

SoftGrid® v4: Under the Covers

Exploring the SoftGrid Platform

October 2006

Contents

Introduction	1
What is Application Virtualization?	2
What Does Application Virtualization do for you?	3
Virtual applications, not virtual machines (or operating systems)	3
How do applications get to the desktop?	4
Icon delivery and file type associations	5
Beyond Terminal Services all the way to the desktop	6
Complement to Electronic Software Distribution	6
What SoftGrid Does and Does Not Do	7
Internet Explorer and Java Virtual Machines	7
Print drivers	7
OS updates	8
Antivirus software	8
Mobile mode	8
What Do You Need to Make This Work?	9
How the Components Work and What You Need to Deploy Them	11
SoftGrid Server	11
SoftGrid Data Store	11
SoftGrid Management Console	12
SoftGrid Client	12
The Art and Science of Sequencing	14
What is sequencing?	14
How does the Sequencer work?	14
What do I get when I'm done sequencing?	15
Fine-tuning	16
Creating and following recipes	17
The real "time-to-deploy"	17
License control	18
Real-life Deployment Considerations When Using SoftGrid	19
.SFT storage	19
Distributed environment	19
Applications with hard-coded paths	19
SoftGrid for Web Applications	19
Windows User Profile storage	20
The Kinds of Problems Being Solved by SoftGrid	21
Application conflicts	21
Reducing the number of silos in a Terminal Services environment	21
Regression testing	22
Multi-user application problems	23
Intangible benefits explained	23
What's New in SoftGrid v4	25
Virtualization	25
Streaming	26
Sequencing/Packaging	26
Management	27
Application-Level Security	27
About Microsoft® SoftGrid®	28
Conclusion	28

Copyright

©2006 Microsoft Corp. All rights reserved.

This material is protected by the copyright laws of the United States and other countries, and is the property of Microsoft Corp. It may not be reproduced, distributed, or altered in any fashion by any entity (either internal or external to Microsoft), except in accordance with applicable agreements, contracts or licensing, without the express written consent Microsoft. Microsoft shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof.

The information in this publication is subject to change without notice.

Introduction

As SoftGrid® becomes better known, we have found people want a deeper technical discussion than they might find in a simple brochure or PowerPoint presentation. So we've decided to share our experiences, and our customers', by creating a document that dives into SoftGrid's technical aspects and answers many of the most frequently asked questions about it.

SoftGrid was developed to solve very specific enterprise problems. Aside from appreciating the technical "coolness" of the product, let's consider the business reasons why the product is being so widely adopted:

- Reduces the cost of administering applications through centralization of management, provisioning, support, and deployment
- Dramatically decreases the risks, cost, and time to deploy applications to desktops, laptops and servers
- Eliminates application conflicts, allowing more applications to run on multiple platforms
- Reduces regression testing since application conflicts are eliminated
- Reduces the number of servers in and complexity of a Terminal Services or Citrix farm
- True on-demand availability of applications, without local installation, reducing administration time and cost
- Preserves existing software investment, turning applications into network-enabled utilities
- Ensures license compliance, reduces software piracy, and helps to better manage software assets

Achieving these benefits for your IT organization is a lofty goal, and if you're like many of our customers, you'll have questions similar to the following:

- How does it all work?
- Where do the applications run?
- How do you "package" an application to run with SoftGrid?
- Why does it allow non-TS-friendly apps to run on Terminal Services?
- Is it only good for terminal servers and Citrix?
- Isn't this like machine virtualization?
- What type of hardware do I need to run a SoftGrid Server?
- Doesn't this become a single point of failure?
- How do you "virtualize" an application and still have it execute locally?
- Does it help with distributing operating system service packs and hotfixes?
- Can I virtualize Internet Explorer?
- What *won't* SoftGrid do?

Any of these sound familiar? In the following sections, we'll guide you through a high-level overview of the SoftGrid v4 architecture, cruise through some of the basic (and not-so-basic) concepts behind the technology, and then dive into how it all works and what you need to accomplish to have a functioning system.

What is Application Virtualization?

The idea behind application virtualization is a simple one: *you run an application on your desktop without ever having installed it*. Unlike using Terminal Services, this application executes locally, using local resources (e.g., processor, memory, disk, and network card). In other words, the application runs, saves data, prints, and acts as if it's installed locally even though it is not.

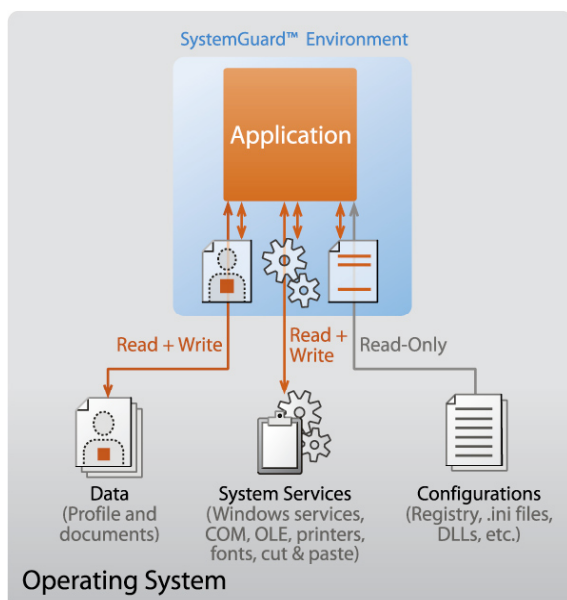
Besides this, you can run multiple versions of the same application on your desktop without conflict—again, with all the applications executing locally – and not as “screen scrapes” from a remote terminal server.

Let's clarify things a bit with a simple visualization. The image below shows the basic concepts of application virtualization—the application executes on the local machine using its resources, but is not allowed to modify anything. Instead, it runs in a small virtual environment that contains the registry entries, files, COM objects, and other components that it needs to execute. This virtual environment acts as a layer between the application and the OS. The virtual layer is very “light” (generally only a couple megabytes of memory) and loads just prior to the application loading.

This is a high-level look at how the application executes. Obviously, there is much more to this product, but the basics are always the same:

- The application is *not* installed on the local OS or file system.
- The application *does* execute locally using local resources just like a locally installed application.
- A virtual environment is loaded for each application.
- The virtual environment isolates the application from the operating system, protecting the OS from application changes and other applications.
- Even though the application is not installed, it can interact with the local OS the same way that a locally application does, making direct use of all the system's peripherals (e.g., local disks, USB devices, network printers, etc.).
- Only the required pieces of code go onto the client computer (unless a decision is made to pre-cache applications).

Later in this paper, we'll talk about how applications are sequenced, delivered to the desktop, and modified. We'll also discuss the components that are required to make this system work and make it highly available.



With SoftGrid's application virtualization, an application cannot write to the registry or system files outside its virtualization environment. This protects the Operating System from changes normally made by applications.

Applications do have the ability to read information from the local system. This information may include local registry settings or files, but the application cannot modify either of these. If it attempts to modify the OS, the modification is caught in the virtual environment and stored there.

What Does Application Virtualization do for you?

From a technology standpoint, the idea of virtualizing an application is certainly intriguing. But the real test for the long-term success of any technology is: *"Will this benefit me or improve my existing environment? If so, how?"*

SoftGrid works by solving a particular set of problems for an organization. How it benefits you and improves your environment depends on the answers to a few questions:

- Do you currently suffer from any of the following application deployment challenges:
 - Intolerably long time to deploy applications to the desktop
 - Complexity of application packaging
 - Lack of a means to monitor application license compliance
 - Frequently changing applications
- Do you have applications that conflict with one another but must be run simultaneously?
- Is your helpdesk staff required to "fix" applications on a regular basis due to user changes or modifications by other programs?
- Does your organization spend a large amount of time regression testing new applications or application updates?
- Is there a need to run multiple versions of the same application (such as Microsoft Access or other Office components)?
- Do you have a terminal server or Citrix environment with multiple application silos as a result of application configurations or conflicts?
- Do you have any applications that just *will not* run in a terminal server environment?

If you are not facing *any* of these issues, then SoftGrid may be a product that would not significantly benefit you or your environment. As such, we can assume that your environment has the following attributes:

- Small/limited application portfolio
- No application conflicts
- Limited application changes or updates
- No need to monitor for license compliance
- No need for application roll-back after an installation
- Applications are not packaged, or packaging is a simple process
- No terminal servers or Citrix, or only a very small terminal server environment with limited applications and no application silos.

If that describes your environment, then consider yourself fortunate! If not, then the SoftGrid technology may be your best bet when it comes to virtualizing your applications. But what exactly is this?

Virtual applications, not virtual machines (or operating systems)

A lot of people tend to make comparisons between SoftGrid and virtual machines (such as Microsoft Virtual Server). To address this, we should discuss what a virtual machine is and what it is not.

A virtual machine is a great tool for virtualizing the machine on which you are installing an operating system (and applications). Think of it this way: a virtual machine provides an abstraction layer between your hardware and the operating system that's running on top of it. It also allows you to manage and simultaneously operate multiple environments on a single machine. The power of machine virtualization is that you can consolidate servers and PCs, sharing under-utilized hardware resources and reducing the total cost of ownership when it comes to

MYTH: SoftGrid is just like a virtual machine.

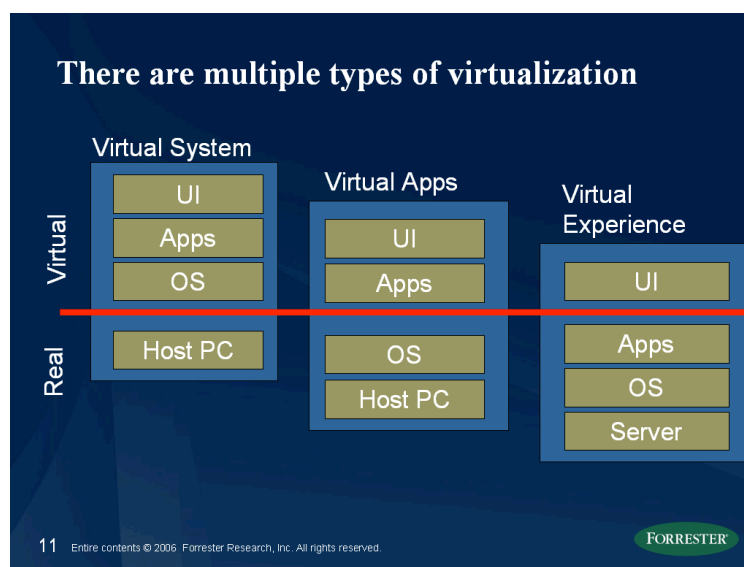
FACT: SoftGrid and machine virtualization play in two distinct arenas. Virtual machines virtualize a hardware platform/operating system. SoftGrid virtualizes and executes applications within a native OS.

hardware purchasing, monitoring, management and maintenance.

SoftGrid takes this concept and moves it up the logical stack. In fact, it is complementary to virtual machines – many customers use both. The abstraction layer created by SoftGrid lies between the operating system and the applications that run within it. By *virtualizing* all the aspects of an application, you don't affect the operating system or other applications running on that machine.

SoftGrid changes how the operating system perceives applications whose very installation and execution can reduce the stability of the operating system and other applications. The power of SoftGrid is that it allows applications to be delivered dynamically as services that can be added or removed without leaving a trail on the client system. This in turn reduces the total cost of deploying and maintaining applications and systems.

This diagram, from Forrester Research, illustrates the difference between virtual machines, virtual applications and virtual interfaces/experiences (Terminal Services, Citrix):



Source: *Desktop Virtualization Is The Future Of The Corporate PC*, Forrester Research, Inc., January 2006

How do applications get to the desktop?

Now that we know where the applications run and that there are real benefits to this technology, the question becomes: *"How does the application package get to the desktop if the package isn't installed?"*

To state it simply, we could say that it runs out of a cache on the local machine. This cache is populated from a SoftGrid Server that maintains the application packages. Once the application is cached, the SoftGrid Client virtualizes the file system, registry, and other application elements and loads the application out of cache and into the virtual environment.

Of course, there's still the question of how the application gets into the cache in the first place. To visualize this, we've created a diagram (below) that shows the paths the data traverses prior to an application launch.

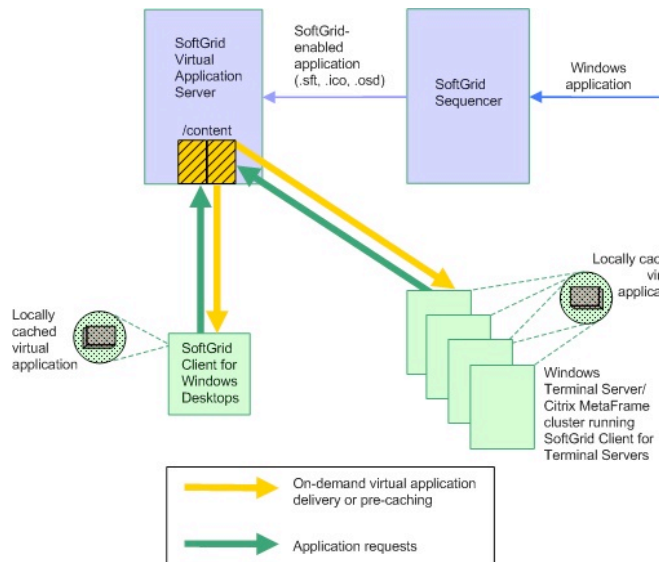
There are several components in this system. The two components we're concerned with are the SoftGrid Server and the SoftGrid Client.

MYTH: SoftGrid streams the application to the client each time the application is launched, taxing network bandwidth and slowing launch time.

FACT: SoftGrid streams only about 10-30% of the application to the client the FIRST time they run the application. Subsequent launches come from cache.

The SoftGrid Server maintains the application packages and streams them to the client's cache. The word "stream" means that this is not a flat file transfer of the package. Initially, only portions of the application are sent to the client so that the user can launch the application. Once the application is launched, the remaining segments of the package are sent to the cache as the user runs the application and accesses functions that are not already cached. The transfer of the package is done via an RTP stream between the client and the server.

The on-demand delivery of applications in the SoftGrid system is a very efficient way to deploy and manage applications. The first parts of the package that are sent to the client are the minimum pieces the application requires to launch (about 10-30% of the application on average). In this way, the application executes without the user having to wait for the entire application package to be transferred to the client.



Once these pieces are in place (generally 5-15 seconds on first launch) the application will open and be ready for use. Subsequent launches will not require these components to be re-streamed, because they are now cached on the client, therefore launch is instantaneous.

Before we go further, we should mention that it is possible to automatically pre-cache an entire application on a user's machine without having to go through the process of streaming the first time.

The SoftGrid system is permission-based and uses Active Directory (and other directory services) as its account authority. This means that every time a user tries to run an application (whether for the 1st or 101st time), the system will check AD to ensure that the user is authorized. In addition, the

system can be configured to perform audit-based license tracking or strict license enforcement. For example, an administrator can assign a concurrent license to an application so that no more than 20 users can run the application simultaneously. In this case, a 21st concurrent user will receive a message stating there are no more licenses available.

Icon delivery and file type associations

Now that we have a basic process for delivering the application (or its package) to the user, we need to determine which applications will be available to those users so we can deliver the application icons. SoftGrid does this with two components. The first is Active Directory. SoftGrid has the ability to use the user and group objects within AD as authentication objects. This simplifies administration and allows administrators to use existing authentication mechanisms to deploy applications.

The second component is the SoftGrid Data Store. This database stores the application configurations within SoftGrid and supports both MSDE (for small implementations) and Microsoft SQL (for large or fault tolerant implementations). The database itself is very light, using about 1 MB of space for 100 applications.

With the SoftGrid Data Store and Active Directory support in place, icons (and therefore applications) are delivered to users based on their normal authentication. Once a user logs into their workstation, their Windows credentials are passed to the SoftGrid Server. The SoftGrid Server processes the login against the applications in the SoftGrid Data Store for a list of authorized applications. These applications (or icons) are then placed on the user's Start Menu, Quick Launch Bar or desktop.

The SoftGrid system can also change file type associations on a Windows system on-the-fly. For example, when publishing Microsoft Visio through SoftGrid, the administrator has the option to configure the package with the .VSD extension. If this is done, then when a user authorized to run Visio logs into their Windows session, the .VSD extension will automatically be registered by the SoftGrid client and will be associated with the SoftGrid-enabled copy of Visio. When a user clicks on a .VSD file, the SoftGrid-enabled copy of Visio will be launched.

Beyond Terminal Services all the way to the desktop

Because SoftGrid removes application conflicts and allows non-multi-user applications to run under Terminal Services, it initially gained wide acceptance in the Terminal Services and Citrix communities. SoftGrid has subsequently maintained its reputation for providing significant benefits to a Terminal Services or Citrix environment. The gist of why this is should be apparent by now: many of the problems that are present in Terminal Services and Citrix—application conflicts requiring server siloing, applications that are incapable of running in multiple sessions or with multiple users, complexity, difficulty in management and regression testing—are alleviated with SoftGrid. Any company that invests in server-based computing and knows firsthand how much these issues cost in terms of time and resources should appreciate the value that the SoftGrid product has to offer.

MYTH: SoftGrid makes it unnecessary to use Citrix Presentation Server.

FACT: In certain circumstances, this is true. However, many customers use SoftGrid to not only streamline their application management on the desktop, but also to enhance the capabilities and manageability of their Citrix environments and to extend the concept of server-based computing.

In this sense, it is certainly worth taking a look at SoftGrid for your Terminal Services and/or Citrix environment. But before you do, consider another one of SoftGrid's strengths: you can redeploy your Terminal Services and Citrix application packages in exactly the same manner to your desktops, from the same management console, merely by adding the SoftGrid Client.

You may be thinking you'll have to overhaul your entire network to the point where all PCs are stripped-down systems with nothing but the SoftGrid Client and an operating system. With SoftGrid, you don't need to do that. You can deploy SoftGrid and deliver applications to your existing desktops with few changes or back-end preparation. SoftGrid applications run alongside locally installed applications, allowing you to deploy applications without having to uproot your existing infrastructure.

Additionally, many key applications require too much overhead to deploy using a TS/Citrix server farm or mid-range servers. In this case, it may make sense to leverage the distributed processing power of your users' PCs. You need to be the judge of your overall environment, but don't underestimate SoftGrid's power as a strategic and cost effective tool for achieving the right balance of processor/bandwidth utilization for your organization.

Complement to Electronic Software Distribution

By now you might be thinking that that SoftGrid's strengths sound similar in concept to Electronic Software Distribution (ESD) solutions like Microsoft SMS or Novell ZENworks. If you *are* thinking this, you're right in some respects. ESD is designed to manage desktops and terminal servers, from operating system patches and configuration to deploying the applications themselves.

Where ESD fails is critical: *application conflicts*. This is the same area that plagues application vendors across the board. It has been estimated that 30-40% of all Windows applications conflict with other applications. It is also estimated (conservatively) that 10-15% of ESD software installations fail when deployed.

This is where SoftGrid excels. And if you were to use SoftGrid as a complement to your ESD tool, you could "have your cake and eat it too." SoftGrid manages and distributes troublesome applications while you leverage your ESD solution to handle the management and configuration of the operating system (e.g., delivering OS security patches, deploying bare metal images, etc.)

SoftGrid now has a version that has been specifically designed to work within Microsoft SMS. SoftGrid for Microsoft SMS allows you to virtualize applications and either deploy them on-demand using SoftGrid's streaming technology, or push them using SMS' delivery mechanism. There is also a version call Virtual Extensions (VX) that just lets you virtualize the applications.

What SoftGrid Does and Does Not Do

SoftGrid specializes in application deployment and execution, not in deploying or virtualizing the operating system or OS changes. With that in mind, take a look at the following "rule of thumb" list for what SoftGrid will and will not do.

What it does	What it doesn't do
Applications	Internet Explorer
Fonts	Print drivers
Application registry settings	OS Hotfixes
Application file system changes	Antivirus software
Services	
Runtime objects	
MDAC versions	
Java virtual machines	
Common program files	
Database drivers	
Mobile mode	

We will address each one of these individually, but if you take a look at the items that SoftGrid does not do, they all share a common thread: they are all tightly interwoven with the operating system. Certain application components like MDAC or shared program files seem to fit into the OS category; but really they do not. What we're talking about here are OS components and software tied to the OS versus applications that install and run on a PC.

Internet Explorer and Java Virtual Machines

Internet Explorer is tied in very closely to the OS—too closely to be virtualized within SoftGrid.

SoftGrid does help with the Java Virtual Machine. Oftentimes users will be required to run Web applications that have different JVM version requirements. This creates a conflict, because only one JVM can be associated with IE at a time. With SoftGrid, you can virtualize the JVMs and launch IE using any JVM you choose. As a bonus, the different JVMs can even be running simultaneously

Print drivers

Print drivers cannot be virtualized using SoftGrid. The basic requirement of a print driver (or any driver for that matter) is that it needs to be seen by the OS, and by any application running within the OS. By virtualizing and isolating a driver, you override this requirement and defeat the purpose of the driver itself. Future versions of SoftGrid may make it possible to virtualize a driver within an application's virtual environment; but this is not possible as of this writing. Does this mean that if I have a program that installs a print driver, I can't virtualize the application and use it with SoftGrid? Not at all. You can still virtualize these applications. The only difference is that you must install the print driver component locally on the device, separate from the SoftGrid package. (You can do this via ESD, script, script at application launch, or manually.). The application then uses the driver just as it normally would.

It's worth noting that now, with SoftGrid v4, you can now virtualize Windows Services, which was not possible with earlier versions.

OS updates

Because SoftGrid is specifically designed to abstract the applications it delivers from the base operating system, it won't deliver updates to the OS. Because OS updates and service packs modify boot and required system files even before a GUI loads, SoftGrid is not able to virtualize these updates. You should use another mechanism (such as ESD) to update the OS.

Antivirus software

Most antivirus software packages are closely tied to the OS and include services that load at boot time. Because of this, SoftGrid does not virtualize antivirus software.

Mobile mode

Streaming the pieces of an application in sequence works well for PCs and servers that are on a high-speed local area network; but that doesn't help when a salesperson needs to edit a proposal on the airplane.

Using its "disconnected operation" (DO) feature, SoftGrid can completely cache all elements of each application so that nothing more is required from the SoftGrid Server. An administrator can tell the SoftGrid Client on the machine to allow a user to have access to a certain application for, say, 30 days, in order to ensure that IT administrators maintain control over licenses and versions of applications. You can detach for up to 9999 days in the current version of SoftGrid. Future versions will allow more granular and extended control of taking applications offline.

Operating in mobile mode does not violate SoftGrid's authorization mechanism. A user must be authorized against the SoftGrid server before they can cache their applications and disconnect from the network. In order to continue running applications in a disconnected state, the users will have to reconnect to the SoftGrid server before their timeout period expires in order to reset the disconnected operation timer. This can be done via a VPN over a low bandwidth connection.

High availability is another benefit of the DO feature. Should a network outage or other trouble interrupt the connection between the terminal servers or PCs and the SoftGrid Servers, users will still be able to run applications.

Disconnected mode works on a per-user and per-application basis.

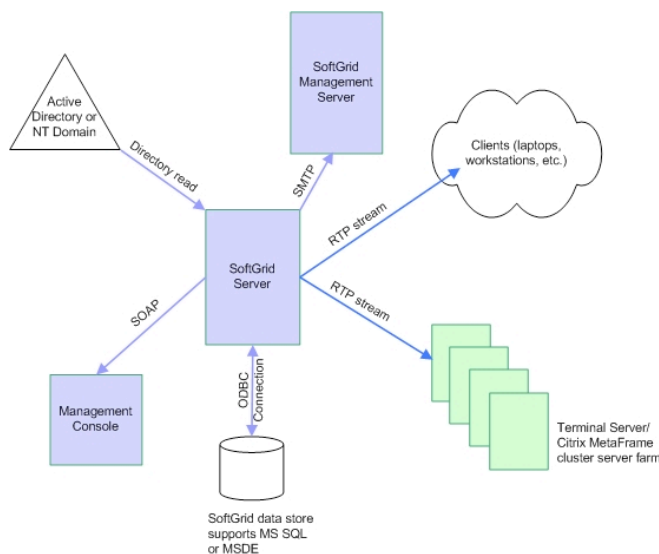
What Do You Need to Make This Work?

The basic components required for SoftGrid v4 are fairly simple, regardless of how large or small your environment is. Like anything else, though, as the environment scales and fault tolerance becomes a requirement (rather than a “nice-to-have”), the design *can* become more complex.

To get a basic system up and running requires seven components:

- SoftGrid Sequencer (we’ll discuss this component in The Art of Sequencing section)
- SoftGrid Server to maintain the packages
- SoftGrid Data Store to maintain application information
- SoftGrid Management Console
- SoftGrid Management Server
- An authoritative source for accounts (Active Directory or NT domain)
- SoftGrid Clients

The diagram below shows the basic communication paths and components in a SoftGrid environment. We’ll explain in more detail the components that are required for any SoftGrid implementation.



The first component you should look at is the SoftGrid Server itself. This server will run the SoftGrid Server services, host the applications packages and, in some cases, host the SoftGrid Data Store. Here’s a good rule of thumb for server sizing: a dual processor server with 2 GB of RAM can support 1,000 concurrent users or 40 terminal servers. The actual server requirements are outlined later in this paper.

The second item we’ll draw attention to is the SoftGrid Data Store. This database supports both MSDE and MS SQL, and maintains the SoftGrid application configurations. If you require a high availability solution, you should consider investing in a clustered or replicated SQL data store. You can use MSDE in small environments (where it is generally installed directly on the SoftGrid Server itself) to save on hardware and reduce implementation costs.

A quick note about the layers of security within SoftGrid:

Authentication: Users cannot launch an application without the SoftGrid Server confirming access privileges unless in Mobile Mode (which requires pre-authentication before the user goes offline).

Server: Essential application elements (those that are used for authentication) always reside on the SoftGrid Server, preventing deactivated users from launching applications.

Transport: SoftGrid employs Transport Layer Security (TLS) and other encrypted tunnel methods to prevent session hijacking.

Client: Cached applications are protected on the user’s computer. A user cannot access or view the file system from Windows Explorer or a DOS shell.

The third component is the SoftGrid Management Console. This console is an MMC snap-in that communicates with the SoftGrid Management Web Service running on the SoftGrid Server via Simple Object Access Protocol (or SOAP). SOAP is an XML-based protocol that is very light and generally used in Web environments bound to HTTP. The Management Console communicates with the Management Web Service to add or change configuration information. The SoftGrid Server in turn writes this information to the Data Store over a standard ODBC connection.

The fourth component is the SoftGrid Management Server. This server is primarily used for monitoring and reporting, and is an optional component that is not required for functionality. This component may be required as the environment grows in size, but can generally be left out of smaller implementations.

The remaining components in the diagram are probably already in place in your environment. These include the following:

- An Active Directory or NT domain
- Client workstations and laptops
- Terminal servers or a Citrix server farm.

MYTH: SoftGrid is a single point of failure on the network.

FACT: Anything you implement on any network can become a single point of failure. We recommend building your system to be highly available according to the requirements of the environment using load balancing, replication and other redundant components to ensure access.

The domain provides a list of users and groups to SoftGrid. Administrators then decide which users have access to the applications within the environment. You enable the SoftGrid Management Web Service to find users and groups by providing it with a non-administrative user account that has read access to the domain. This allows you to assign specific applications to users or groups within the Active Directory or NT domain.

The desktops and terminal servers both fall into the category of "clients." These machines act as SoftGrid Clients to the servers and, upon authentication, receive their applications via stream from the SoftGrid Servers.

It should also be noted that the SoftGrid platform can also run within a Microsoft Systems Management Server (SMS) infrastructure with SoftGrid for SMS, which enables you to manage virtualized applications from the SMS management console. In addition, you can use the SMS delivery mechanism to push entire virtualized applications out to target computer. Another unique benefit is that you can use SMS software asset discover and metering to track virtualized applications.

How the Components Work and What You Need to Deploy Them

In this section, we'll outline the basic requirements for each component and, if necessary, suggest ways to make that component fault tolerant.

SoftGrid Server

SoftGrid Servers manage and stream applications to clients over a network. They also handle authentication and licensing. They accept and respond to requests made by the SoftGrid Client and the SoftGrid Management Console.

SoftGrid Server recommended system requirements

The minimum requirements stated by most software vendors are generally less than optimal for the real world. To determine the requirements for a larger production system, you'll need to take into account the number of concurrent users and the number of applications being distributed.

SoftGrid Server:

- Intel Pentium III 1GHz
- 1GB RAM per CPU (512MB minimum)
- 200MB available hard disk space (not including application storage)
- Microsoft Windows 2000 Server/Advanced Server or Windows Server 2003

When determining the minimum hard disk space required, consider the following:

- OS requirements
- Space for SoftGrid log files (70MB minimum)
- Space required for SFTs. SFTs are application packages for SoftGrid (see below). A good rule of thumb is to take the size of an application—once it's installed on a PC—and add 10%.

Because the SoftGrid Server services are not very processor-intensive, there is no formula for processor sizing. You will run out of memory, disk or network bandwidth long before you run out of processor cycles.

On the OS side, the SoftGrid Server requires Windows 2000 Server (or Advanced Server if you're using Windows Load Balancing) or Server 2003 and requires that you install the Microsoft .NET framework (which is included with Server 2003) to enable the management features.

Creating a highly available SoftGrid Server

When defining high availability, just remember that SoftGrid acts most like a Web server or any other Web service. In other words, when determining your high availability solution, treat SoftGrid like a Web server. In our experience, both Windows Load Balancing and third party hardware load balancers fit the bill. In either case, you build multiple, identical SoftGrid Servers, and then load balance TCP ports 80, 554, and 8080. The clients then reference a DNS name that resolves to the load-balanced IP address. If a server in the group fails, the clients can continue to access the system.

SoftGrid Data Store

While the SoftGrid system does maintain a database of its configurations, this database is quite small. Generally, this database will only grow approximately 1 MB per 100 applications published. Because the majority of operations are read operations, there's not a lot on the hardware side that you'll be required to support.

If you're using MSDE, follow the guidelines for the SoftGrid Server itself and install the MSDE database onto that server. If you're implementing SQL, this database can be hosted on an existing SQL server, or you can implement a new one. In either case, SoftGrid supports MS SQL 2000 with Service Pack 4 and MS SQL 2005. If the SoftGrid database is the only database on the server, a single processor with 512 MB of RAM should suffice. If you're hosting multiple databases on the same server, you should ensure (prior to implementing SoftGrid) that the current databases are not already overloading the system.

Application metering considerations

If application metering is enabled, the following items should be considered:

- The Data Store will grow approximately 1K for every metering transaction (e.g., application launch, application shutdown, etc.). For example, if there are 2,000 users running an average of 5 applications each per day, the Data Store will grow roughly 10MB per day.
- The Data Store size is capped at 1GB and the oldest entries are automatically pruned when the database hits 95% capacity or roughly 950MB.

Making the data store highly available

Like any other database, this one can be replicated or clustered. If your SoftGrid Servers are centrally located (all in one datacenter or campus) a clustered data store will provide the utmost in high availability.

If your SoftGrid Servers are co-located or spread across the WAN with limited bandwidth between the sites, it may be best to replicate the data store using SQL replication. In this situation, it is recommended that administrators back up the SoftGrid database from the main site and restore it at the backup site on a regular basis using SQL Enterprise Manager, overwriting an existing default SoftGrid database.

SoftGrid Management Console

The SoftGrid Management Console is a GUI in the form of a Microsoft Management Console (MMC) snap-in that manages and configures the SoftGrid Servers and SoftGrid Management Server.

Only authorized administrators are allowed to perform system management tasks using the SoftGrid Management Console. Overall system management requests are only honored if the requesting client is authenticated using an account that had been designated as a SoftGrid Administrator. The following hardware and software are recommended for installing and running the SoftGrid Management Console:

- Intel Pentium III 600MHz
- 128MB RAM
- Microsoft Windows 2000, XP, or 2003
- .NET Framework 1.1

SoftGrid Client

The SoftGrid Client acts as an agent that is installed locally and facilitates application streaming and management through interaction with the SoftGrid Server. Both SoftGrid Desktop Client and SoftGrid for Terminal Services Client include a GUI and a command line interface (CLI) for client configuration (SFTMIME.) The SoftGrid for Windows Desktops Client is installed on user desktops and laptops, while the SoftGrid for Terminal Services Client is installed on a terminal server/Citrix server.

SoftGrid Desktop Client minimum system requirements:

- Intel Pentium III 600MHz
- 128MB RAM
- 10MB available hard disk space for installation + 2GB for cache
- Microsoft Windows 2000 Professional or XP Professional
- Any additional requirements needed by the applications and the OS

SoftGrid for Terminal Services Client minimum system requirements:

- Intel Pentium III 850MHz
- 2GB RAM (minimum 256MB, actual dependent on number of users and applications)
- 10MB available hard disk space for installation + 4GB for cache
- 1GB Page File
- Microsoft Windows 2000 Server/Advanced Server or Windows Server 2003 with Terminal Services enabled
- Any additional requirements needed by the applications and the OS
- Separate SCSI drives for OS and SoftGrid software for maximum performance

The Art and Science of Sequencing

What is sequencing?

Conceptually, sequencing is a lot like packaging in the traditional ESD sense. But the term “sequencing” doesn’t really describe the process well enough. When we think of packaging, we think of creating a set of files and configurations that will later be installed automatically on a client. In this sense, it’s all about getting the application to install properly, most of the time without user input.

Sequencing is really about *executing* or running the application, with the installation process as a means to that end. By installing the application using the SoftGrid Sequencer, you are creating not only the application package but also its virtual environment, and determining how it will be streamed to the client. Instead of trying to package the application for installation, you go through the basic installation of the application (into a virtual environment) and then into an execution phase where you actually run the application in the virtual environment. This provides packaging capabilities that don’t exist when creating a traditional installer. For instance, as an administrator, you now have the ability to package an application and prepare it exactly as you want it to execute on the target computers—including post-launch events such as product registration and additional configurations necessary the first time you run certain applications. All this can be done in an intuitive way, using the normal GUI interface that would be used if the application was being configured on a target machine (as opposed to creating transform files [MST] and other scripts necessary to modify an application installation on a client system.)

The trick with sequencing is to capture what files the application normally would install (and in what order these files are loaded at execution time) and determine what is needed OUTSIDE the application that will be placed in the virtual environment for it to execute properly.

All in all, sequencing an application is a much simpler process when compared with traditional packaging solutions. This is due to a number of factors:

- The installation (or monitoring) phase of sequencing is completely GUI driven. That is, an administrator installs and configures the application in the same way they would if installing the application locally on a desktop or terminal server. This is different than other packaging processes, which require the use of transform files (MST) or other scripts to perform post-installation configuration or customization.
- Sequencing alleviates the overwhelming majority of Terminal Services application integration issues (e.g., applications that only natively support a single instance per server, applications that have user preferences which leak across user sessions, etc.).
- Sequencings are highly portable across Operating System classes (desktop vs. Terminal Services) and families (Windows 2000 vs. XP and Server 2003)
- Sequencings can contain dependencies like specific versions of a database (Oracle, SQL, Sybase) client, OS component (MDAC, .NET CLR), browser component (JRE, JVM, plug-ins), etc.
- Updates or modifications to packages are always added directly to a sequenced package and eliminates the need to create and manage separate installers, scripts, etc. This can drastically reduce the number of individual installers (e.g., MSI's) in an application library.

How does the Sequencer work?

During sequencing, the administrator walks through the process of installing, configuring and executing the application. The first phase in the process is the installation of the application, during which the Sequencer records all interactions between the application and operating system. From this, the Sequencer gathers the information to create the basic package and the application’s virtual environment. Once the application is installed on the sequencing

workstation, you're able to configure the application as if it were a local application. This may include copying configuration files, modifying INI settings, etc.

After you've configured the application and it's ready to run, the Sequencer prompts you to execute the application and perform some of the same functions that your users will be likely to perform. The idea is to step through some of the common tasks the application will need to do upon opening. This creates the "sequence" part of sequencing (i.e., the order of the parts of the application that need to be downloaded first).

A good example is Microsoft Excel. In Excel, you would most likely select something in the File menu, and then maybe add some text and numbers into the spreadsheet, and maybe even add an equation. This tells the Sequencer that it needs to first download the minimum components required to launch the application, and then send down the files (or parts of the files/code blocks) that it needs to perform these functions. These will become the first parts of the application in SEQUENCE to be sent to the client. This allows end users to get the application up and running as fast as possible without having to wait for the entire application to stream to their desktop, laptop or Terminal Services session.

SoftGrid also supports sequencing of applications using a command line interface (CLI). This lets you sequence applications without using the Sequencer GUI. If an application's installer package has its own GUI, it is displayed during the sequencing process; otherwise the sequencing runs in a minimized state without prompting for input. The CLI can save significant amounts of time in environments that have large installer libraries (i.e. MSI, etc.) which administrators want to convert to SFT packages.

SoftGrid v4 now supports auto-detection of file type associations, DDE actions and shell extensions, eliminating the manual and error prone process of defining FTA for the sequenced applications.

What do I get when I'm done sequencing?

The Sequencer outputs a minimum of four files at the end of the process:

- An .SFT file ("SoftGrid" file)
- An .OSD file ("Open Software Description")
- An .ICO file ("Icon" file)
- An .SPRJ file ("Sequencer Project" file)

Each of these files has a specific use within the environment, and is described in detail below. .SFT, .OSD and .ICO files are stored on SoftGrid Servers and are made available to SoftGrid Clients (by various means) before the applications can be made available to users. .SPRJ files are needed to maintain/update SoftGrid (.SFT) files on a sequencing station.

.SFT file

The .SFT file contains one or more applications (without or without dependencies like database clients, OS/browser components like MDAC or JRE) that the Sequencer has repackaged—without altering source code—into a series of blocks of code that can be streamed. These blocks will have been optimized for rapid launch of the application on a SoftGrid Client.

The .SFT file also contains a record of files, directories, dependencies, initializations, and registry settings introduced by the application installer(s) or installation process and by the application when each one was launched during the sequencing process. When a client computer launches an application, it connects to the server, and the .SFT file is streamed—on demand—to it. The .SFT file is also referred to as a "package" because it is the virtual equivalent of a package in ESD sense.

.OSD file (see sample on following page)

The XML-based .OSD file contains specific information about the sequenced application and how to launch it. The user, through one means or another, launches this file to initiate the sequenced application's stream and launch. This file acts as a "SoftGrid shortcut" to the application, except that this file is XML and allows you to configure how the application launches.

.OSD files are simple text-based files that are created on the fly during sequencing, and can be modified in Notepad or any XML-editing tool. This file tells the client which server the package is on, what executable to launch in the package, and for which operating systems and configurations the package was created.

You can also modify the .OSD file to include registry settings, scripts, and environment variables that augment or otherwise override the virtual environment within the .SFT. This allows you to make small modifications to the environment without modifying the package, and also allows you to call scripts (either written into the .OSD file or called by the .OSD file) prior to the application launch.

.ICO file

The .ICO file, created in conjunction with the .SFT file, holds an application's icon that was extracted during sequencing. This icon is used for both desktop icons and file associations. A user will see a typical icon (like Word) associated with their desktop shortcut and will not notice the difference between it and a regular locally installed application icon.

.SPRJ file

The SPRJ (Sequencer Project) file is generated when a sequencer project is saved. The SPRJ file contains a list of files, directories, and registry entries that are excluded by the Sequencer. Load this file in the Sequencer to add, change, delete, or upgrade any of the applications in the suite. A common example of when you might use the SPRJ files is when you add service packs to an application.

```
<?xml version="1.0" standalone="no"?>
<SOFTPKG NAME="AutoDesk AutoCAD" VERSION="2002">
  <SUITE NAME="acad2002_2k_cts_trial"/>
  <ABSTRACT>3.0.1.16 Win2k3 J.D.</ABSTRACT>
  <TITLE/>
  <IMPLEMENTATION>
    <VIRTUALENV TERMINATECHILDREN="FALSE">
      <ENVLIST>
        <ENVIRONMENT
VARIABLE="PATH">%PATH%;%CSIDL_PROGRAM_FILES_COMMON%\A
utodesk Shared;</ENVIRONMENT>
        <ENVIRONMENT
VARIABLE="TEMP">%CSIDL_WINDOWS%\TEMP</ENVIRONMENT>
        <ENVIRONMENT
VARIABLE="TMP">%CSIDL_WINDOWS%\TEMP</ENVIRONMENT>
      </ENVLIST>
    </VIRTUALENV>
    <CODEBASE FILENAME="AC2002\acad.exe"
SYSGUARDFILE="AC2002\osguard.cp"
HREF="rtsp://{server_name}:554/ACAD2002_cts_trial.sft
"
GUID="AC135E96-7EFA-484A-96E8-F6844EBBBCF6"
SIZE="219511236"/>
    <VM VALUE="Win32">
      <SUBSYSTEM VALUE="windows"/>
    </VM>
    <RESOLUTION/>
    <OS VALUE="Win2003Svr"/>
    <OS VALUE="Win2003TS"/>
    <OS VALUE="Win2K"/>
    <OS VALUE="Win2KTS"/>
    <OS VALUE="WinXP"/>
  </IMPLEMENTATION>
  <DEPENDENCY>
    <CLIENTVERSION VERSION="2.0.5.0"/>
  </DEPENDENCY>
</SOFTPKG>
```

A sample .OSD file

Fine-tuning

At times you will find it necessary to modify certain portions of the packages or the .OSD files. Modifying the packages is really more of being able to discern the application's problem as opposed to following a list of troubleshooting "How to's." Still, adding a file or modifying an application is not hard to do. You simply use the Sequencer to re-open the package. Using the Sequencer interface, you can modify and change virtual registry settings and copy files into the package. To apply the changes, simply re-save the package.

Application suites

Because SoftGrid isolates each application into its own environment, it can cause confusion when users attempt to use a part of an application that integrates tightly with another application. This is common with an application such as SAP that generally has functions that hook to Excel and dump data into a spreadsheet for the user. Of course, SAP looks for a locally installed version of Excel, which will not be present in the real or local registry of the local if Office is being delivered via SoftGrid. Aside from simple file associations, all applications that need to “talk” to one another need to be contained within the same .SFT to provide integration between these applications. This creates a “suite” of applications. You can make each application within this suite available individually, or you can show the user only the primary application (with the secondary applications being called as needed). In either case, the applications MUST be sequenced together. In the case of SAP, this means installing SAP into the Sequencer, and then following the SAP install with an install of Excel into the same package.

Integration with locally installed software

SoftGrid-enabled applications can communicate with and leverage locally installed programs. For example, a sequenced CRM application could leverage a locally installed copy of Outlook for sending e-mail to clients. This virtual-to-local communication works for just about everything because the virtual environment can “see” the local registry and file system. The only difference is that a sequenced application cannot change the underlying OS. Any changes that it attempts to make will be captured in the virtual environment and stored in the SoftGrid User Profile (“SoftGrid Client” directory, which by default is stored in the Application Data directory).

For organizations planning on virtualizing the majority of the applications in their environment, sufficient time should be spent planning the core image (also known as the Standard Operating Environment or SOE) of their desktops, laptops and terminal servers to ensure that the appropriate applications and dependencies are locally installed and documented for the sequencing engineers. This normally includes software like the core (most current) version of Microsoft Office, Adobe Reader, WinZip and utilities like Antivirus, personal firewall as well as any other items used by everyone. Some organizations also include core versions of database drivers and other shared components. Then, during sequencing, it is unnecessary (though harmless) to include these items in the sequenced package because the virtual application can be configured to use the locally installed software.

Creating and following recipes

Once you’ve mastered the art of sequencing, or at least hit it right on for a single application, we recommend that you document your success. That way, the next time you update the application, re-sequence a new version, or have someone else do it for you, you won’t need to recall the few little changes that you needed to make it work. This document is called a recipe; and, like any good recipe, if it is followed to the letter, it should yield perfect results every time.

The SoftGrid Knowledge Base has both a recipe template as well as a white paper on sequencing best practices that you should download before you start sequencing in volume. Visit the knowledge base at <http://support.softgridcity.com>.

The real “time-to-deploy”

While SoftGrid has proved very capable of significantly improving the turn-around time for deploying applications, there are a few realities you still need to consider. It is important to note that, as described above, sequencing is somewhat of an art, and takes knowledge, practice, and intuition to perform with efficiency. We have found that this can be accomplished in a relatively short time, and once mastered, the time-to-deploy improves considerably. *(Note: It is recommended that individuals who will be responsible for sequencing applications attend official training and get certified on the product. Alternatively, customers can turn to a SoftGrid Consulting Partner, who has certified engineers on staff.)*

We have also found that you must consider a number of factors when planning a deployment of SoftGrid within an enterprise. Ideally, an application sequenced on one platform should work on all platforms. However, different platforms typically have different fonts, drivers, services, system directories, etc., set up on their base images. This makes sequencing on just one platform difficult, because missing components may differ from one platform to the next. These missing components need to be included in the sequenced application to ensure proper operation.

On average, the vast majority of the applications need to be sequenced only once to work on all platforms. For the remainder, time will be better spent sequencing the application separately on each platform rather than hunting down the variables. We have found that most of the applications sequenced on Windows 2000 work on Windows XP and Windows 2000/2003 with Terminal Services or Citrix.

License control

SoftGrid is designed to easily deploy, manage and terminate access to applications based upon the various software licensing mechanisms used across the industry. SoftGrid supports four types of software licensing models found in the Enterprise:

- *Named user licensing* – Only named users (users specifically identified in the system) are granted permission to use the application.
- *Concurrent user licensing* – This is where a block of licenses is available to the user pool on a first-come, first-served basis. For example, if a company has purchased 100 concurrent licenses for an application, 100 users can log in and be assigned a license for the duration of its use. If the 101st person tries to gain access to that application, he or she will receive a message stating that there are no more licenses available. Keep in mind that if you have users that are operating in disconnected mode, those users will ALWAYS be considered concurrent, increasing your license usage. SoftGrid currently does not have the capability to check licenses in and out for users that will go offline, so it is the responsibility of the administrator to make sure that license compliance is maintained. (*Note: Using SoftGrid does not allow you to “break” vendors’ software license agreements. As such, customers must purchase licenses in accordance with the license agreement of the software vendor.*)
- *Unlimited licensing* – This is where the system is set up so that any user with permission to access the server would also have access to that application. One example of where this would be useful is when a company has a set of applications they developed in-house and deployed on a SoftGrid system. The applications could be made generally available to all or to a subset of users.
- *Timed licensing* – This is the where an application is limited in some way, such as by time. The SoftGrid system grants a temporary license, but can set a time stamp on the validity of that license. This could be used, for example, with try-before-buy systems.

Real-life Deployment Considerations When Using SoftGrid

At this point, we would like to take you into the real world and discuss some of the issues that occur frequently in organizations that are looking to deploy SoftGrid. While not all of the issues below will exist in your particular situation, each should be considered before planning your evaluation or deployment.

.SFT storage

Obviously, since the .SFTs contain all the application files, it is important that you plan sufficiently for storing the files. The information in the sections above for planning the required capacity for the SoftGrid Server are a good guideline to ensure that you will have enough space. In an implementation with 50 or so applications, you can easily store all of the application files on a basic server with a RAID storage array. In some environments, our clients have found that it is beneficial to use a SAN or equivalent solution, especially as the number of applications grows.

Distributed environment

While SoftGrid scales very well in a centralized location, there are certain measures that you need to take to use it in a distributed environment. Because SoftGrid does not currently have a management solution for distributing packages across a WAN, you may want to use file replication (performed manually using XCOPY or FTP or automatically using Microsoft FRS, RoboCopy, etc.) to transfer the files from the central servers to the other servers during off hours (or even burn the files to CDs and mail them to the remote site). Then you can configure the clients in the branch site to stream their packages from a local SoftGrid Server(s). This can be done in a number of ways, including through the configuration of an environment variable, DNS or a hardware load balancing infrastructure.

Rather than deploying SoftGrid Servers across the WAN, you could use SoftGrid to load applications onto terminal servers or Citrix, and then have users access their applications over the WAN from there. Of course, this would make your users more dependent on the WAN for applications, but may be more efficient.

Applications with hard-coded paths

Another difficult situation is application installers that do not provide the option to choose the install path. SoftGrid runs applications in an isolated disk path (the default is "Q:\\" but can be changed to use any available drive letter) and if the application is set to run from "C:\\" you will be unable to execute. It is possible to work around this limitation with some of these applications.

As you are sequencing, let the application install where it needs to (most likely the C: drive). While still monitoring your install with the Sequencer, go to the Q: drive and create a "dummy" folder that will become the installation path. When you turn off monitoring, you will tell the Sequencer that the "dummy" directory is the install point for this application. When the Sequencer does its scan for changes, it will put everything in the VFS (virtual file system). You may have to try this a few times to get it right, but again, with practice the process becomes much easier.

SoftGrid for Web Applications

SoftGrid can launch applications OUTSIDE of the package or software that is installed locally on the target system. Remember the example from an earlier section of Internet Explorer with a sequenced JVM. What this means is that things like Flash, which can operate as a plug-ins to Internet Explorer and run within an Internet Explorer window, can be sequenced as part of a package. When delivered to a user's session, the package will call the locally installed IE and

bring it into the virtual environment. This version of IE can now “see” into the package that includes Flash or any other “plug-in” applications.

One of the benefits of running IE in a virtual environment is that SystemGuard protects the OS from the actions of the browser. This includes things like Spyware, which can be contracted during the course of web browsing. To combat this, an administrator could replace the default IE icon (that appears by default policy on the desktop, Quick Launch Bar, and the Start Menu) with a SoftGrid shortcut that points to a package that contains your standard browser configuration (e.g., versions of things like Java and Flash and browser settings like your organizations homepage and proxy server). When users want to launch Internet Explorer, they will click on the SoftGrid-supplied link instead of the default IE icon, which launches the .OSD and in turn launches IE in the virtual environment. As users browse the web, SystemGuard will protect their machines from the effects of Spyware while simultaneously allowing objects like Cookies and Favorites to fall through to the local OS. If a user contracts some form of malware, the underlying OS of their machine will be unaffected and they can be reverted to a clean version of the virtual environment by simply removing the package from their system and performing a desktop configuration refresh. (The easiest way for a user to do the refresh is to logoff and log back on to their system; alternatively, it can be done using the Control Panel Applet for the SoftGrid Client or through an Application Refresh Button available for download from the SoftGrid Support Knowledgebase.) Reverting the user’s IE virtual environment will not remove his or her Cookies or Favorites, which again are stored in their normal locations in the local OS.

Windows User Profile storage

The SoftGrid client saves user-specific files in the user’s Application Data directory under a “SoftGrid Client” directory by default (this can be changed, to a user’s home directory, for example). These files are normally installed on the local OS and modified by the user during program use. Because the application is not installed, these “user files” must be stored in a location specific to each user.

For roaming profiles in a TS/Citrix environment, the Application Data directory can get very large and slow down logins. The preferred way to alleviate this issue is to redirect the Application Data folder and exclude it from copying during login via Group Policy. This will improve the access time for the data and keep your terminal server logins efficient and fast. Alternatively, you could store the SoftGrid user-specific files in users’ home directories on a centrally accessible network store.

The Kinds of Problems Being Solved by SoftGrid

If you've read this far, you know that SoftGrid can add a great deal of value to your IT infrastructure. In fact, your local SoftGrid team has probably done a good job in helping you explain to your organization why it is important for you to implement SoftGrid. In this section, we will take a technical look at some of the problems SoftGrid talks about solving and show you how they can be solved for you as well.

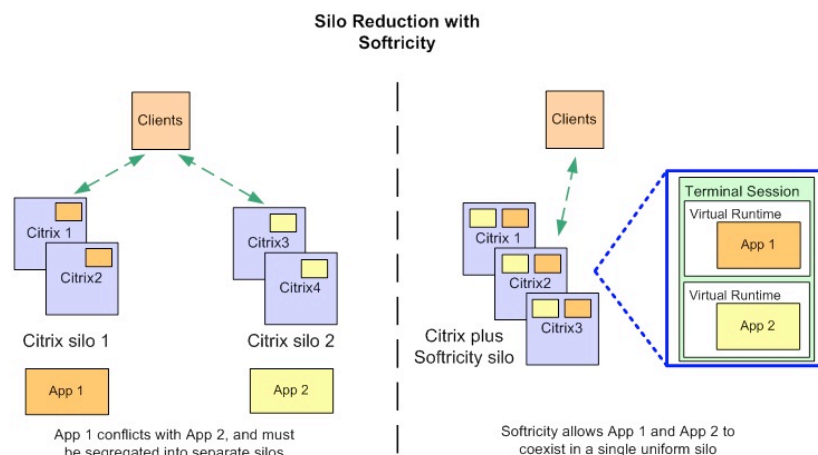
Application conflicts

As we've already mentioned, the resolution of application conflicts has far-reaching benefits. The virtualization and complete isolation of each application allows the operating system to maintain its original deployed configuration, and thus retain its stability. Applications that might normally share system files and overwrite vital components never touch one another. Applications are still able to communicate with other SoftGrid applications or locally installed applications through file associations and other shared system services such as cut and paste, OLE, and printers.

Reducing the number of silos in a Terminal Services environment

Server-based computing with Microsoft Terminal Services or Citrix MetaFrame is a powerful methodology for deploying applications to an enterprise. When it comes to actually implementing the server farms, however, you need to focus on application configuration, primarily when designing the farm layout. In a traditional scenario, you need to test each application in conjunction with all others that will reside in the server farm. Whenever you discover conflicting applications, another silo must be formed to segregate them from one another. Both MetaFrame and Terminal Services handle this separation well, but having to continually create silos becomes expensive in terms of hardware and the associated labor costs of maintaining and supporting each server.

Adding SoftGrid into the equation simplifies any siloed Terminal Services or Citrix environment by eliminating the silos. You create one uniform farm running all applications across all servers. The benefits of single silo design include: reduced hardware (number of servers required), reduced server licensing costs, reduced monitoring and maintenance costs associated with the farm, standardization of the farm on one single OS image per hardware type, etc.



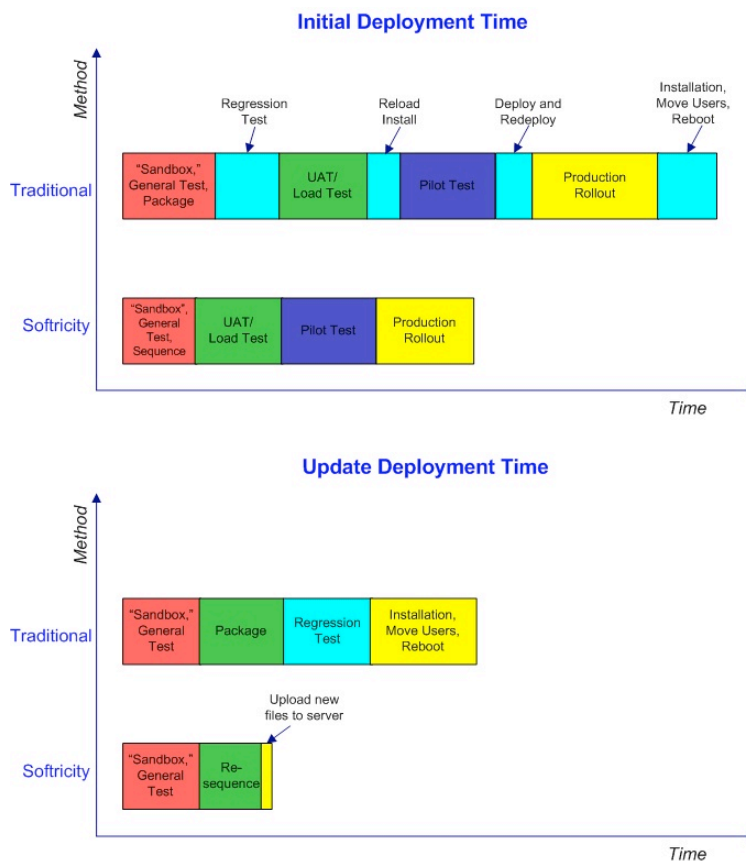
You shouldn't take reducing the number of servers in a Terminal Services or Citrix farm lightly. Additions to a farm due to siloing can have both short-term and long-term cost ramifications. A simple addition of three servers to support a new app combines not only the initial hardware purchase and licensing costs but also the upgrades, yearly maintenance, IT staff support time for patches and upgrades, not to mention power, rack space and network ports. Add that up over a 3-year period, and you'll agree that those three servers will cost a company a significant sum. Using SoftGrid, the farm can host multiple versions of applications, multiple configurations of one application, or run otherwise conflicting applications, and can consolidate the server resources for the most efficient use of the servers.

This single silo design allows the load balancing of users across all servers, as opposed to a limited number of servers per silo. This frees you to manage your resources based on the server load, as opposed to application configuration. With better utilization of the existing servers, you can free servers for repurposing or future growth.

Regression testing

If you have been an administrator for a while, you know how painful it can be to your time and stress level (and career) to deploy an application to a server farm or to desktops that ends up breaking things. Any downtime is considered by users to be unacceptable, right? That's why building and maintaining a TS/Citrix farm or desktop environment to be stable and available at full capacity requires that strict change control policies be instituted. Without these policies, any change, even the slightest patch of a single application, can negatively affect hundreds of users.

We generally recommend a three-tiered testing regime to reduce the possibility of a problem hitting the general user population. When a change is initiated, the IT staff should run the change through its paces in an environment that mimics the production environment, but is completely isolated from it. We call this a "sandbox" for application developers and IT personnel to play in. No harm, no foul. In this sandbox, all the nuances of the deployment of the change should be tested out. Normally, after the application is successfully installed and configured in the "sandbox," regression testing needs to occur to determine if the new application or application update "breaks" anything on the production desktop or server build. If a conflict is uncovered during this process in a non-SoftGrid environment, the new application or update needs to be omitted (for desktops or laptops) or a new silo needs to be created (for a Terminal Services or Citrix farm.) Using SoftGrid, however, eliminates the entire regression testing process and in the case of a TS/Citrix environment, removes the need to create a new silo. This dramatically reduces the time needed to implement new applications or updates.



The next tier moves the change to another isolated environment called User Acceptance Testing, or UAT. This is where functional testing and load testing will occur, including an initial pilot test. SoftGrid will help here as well, because if a fix is necessary, the fix can be rolled to the testing machines in seconds rather than hours by traditional means. And the application being tested will be standard, since everyone will be running from the same .SFT, reducing the number of variables to be tested.

Finally, the application is ready for a production pilot and rollout to the general user population. This is also enhanced by SoftGrid, as updates will not require a complete reinstallation of the application on the machines. For TS/Citrix environments, rebooting servers, pushing users off one server and onto another (and then back again) is reduced, and a rollout can occur in much less time.

Multi-user application problems

Oftentimes administrators use SoftGrid to solve multi-user problems in applications. These problems generally occur when an administrator wishes to deploy a new application via Terminal Services or Citrix but during testing determines that the application does not function properly within a Terminal Services/multi-user environment.

There are several reasons why applications will not run or run improperly in multi-user environments. The majority of applications with these types of problems can be made to run in Terminal Services without issues using SoftGrid. Some of the common application problems associated with a multi-user environment include:

- Application uses an .INI in the system directory and all users must share it
- Application locks a common file (like a Java file) and will only allow one user to run it at a time
- Application writes registry information for the user to *HKEY Local Machine* instead of *HKEY Current User*
- Application sees itself executing and will not start another process.

Out of the four common problems listed above, SoftGrid solves the first three almost every time. The fourth issue (while not as common as the first three) is not solved by SoftGrid. Because the application still executes locally on the terminal server, its executables still show up as loaded and running. So if your application looks for a running instance of itself, SoftGrid cannot help.

Intangible benefits explained

There are a number of other benefits you get when implementing SoftGrid, some which are organizational or process-oriented, and some which are economic. These are hard to define precisely in terms of dollars, but they do provide real advantages to an IT organization.

- *Centralize management and access* – By centralizing control of applications, the IT organization can respond faster to user needs, better control all aspects of the user experience, and mitigate a wide variety of support problems. By allowing a single interface to manage all applications across all TS/Citrix servers and desktops, the repetition of management tasks is reduced to a single iteration. This results in considerable time, labor, and cost savings.
- *Manage software licenses* – A number of SoftGrid users have acquired the SoftGrid platform specifically to control their software licenses after having to pay huge fines from software vendors such as Microsoft. Within the SoftGrid Management Console you can manage and track all software licensing from a single machine. By tracking real-time license usage and activity per user, you can identify and reassign inactive licenses, rather than paying the cost for additional licenses for new users.
- *Increase user productivity* – Every software vendor claims this, but, considering the capabilities of SoftGrid, many of the points have merit. You can determine for yourself the true value these have for your organization:

- *Support* – When a user has a problem with an application and calls the help desk, the last resort fix for an application problem is usually to go to the user's system, uninstall and reinstall the application. With SoftGrid, merely by unloading and reloading the cache, the application is fully restored. Support for this problem just went from hours to minutes.
- *New functionality* – If your software vendor has released a new version of the software with needed functionality, the time to deploy this is significantly reduced from traditional means, and this can have an effect on the productivity of your organization.
- *Improve flexibility in testing* – As mentioned above, it is still vital for an organization to thoroughly test and pilot an application. SoftGrid gives you advanced flexibility in a number of ways. For one thing, a pilot can be deployed without having to go to great lengths to set up a clean environment, or swap out testers' systems. Instead, the application can be pushed only to the pilot group, and even run alongside the current version. If the new version fails, users can simply click on the older version to continue their work, eliminating any lost work time. Also, if a problem gets past rigorous testing and gets deployed to the organization, rollback of that bad version can be instantaneous rather than a reversal of months of preparation. And, since everyone testing the application will be working from the same sequenced package, the variables that could occur with differences between installations are greatly reduced.

Using SoftGrid allows for a significant reduction in risk of lost productivity and other problems that become critical to an organization. If your company's goals for the introduction of new technology are to automate processes and utilize human resources more efficiently, SoftGrid will help get you there.

What's New in SoftGrid v4

SoftGrid v4 combines the latest advances in application virtualization with software streaming technologies to create an application environment that is available on-demand, flexible to change and significantly reduces the costs of software deployment and management. The latest version is based on thousands of hours of development, combined with feedback from hundreds of customers and partners who have been using SoftGrid for the past six years to virtualize over 25,000 applications and deploy to enterprises with tens of thousands users.

SoftGrid v4 adds new features that significantly extend the value of customers' virtual application environments. With these new features, customers will be able to virtualize more applications at much faster speeds, manage them more efficiently, and patch and update applications with greater ease.

Improved "Return on Virtualization™" savings result from enhancements in the following areas:

- Dramatically faster application sequencing and streaming
- Greater centralized control of all applications throughout the network, including remotely controlling SoftGrid clients
- Quicker deployments of new applications and seamless upgrades to new application versions
- Faster provisioning and de-provisioning of applications to end-users

Key new features of SoftGrid v4 are as follows:

Virtualization

Additional virtualized application configurations	What it does	Customer benefits
Virtualized Windows Services	Virtualizes Windows Services, programs that normally start at run-time to support functions from other applications (i.e., Hummingbird, FlexNet). Services begin and shutdown in conjunction with the main application's execution.	Enables customers to virtualize all aspects of any Windows application, without any extra effort.
Virtualized Transactional User Profiles	Separates the state of user interactions from the application's own state.	Shrinks Windows profiles while enabling seamless roaming between machines. Ensures user application profiles are consistent and easy to restore in case of application or hardware failures. Reduces boot time of Windows since target computer has fewer local configurations (since the applications configurations are virtualized).

Streaming

New feature	What it does	Customer benefits
Memory Locked Cache	SoftGrid File System provides more robust application memory allocation, enabling applications to be locked in memory.	Preserves all applications in the local SoftGrid application cache even in the case of unexpected computer shutdown. Applications are never lost, preserving end-user productivity.
Faster Streaming	On average, streaming has been accelerated by approximately 40%.	Faster access to applications the first time they are requested on-demand.
Unattended Application Pre-Cache	Ability to pre-cache computers with an entire application, even with no users logged onto the system (with Microsoft SMS).	Useful in both desktop and Terminal Services environments where SMS will send packages to the server and automatically pre-cache them into the SoftGrid virtual file system on a scheduled basis. Eliminates need for administrator to manually schedule delivery.
End-User Pre-Caching	Ability for end-users to initiate pre-cache applications for disconnected use (once authorized by IT).	Eliminates time previously required by IT to configure for mobile mode, eliminates lag time for end-users and enhances end-user productivity.

Sequencing/Packaging

New feature	What it does	Customer benefits
Agile Sequencing	Simplifies and speeds virtualization of applications.	Most applications are sequenced in 1/3 to 1/2 the time previously required. Even extremely large applications are drastically accelerated, i.e., takes just 60 minutes to sequence Cerner Millennium, a 1.5GB application.
Batch Sequencing	Automates batch packaging and enables "sequence once, run anywhere" on multiple Windows operating systems.	Reduces the time and effort required to virtualize applications, regardless of the number and type of Windows OS supported.
Enhanced Local Interaction	Automatically detects file-type associations and automatically associates DEEs.	Dramatically shortens and simplifies the application publishing process, and makes it easy to integrate one application within another.
Sequence Auditing	Sequencer logging and information display capability for each virtual application, including its update history.	Makes it easy to any IT staff to replicate sequencing without duplicating efforts, re-inventing the wheel.

Management

New feature	What it does	Customer benefits
Remote App-Desk	Ability for IT to remotely manage SoftGrid clients on desktop, laptops, terminal servers and virtual machines.	Allows IT to diagnose and remedy application problems without visiting the remote computer, and without end-user involvement.
Active Upgrade	Ability to update application versions without having to disconnect end-users running an earlier version of the software, or take down machines for a change window.	Eliminates time-consuming, costly change windows processes.
One-click Server Migration	Ability to migrate a SoftGrid Server's configurations to one or more servers with the click of a button.	Enhances scalability of SoftGrid platform.
Real-time Monitoring and End-User Session Control	Ability to monitor the performance of current user connections, including the ability to terminate an end-user's session.	Improves management of TS environments, i.e., can terminate if a user leaves a session open to free up concurrent license.
Application Check-pointing	Can periodically take "preferred state" of applications – including all user preferences and application states – and easily back up and restore the application and state at a later time.	Enhances business continuity for applications.
Hierarchical Application Tree	Assign properties to groups of applications within a nested, hierarchical model – allowing a single UI and management structure.	Easily manage thousands of applications across thousands of users.

Application-Level Security

New feature	What it does	Customer benefits
Application Lock-Down	Ability to offer a read-only operating system with fully locked-down security template or group policy object.	Protects applications from being altered or used in an unauthorized manner.
Virtual Administrator Mode	Run applications that require administrative rights in user mode.	Enables end-users to use applications without giving access to entire system and putting it at risk.
Access control	Controls access to applications by giving it only to end-users pre-authorized by IT. This feature works even in an offline mode.	Ensures that applications and their corresponding licenses do not "walk away" when a computer is stolen or lost.

About Microsoft® SoftGrid®

Microsoft SoftGrid transforms corporate computing by making software as instantly available and easy to use as electricity. Using patented SoftGrid technology, including application virtualization, policy-based management and on-demand application delivery, it changes applications into network services that no longer need to be installed. The result is a highly scalable software environment that is securely deployed, managed and immediately available anywhere in the world at a fraction of traditional IT investments – and that delivers dramatically improved business agility and a superior end-user desktop experience.

SoftGrid is the engine that turns applications into centrally managed virtual services that are delivered on-demand. It includes SoftGrid ZeroTouch, the front-end system that enables real-time worldwide availability to applications in a secure, self-serviced environment.

Together, SoftGrid and ZeroTouch consolidate each step of the application management lifecycle and change it from a series of separate, manual processes into an automated, unified system. They enable IT to overcome inherent desktop management challenges that, until now, had been unavoidable. As a result, they make it easy to transform your entire IT environment into a utility: centrally manageable, measurable and easily used by IT and business users with a simple click of a mouse.

Conclusion

To date, we have found an overwhelmingly positive reception to the possibilities inherent in the SoftGrid technology. Our customers have spanned the range from disbelief to overt excitement, but most still say: "That could really help here!"

In this paper, we've shown you that behind the claims and pretty graphics, SoftGrid in fact solves critical problems unaddressed by legacy tools on the market. This product does what it claims, and with careful planning, will actually save you significant time, money, and stress.

While this paper can be a resource for you before or during your implementation, there are other resources you can consult for even more in-depth information about SoftGrid. A SoftGrid knowledgebase is maintained at the public SoftGrid website (www.microsoft.com/softgrid).