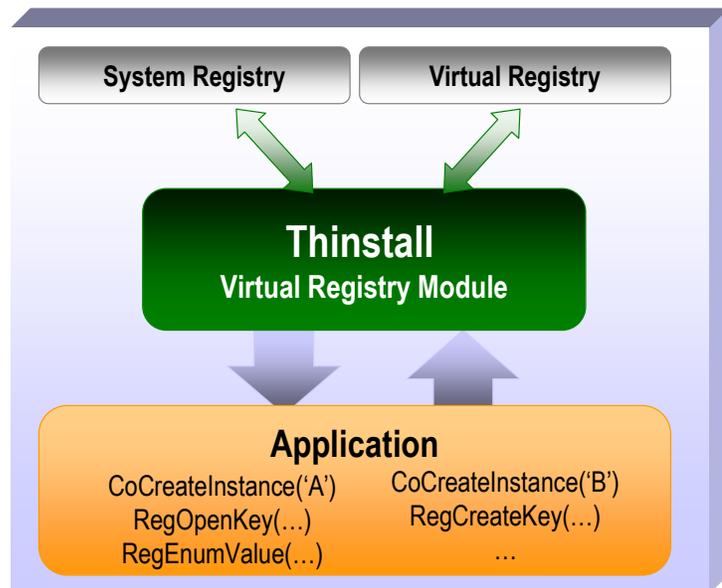
The Virtual File System (VFS) presents a "merged" view of package archive files and system files:

- VFS is always compressed on disk, meaning the installed disk footprint is the same as the pre-install footprint.
- VFS provides transparent streaming decompression for files accessed inside the VFS and only decompresses data on-demand.
- VFS provides transparent encryption/decryption capabilities for file writes/reads inside the VFS.
- VFS allows all processes and libraries loaded through the VM to access files from both the VFS and the normal file system.
- The VFS presents a "merged view" of the file system to applications run by the VM. Files from both systems can exist at the same directory.
- The VFS works with any underlying file system, including FAT32, NTFS, Network shares, and (by design) any future file system.

### The Virtual Registry (VREG)

The VREG transparently merges virtual registry keys with the real system registry. The VREG makes it appear as though the prepackaged registry keys have been installed into the system registry without any real system change needed. To make virtual registry creation easier, Thinstall has COM recording abilities that allow the creator to automatically gather all the registry keys needed by a COM component before deploying it.

Thinstall's Virtual Registry Module makes virtualized Registry Keys appear to exist for all applications and DLLs loaded by Thinstall. Virtual registry keys are overlaid onto the system registry so the application can use entries from either system. The virtual registry is consulted first for a new key, if no matches are found, the system registry is used.



Thinstall's Virtual Registry system exists as a separate module that is linked with the application when one or more registry script files exist in the application. Registry script files are simple text files that describe which keys to super-impose on the real system registry.

#### Example: Virtual Registry for COM/ActiveX

The Virtual Registry presents predefined registry keys to all applications and libraries loaded through the VOS.

- Virtual registry allows COM/ActiveX based applications to run on systems where registry access has been restricted.
- COM and ActiveX controls can be loaded directly from the VFS, eliminating all possibility of removal by another application's uninstall.
- Virtual Registry exists inside the EXE, so it cannot be mis-configured by another program's installer or uninstaller.
- Registry recording system allows recording of all registry keys a COM/ActiveX control would normally create during registration "Regsvr32". Recording occurs on the development computer, and is simulated in the runtime environment.
- The VREG system automatically tracks COM object lifetimes so that it can free DLL and EXE servers.

### Development Environment: Thinstall Studio

Thinstall packages an application and all of the files it uses into a single executable that can be run without installation. Thinstall Studio consists of:

- GUI which can be used to add virtual files and virtual registry scripts
- Registry recording tools to automatically determine required registry keys for COM/ActiveX controls
- Command-line packager that can be used to automate package creation
- Runtime logging component which is used to display application activity include all Win32 API calls, the ability to highlight error conditions, obtains a list of files and registry keys used, and troubleshoot 3<sup>rd</sup> party applications and DLLs without source code.

## Thinstall Application Virtualization in a Locked-Down Desktop Environment

---

Thinstall Application Virtualization is ideal for deploying software to locked-down desktops. Typical scenarios and advantages are described below.

### Using Thinstall to deploy to locked down machines

Thinstall enables applications to run without installation or system changes on Guest and Restricted User accounts. The application is 'packaged' using Thinstall so that the application, the virtual registry and file system, and the Thinstall Virtual Machine are a single EXE. The EXE is copied to the user's computer via network, CDROM, flash device, or Internet download. The end user can run the EXE directly, and has no way of knowing that they are running a virtualized application. The Thinstall VM loads the application and makes it appear as though all the required system changes have already occurred. The application can create COM and ActiveX controls that are registered in the registry as well as load virtual files from locations where it does not have read/write permission.

### Using Thinstall to eliminate regression testing on software sets

Thinstall eliminates or substantially reduces set regression testing by allowing applications to be deployed in an 'isolated' state. When deployed in an isolated state, each application loads all of its configuration files, DLLs, and registry keys from the virtual machine. The possibility of interference from other applications can be eliminated or minimized. For examples, applications that use the Java or .NET runtimes will have the exact version of the runtime they were tested with embedded in the package. As the application runs, it will always load it's version from the package even if the desktop has a different version installed or no version at all.

### Using Thinstall for rapid updates

By eliminating the need for regression set testing, Thinstall can accelerate the application iteration / testing / deployment cycle. Testing can be moved from a per set basis, to a per application basis. Deployment of one application can be decoupled from deployment of other applications. Because applications no longer modify the end-user's registries or file systems, and operate on a semi-isolated environment, the need for set testing can be eliminated for most applications.

If the process is stream-lined using automated testing, a new release could potentially be delivered to users in less than an hour.

*Rapid updates are enabled with application virtualization*

	Typical Deployment	Application Virtualization Deployment
User requests change	5 minutes	5 minutes
Change design	30 minutes	30 minutes
Developer implements change	25 minutes	25 minutes
Rebuild virtual application package	n/a	1 minute
Single application regression testing (with automated testing in place)	n/a	<b>25 minutes</b>
Regression testing multiple applications application	Days to Month	<b>n/a</b>
Application pushed to user	5 min	5 min
<b>TOTAL time for software update</b>	<b>At least several days</b>	<b>About 1.5 hours</b>

**Using Thinstall to deploy multiple versions of the same application simultaneously**

Thinstall allows multiple applications to be installed and running at the same time by isolating registry and file system changes to individual packages. Most applications cannot be installed simultaneously for one of the following reasons:

- **Applications store configuration information in the registry or .ini files.** Thinstall’s virtual registry eliminates the need to modify the system registry, each application instance will see a different instance of the registry at runtime.
- **Applications store program data in global hard-coded locations.** Thinstall’s virtual file system makes DLLs and data files appear to exist at global hard-coded locations without the need for modifying the real file system.
- **Applications require different versions of incompatible runtimes.** Thinstall’s permits runtimes such as Java and .NET to be isolated and packaged with an application. Each running application will use the runtime packaged with the application without requiring the runtimes to be installed.

**Thinstall converts normal applications to Terminal Server applications**

The primary reasons normal applications fail to run in multi-user environments such as Terminal Server or Citrix server include:

- **Attempts by application to write to the registry.** Thinstall virtual registry allows applications to write to global locations. Changes made by the application will be visible only to that particular application.
- **Storing user-specific data in global locations on the file system.** Thinstall is able to redirect file writes from global locations to user specific locations. For example, when an application tries to write data to c:\program files\myapplication\data.dat, the file operations will be transparently redirected to c:\documents and settings\USERNAME\Application Data\myapplication\data.dat
- **Reliance on runtime component requirements that have the above problems.** Using Thinstall, it’s possible to bind all runtime components directly with the EXE. When the application loads or uses the runtime components bound into the package, the runtime components will be presented with the same virtualized environment the users sees.

## Conclusions

---

An *Application Virtualization Platform* enables easier, more cost effective deployment of custom and commercial applications to locked down desktops and Citrix/Terminal Servers. It enables faster deployment, eliminates the need for multi-application regression testing, and allows several versions of the same application to be simultaneously installed. This results in substantial time and development cost savings, as well as virtually instantaneous response to business user's needs for continual application updates. The typical conflicts of installing and running software that requires registry key, file system and DLL modification can be eliminated by deploying software with a thin virtual machine that runs between the OS and the application.

## About Thinstall

---

Thinstall is the next generation of application deployment. Thinstall's *Application Virtualization Platform* virtualizes software delivery and access, enabling Software Companies and Enterprises to deploy software without modifying the local operating system. Companies can deploy custom and licensed software across mixed Windows and locked-down corporate desktop environments without installation conflicts, system changes, or impact on stability. Software updates are easier and faster because regression testing can be reduced to days instead of weeks while installation failures and the associated help desk costs are eliminated.

Applications virtualized with Thinstall can be accessed and updated over an existing delivery infrastructure such as internal company networks (LAN, WAN, Intranet), the Internet, Web servers, CD-ROM, and Flash devices. The applications can be run on-demand on locked down desktops, off line laptops, Terminal Servers, and Citrix environments.

Founded in 1999, Thinstall has deployed thousands of applications to millions of desktops around the world. Customers include GE, the US Military, Boeing, Qualcomm, Lucent, Fujifilm, Northrop Grumman, Morgan Stanley, and Toshiba. Thinstall is privately held with company headquarters in San Francisco, CA.

[www.thinstall.com](http://www.thinstall.com)

[info@thinstall.com](mailto:info@thinstall.com)