Best Practices Guide for Provisioning Services and XenApp

*Designing an enterprise solution for the fast provisioning of XenApp servers*
# Table of contents

Best Practices Guide for Provisioning Services and XenApp ............................... 1
Overview .................................................................................................................. 1

## vDisk Options

- vDisk Types ........................................................................................................ 1
- vDisk Cache ......................................................................................................... 2
  - Target Device – RAM ...................................................................................... 3
  - Target Device – Local Storage ......................................................................... 3
  - Target Device – Shared Storage ........................................................................ 4
  - Provisioning Services – Local Storage ............................................................. 4
  - Provisioning Services – Shared Storage .......................................................... 5

## Network Options ..................................................................................................... 5

- Boot Device Manager .......................................................................................... 5

## Operating System Tuning ....................................................................................... 6

- Event Logs ........................................................................................................... 6
- Auto Update .......................................................................................................... 6
- Group Policies ...................................................................................................... 7
- Organizational Units ............................................................................................ 7

## XenApp Tuning ......................................................................................................... 8

- Application Delivery ............................................................................................. 8
- Application Cache .................................................................................................. 9
- Automate Application Pre-Cache ......................................................................... 11
- Pagefile .................................................................................................................. 15
- Multiple Partitions ............................................................................................... 15
- Drive Remapping .................................................................................................. 15
- Web Interface ....................................................................................................... 15
- Data Collectors ..................................................................................................... 16

## Application Tuning .................................................................................................. 16

- Machine Specific Registry Keys .......................................................................... 17

## Maintenance ............................................................................................................ 17

- Automatic Updating of vDisks ............................................................................. 17

## Revision History .................................................................................................... 19
Overview

Being able to add new XenApp servers to a farm in a matter of minutes, guaranteeing server consistency across similar XenApp workloads and being able to quickly rollout changes to an entire farm are some of the significant benefits for dynamically delivering XenApp by using Provisioning Services. Although the solution is fairly simple to setup and implement, a few areas must be considered if the environment is to work most effectively. These areas cover the following topics:

- vDisk Options
- Network Options
- Operating System Tuning
- XenApp Tuning
- Application Tuning
- Maintenance

vDisk Options

XenApp servers require a solution that delivers speed, consistency and availability while simplifying the environment. These core criteria will help determine the best approach for designing a solution that integrates Provisioning Services with XenApp. The first decision area focuses on the appropriate vDisk options for a XenApp workload:

- vDisk Types
- vDisk Cache

vDisk Types

The first vDisk type is Private Image mode, where each target device will have its own vDisk. The vDisk is configured in a read/write fashion, where all changes are stored within the vDisk for future use. To improve consistency and manageability, Provisioning Services can stream images to multiple devices from a single vDisk. The vDisk is configured as read-only to allow for the sharing between the multiple target devices. Prior to Provisioning Services 5 SP1, any write that the target device tried to perform on the vDisk was stored in a write cache, a temporary storage location destroyed each time the target device was restarted. With Provisioning Services 5 SP1, there is now the option for the re-use of the write cache after a reboot occurs, using a solution called Differential Disks.

With the addition of differential disks, in what circumstances should changes be saved between reboots and when should the changes be destroyed? The best decision is based on the scenario and the benefits of each disk type.

<table>
<thead>
<tr>
<th>Description</th>
<th>Standard Image</th>
<th>Private Image</th>
<th>Differential Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each target devices stores vDisk writes in a unique change file that is destroyed upon each target device reboot.</td>
<td>Each target device has a dedicate vDisk image configured in a read/write fashion. All changes are part of the vDisk.</td>
<td>Each target device stores vDisk writes in a unique file that is kept upon subsequent reboots, allowing the server to keep configuration changes, until the base vDisk is modified.</td>
<td></td>
</tr>
<tr>
<td>Benefits</td>
<td>Servers revert back to a consistent, optimized and approved state after each reboot</td>
<td>Complete personalization of the environment because all changes are stored, but at a cost of storage and support.</td>
<td>Allows for greater levels of system personalization by not discarding system-level changes upon subsequent reboots.</td>
</tr>
</tbody>
</table>
write cache is reset after each reboot

**Recommendations**

| Standard images are a recommended best practice for XenApp servers. XenApp servers delivering the same applications should be: |
| Consistent so users do not experience differences between servers |
| Optimized to allow for the best application response times and processing speed |
| In an approved configuration, especially for industries requiring strict enforcement of standards for certification or compliance. |

| There is little need for Private Images in a XenApp environment because of the differences each image will take. This will go against the core best practice of consistency. |

| Differential images are appropriate for a small subset of use cases where users have the need to install their own applications. In a XenApp environment, this is not recommended. Once the base vDisk is modified, the differential image is destroyed and the user must rebuild their personality into the target device. |

**vDisk Cache**

One of the major decisions when designing a Provisioning Services solution for XenApp workloads is focused on where to store the write cache. Because a group of servers utilize the same vDisk image, the vDisk is configured as read-only. However, this brings about challenges because simply turning on a streamed server results in the server trying to write information into the base vDisk. Instead of writing the changes into the base vDisk, the changes are placed into a write cache. Unless differential disks are used, which are not recommended for XenApp, the write cache is destroyed each time a server is rebooted, thus restoring the server to the base image.

The write cache contains anything that is changed during the time between server reboots. Each server has its own write cache. Depending on what the server is doing and how the environment is configured, the write cache could grow quite large. For instance, starting the server adds to the write cache for server personalization, XenApp local host cache, pagefile, etc. If an application is streamed onto the XenApp server, the entire application profile will also increase the size of the write cache. The more things that change on the server, the larger the write cache will become. Provisioning Services allows the write cache to be stored in many different areas, each providing their own benefits. The options are:

- Target Device – RAM
- Target Device – Local Storage
- Target Device – Shared Storage
- Provisioning Services – Local Storage
- Provisioning Services – Shared Storage

Each write cache storage option has many different benefits and concerns, especially for the XenApp workload. For most XenApp environments, the best solution will be the one that takes on the following characteristics:

- **Fast:** XenApp requires a solution that responds quickly as XenApp is maintaining live, interactive user sessions. Any delays in the write cache might be noticeable to the users.
- **Dynamic:** XenApp servers are delivering many different applications and supporting many different users. Each user and application will have an impact on the write cache. The amount
consumed will change day-to-day. The risks of exhausting write cache space would be detrimental to the success of a XenApp environment.

- **Available:** XenApp servers must be protected from environment failures, because each server is supporting many users simultaneously. The write cache solution selected should be one that does not impact high-availability options from functioning.

Based on the aforementioned criteria and the explanation of the different options for write cache, XenApp servers provisioned with Provisioning Services are best suited for

1. Virtual XenApp Server: Target Device – Shared Storage
2. Physical XenApp Servers: Target Device – Local Storage

**Target Device – RAM**

- **Definition:** The first option for write cache storage location is the target device’s RAM. A portion of the target device’s RAM is set aside and used for the write cache.

- **Benefits:**
  - The main benefit for using the target device’s RAM is it provides the fastest type of write cache.

- **Concerns:**
  - A portion of the RAM cannot be used for the server workload. RAM is often better used for XenApp applications or user sessions than for write cache. Plus, using RAM to support the write cache is more expensive than using storage.
  - Part of the challenge with using RAM as the write cache storage is determining the amount of RAM required. Provisioning Services can set aside a certain portion of RAM for the write cache, but what happens when the RAM runs out? The write cache is critical to the stable functioning of a provisioned server. When available write cache is exhausted, the server can no longer make changes, which results in a server failure. If the write cache size is not estimated correctly, using a Target Device’s RAM might pose detrimental to the stability of the environment.

**Target Device – Local Storage**

- **Definition:** The second option for write cache storage location is the target device’s local storage. This storage could be the physical disk drives on the physical server, or it could be the virtual disk on a virtual server

- **Benefits:**
  - This solution does not require additional resources, in that most physical servers being provisioned already have local disks installed and unused.
  - Although target device local storage is not as fast as RAM, it still provides fast response times because the read/write to/from the write cache is local, meaning that the requests do not have to cross the network.
  - Trying to estimate the size of the write cache is difficult and if done incorrectly, can result in server failure. However, local storage typically provides more than enough space for the write cache, without requiring the administrator to estimate space requirements.

- **Concerns:**
  - If the target device is virtualized, using local storage will prevent live migration processes from succeeding because the storage is not shared across virtual infrastructure servers, like XenServer.
Target Device – Shared Storage

- **Definition:** The third option for write cache storage location is on a shared storage device attached to the target device. This solution is usually only valid for environments virtualizing the target device with a solution like Citrix XenServer. This storage is assigned to each virtual machine from a shared storage repository.

- **Benefits:**
  - Although target device shared storage is not as fast as RAM or target device local storage, it still provides fast response times. If the shared storage infrastructure is a SAN or NAS, the reads/writes will still perform adequately because the optimized shared storage infrastructure will help overcome the time added for traversing the network.
  - Although the configuration of this solution requires the identification of the shared storage size, the costs associated with over-estimating are not nearly as detrimental as overestimating with RAM. Storage costs are significantly cheaper than RAM so a sizeable buffer over the space estimates is of little concern.
  - Because the target device’s storage is accessible from multiple virtual machines, virtual server live migration, like XenServer XenMotion, is viable.

- **Concerns:**
  - This solution requires the setup and configuration of a shared storage solution. Although if XenServer is already being utilized, the same shared storage solution can be used for the write cache storage.

Provisioning Services – Local Storage

- **Definition:** The fourth option for write cache storage location is on the Provisioning Services’ local storage. This storage uses the physical disks installed within the Provisioning Services.

- **Benefits:**
  - This solution is extremely easy to setup and requires no additional resources or configuration within the environment.

- **Concerns:**
  - Requests to/from the write cache must cross the network and be serviced by the Provisioning Services streaming service. Because the write cache is across the network, servicing write cache requests will be slower than the previously mentioned options.
  - The streaming service is responsible for sending the appropriate parts of the vDisk to all target devices. Having the write cache on the Provisioning Services server will negatively impact the server’s scalability because the streaming service must also service the write cache requests.
  - Provisioning Services includes a high-availability option, but in order for this solution to function, all Provisioning Services servers must have access to the vDisk and the target device’s write cache. When the write cache is stored on one Provisioning Services server’s local storage, this makes it impossible for other Provisioning Services servers to gain access, thus denying the ability to enable Provisioning Services high-availability.
  - Although disk space is fairly inexpensive, chances are the Provisioning Services does not have an extensive supply of storage space. With each Provisioning Services server supporting a few hundred target devices, it is quite possible the total write cache could exceed hundreds of gigabytes of storage space. This could result in
exhausting all local storage on the Provisioning Services server causing a server failure.

Provisioning Services – Shared Storage

- Definition: The fifth option for write cache storage location is on the Provisioning Services server’s shared storage. This option utilizes a share storage solution that is connected to the Provisioning Services server.

- Benefits:
  - The shared storage solution allows for Provisioning Services high-availability as each server can access the vDisks and the write cache.
  - Size concerns are mitigated because shared storage devices typically contain significant amounts of storage and can be expanded easily.

- Concerns:
  - This is one of the slowest solutions because requests to/from the write cache must cross the network and be serviced by the Provisioning Services streaming service. The Provisioning Services server must then forward the write cache requests onto the shared storage, thus resulting in two network hops for the write cache.
  - Provisioning Services scalability is impacted as the streaming service is responsible for handling Provisioning Services write cache requests and forwarding them onto the shared storage.
  - The solution requires the installation and configuration of a shared storage solution into the environment. If one is already present, then this concern is mitigated.

Network Options

In the most basic configuration, Provisioning Services utilizes DHCP to assign each target device with an IP address and the location of the boot server and boot image (options 66 and 67 if using PXE). However, in certain environments using DHCP or modifying the DHCP scope options to include PXE is not possible. In situations like this, you may wish to consider using the Boot Device Manager option.

Boot Device Manager

Boot device manager (bdm.exe) is a utility included with Provisioning Services that allows target devices to boot from a vDisk image without the need of PXE. The boot device manager allows the bootstrap image, which is used to connect the target device to Provisioning Services servers to receive the appropriate vDisk image, to be stored on a USB, CD-ROM, ISO image or a hard disk partition. Simply configuring the target device to use one of these alternative methods will allow the target device to boot using Provisioning Services without the need for PXE.

For XenApp environments, booting from the boot device manager’s image is done by the following:

- Virtual XenApp server on XenServer:
  - Store the boot image in an ISO and move the ISO file to the ISO Image storage repository on XenServer
  - Assign the boot image to the DVD drive for each virtual machine
  - Configure the virtual machine to boot from the DVD

- Physical XenApp Server:
  - Burn the ISO image to a DVD or CD
  - Keep the CD/DVD in the drive, as it is needed for each start up
Configure the server, in the BIOS, to boot from the DVD drive

Operating System Tuning

The base part of any vDisk image is the operating system. Having an environment that is optimized will help meet the goals of a well-tuned XenApp environment. This section focuses on the best practices for the following:

- Event Logs
- Auto Update
- Group Policies
- Organizational Units

Event Logs

Event logs are helpful to identify and troubleshoot issues within the environment, and they also act as a security audit log, which is extremely critical in high-security environments. Because the best vDisk type for XenApp workloads is a standard image, any changes made to the system are lost upon reboot, which includes the event log. This eliminates the possibility of tracking security audit logs, application issues or operating system issues.

Through registry keys, or a policy setting in Windows 2008, the location of the event log files can be modified to a different location, but that location must be a local drive. The event log service starts adding items before the network services start. If the event logs are stored on a network share, the event log will fail to save messages because the network is not yet available. However, events can still be consolidated by the following methods:

- Local Storage: If the vDisk cache best practice is being followed, each target device is configured with local storage: physical servers use local disk while virtual servers use a defined virtual disk. Either way the local storage appears as a locally attached disk to the provisioned XenApp server. Redirecting the event log to the defined local disk allows event log information to persist between server reboots.
- Citrix EdgeSight: EdgeSight for XenApp is included in XenApp Platinum. EdgeSight allows an administrator to collect, in real-time, any defined error messages from any XenApp host. This allows for the tracking of only the most important and critical alerts. The alerts that are tracked are completely configurable. The solution can work with XenApp servers running on Windows 2003 and Windows 2008.
- Microsoft Event Collection Services: In Windows Server 2008, Microsoft includes an Event Collection solution that collects all events for a configured alert level across any defined server. These events are stored within a single log file for future analysis.

Auto Update

Automatic updates for the operating system or installed applications are a great way to keep software current and programmatically address identified security holes. Even without Provisioning Services, it is a recommended best practice to disable the automatic updates for applications and the operating system because all changes to XenApp servers should be tested and validated before rolling out into production. The automatic update feature circumvents this best practice.

For systems delivered with Provisioning Services, automatic updates can create another potential issue. In a XenApp environment, a group of XenApp servers will be provisioned from a standard image, where changes are kept in the write cache and lost during reboot. If this logic is applied to the automatic update features of the operating system and applications, one can quickly realize that every time the server is rebooted, the update services will identify new updates and make requests to update. This will happen for every XenApp server after every reboot, until the base image is updated.
to include the new patches. As a recommended best practice, the operating system and applications should have the automatic update features disabled. The XenApp administrative team should assess each update. Once the update is approved, the base vDisk image for the XenApp server should be updated. When the XenApp servers are rebooted, they will be using a vDisk image containing the latest updates.

**Group Policies**

Active Directory group policies are a great way to enforce system-level or user-level settings on a group of servers. Using Provisioning Services brings about a new set of recommended policies, as explained in the following table:

<table>
<thead>
<tr>
<th>Policy Path</th>
<th>Value</th>
<th>Justification</th>
</tr>
</thead>
</table>
If the target device updates the password on its own, Provisioning Services will supply the wrong password, thus causing issues for domain membership. |

**Organizational Units**

As a long-standing best practice, XenApp servers have warranted their own organizational unit within Active Directory for organizational and policy enforcement purposes. The recommendation has also included breaking out specific XenApp roles into their own OU, so each identical group of servers would have the same policies applied. Typically, this creates an Active Directory structure like the following:

With the inclusion of Provisioning Services into the XenApp architecture, this recommendation does not change. In fact, this best practice becomes even more important because there will be special policy settings specifically for provisioned servers, explained in the previous section. Depending on how Provisioning Services is integrated with XenApp will help to determine if new OUs are required.

- If the OU contains a set of XenApp servers all provisioned with the same vDisk, then any Provisioning Services related policies can be applied to the entire OU.
- If the OU contains provisioned and non-provisioned XenApp servers, all hosting the same applications, then a new OU should be created that contains only the provisioned XenApp servers.
- If the OU contains provisioned and non-provisioned XenApp servers hosting different applications, then multiple OUs should be created containing only identical servers.

With Provisioning Services, the XenApp OU structure might resemble something like the following:
Each OU contains:
1. Similar servers: Applications, infrastructure components, XenApp components
2. Similar delivery processes: Provisioned or not provisioned

XenApp Tuning

Installed on top of the operating system is XenApp. The XenApp Prep utility manages many of the technical processes that must be done to a XenApp server so it can be provisioned successfully with Provisioning Services. However, there are a few other areas that must be considered:

- Application Delivery
- Application Cache
- Automate Application Pre-Cache
- Pagefile
- Multiple Partitions
- Drive Remapping
- Web Interface
- Data Collectors

Application Delivery

One of the major challenges with creating a base XenApp image is determining what to include and what not to include. Of course, you need the operating system and XenApp and Provisioning Services tools, but beyond that what is recommended and why? Take the following scenario: due to business reasons, an environment has three sets of XenApp servers hosting different line-of-business applications. All three line-of-business applications are dependent on Microsoft Excel for viewing and editing integrated spreadsheets. Should Microsoft Excel be part of the base image or should it be a streamed application? There answer is... there is no right or wrong answer; it is all dependent on other factors within the environment.

The decision to include core applications is oftentimes a result in the belief that the base image should contain the greatest number of items that are common between XenApp servers. If every server requires the same application, more network bandwidth will be used when the application is streamed to every server as part of the application streaming process. Also, application streaming, in the default configuration, does not initially start as fast as a previously installed application because the application must be sent across the wire. Thus, users will experience latency while the application is streamed for the first time (this latency can be overcome with application pre-caching, as explained in the Application Cache section).
There is also a business aspect to this decision. In some organizations, one set of administrators is responsible for applications and another set is responsible for the XenApp configuration. By separating the applications from the base image, the technical solution can align more closely with the organizational structure of the business.

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Base Image Application Inclusion</th>
<th>Base Image Application Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Base image and applications included in a single server role, which will allow for the fastest rebuild and delivery times.</td>
<td>• With fewer items included in the base image, maintaining the image is easier as only the core operating system and XenApp are provisioned initially.</td>
<td></td>
</tr>
<tr>
<td>• Less network bandwidth is used because the applications are already present in the base image and no additional installations are required. The complete image can be deployed during off hours so as to not impact availability of the XenApp servers during the day.</td>
<td>• XenApp base images are managed with one set of tools, and applications and corresponding updates are managed with a different set of tools. This makes it easier to have multiple administrators responsible for different areas of expertise.</td>
<td></td>
</tr>
<tr>
<td>• Application startup time is shorter because the application is already part of the image and does not need to be streamed.</td>
<td>• XenApp silos become a thing of the past. A single XenApp server image can be used to deliver any application at anytime. A single image to maintain for the entire XenApp farm greatly simplifies support.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concerns</th>
<th>Base Image Application Inclusion</th>
<th>Base Image Application Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Core application versioning changes can impact the base image. For example, if all servers require the same version of Microsoft Excel, no issues are raised. However, when individual application upgrades occur, some applications might require Excel 2007 while others require an older version. This would have a profound impact on the contents of the base image.</td>
<td>• Streaming applications will increase network utilization during startup, although this can be mitigated with application pre-caching.</td>
<td></td>
</tr>
<tr>
<td>• Modifying an application within an image will impact all servers that rely on the image. This might require different levels of change control for applications that are integrated with the base image and applications that are not integrated.</td>
<td>• In the default configuration, the first user starting an application will notice a pause before application startup as the application is streamed to the server. This can be overcome with application pre-caching.</td>
<td></td>
</tr>
<tr>
<td>• More images to maintain, which can make support more difficult, although the number of Provisioning Services images is still far fewer than the number of images required for other server build solutions.</td>
<td>• Not every application can be streamed. Although XenApp streaming has improved to include more applications, there are some that cannot be streamed because they require special functionality</td>
<td></td>
</tr>
</tbody>
</table>

Regardless of the decision on which applications to include and exclude in the base image, the following are general best practices for the base image:

- All relevant operating system and XenApp hotfixes and service packs should be included in the base image.
- The most common operating system and XenApp configuration should be used for the base image. If 80% of the servers require a specific setting while another 20% do not, the base image should include the special setting.
- The base image should include all appropriate XenApp plugins. If application streaming will be used, the streaming plugin should be installed as part of the base image.
- Depending on the usage of server certificates, the appropriate root certificate should be part of the base image.

**Application Cache**

Some environments have chosen to simplify their XenApp environment with the use of XenApp application streaming. This solution utilizes a XenApp server (provisioned or static) without any applications. Once the XenApp administrator publishes a streamed application to a XenApp server,
the application profile is streamed to the XenApp server as needed. The act of application streaming has a direct impact on the provisioned server.

The streamed application is a change to the base vDisk image. These changes are stored within the write cache for each provisioned XenApp server. Depending on the size of the application, the simple process of delivering an application on top of a Provisioning services’ streamed XenApp server can make the write cache grow by many gigabytes as shown in the following diagram:

Using a provisioned XenApp server generates the typical swap and temp files, which is added into the write cache. When a streamed application is requested for the first time, portions of the application profile are delivered to the XenApp server from the Application Hub. This process immediately increases the in use size of the write cache, in addition to delaying application startup times as the profile is delivered.

Depending on the write cache option selected, this could have a significant impact on the usability and speed of the XenApp server. If the write cache size is a concern, then a pre-cache option exists that speeds up application launches as well as reduce the size of the write cache. This process is shown in the following figure.
In this example, the vDisk image includes a pre-cache of the streamed applications. Users still have to access the applications as before, but the applications are already present on the vDisk so the application stream process is complete, eliminating the need to stream portions of the profile and impacting the write cache.

The challenge with doing the pre-cache of the streamed application is that each time a streamed application is updated, the application cache within vDisk image must also be updated. This adds more steps into the application update process. However, not every application update requires an update to the application cache on the vDisk, only major updates are a concern.

For example, if an application has a new patch or a new file update, simply updating the application profile is adequate. When a user tries to start the application on a provisioned XenApp server, most of the application cache is correct except for one or two files. Those two items are updated from the application hub, and only slightly increase the size of the write cache. However, if a large update to an application is performed, like adding a service pack to Microsoft Office, then it would be advantageous to refresh the application cache on the vDisk because these updates impact a large percentage of the files in the cache. When the application is executed, all of the updated files are streamed down and placed in the write cache.

A third option exists that is able to overcome the challenges with pre-caching and on-the-fly streaming, and that is Cache Redirection. The overall goal is to keep the maintenance of the environment simple while providing an optimized environment for the user so they experience a high-definition experience.

If the physical and virtual XenApp servers are designed in such a way that they are allocated storage to hold the Provisioning Services write cache, the same defined storage can also contain the application streaming cache, as shown in the following figure:

This type of architecture provides the following benefits over the other to previously mentioned options:

- Faster application launch after server reboots because the application cache persists
- Easier application maintenance because the application cache is decoupled from the vDisk

### Automate Application Pre-Cache

A common concern with streamed applications is the initial startup time. When a user tries to start a new application, the application is not yet on the server. The application request will start the streaming process where parts of the application are sent over the network to the XenApp server. Depending on the application and environment, application startup time can vary greatly. XenApp
allows streamed applications to be pre-delivered allowing for faster startup time. To accomplish this when using provisioned servers, do the following items:

**Note:** This process will increase the size of the write cache because the application stream happens to the vDisk in standard image mode (read-only). For a greater understanding into the write cache and application streaming, see the previous section on Application Cache.
### Pre-Cache Streamed Applications

<table>
<thead>
<tr>
<th>Screenshot</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Screenshot](image1.png) | On the Provisioning Server, launch the **Provisioning Server Console**.  
- Select a **Target Device** -> File -> Properties  
- Select the **Personality** tab and select **Add**  
- Enter in the following:  
  - Name: **RADE**  
  - String: Path to all application profiles to be delivered to the target device separate by commas.  
- Select **OK**  
- Select **OK** |
| ![Screenshot](image2.png) | Within Active Directory Users and Computers on the Domain Controller  
- Select the Organizational Unit containing the servers requiring pre-application delivery  
- With the OU highlighted, select **Action** -> **Properties**  
- Select the **Group Policy** tab and select **New** entering in a name for the policy  
- Select **Edit** |
| ![Screenshot](image3.png) | The Group Policy Object Editor appears.  
- Select **Computer Configuration** -> **Windows Settings** -> **Scripts (Startup/Shutdown)**  
- Highlight the Startup name  
- Select **Action** -> **Properties** |
Pre-Cache Streamed Applications

<table>
<thead>
<tr>
<th>Screenshot</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Screenshot](image) | The Startup Properties screen appears
- Select **Add**
- Select **Browse**
- Within the Browse window, right-click and select **New -> Text Document**
- Rename the document to **PreCache.vbs**
- Right-click the PreCache.vbs file and select **Edit**
- Paste the script, contained in the following row, within the file. **Save** and **exit** notepad
- Highlight **PreCache.vbs** and select **Open**
- On the Add a script window, select **OK**
- On the Startup Properties window, select **OK**
- Close the remaining windows and screens |

**3**

- **Set wshShell** = CreateObject("WScript.shell")
- **intLoop** = 0
- **intStart** = 1
- **intEnd** = 0

' Populate registry with Streaming Application paths based on personality settings
wshShell.run chr(34) & "c:\Program Files\Citrix\Provisioning Server\getclientinfo" & chr(34) & "ra\ /r=HKEY_LOCAL_MACHINE\SOFTWARE\BUILD\RADE", 1, true

' Read unique application names
strRadeApps = wshShell.RegRead("HKLM\Software\Build\RADE")
Set objRegEx = CreateObject("VBScript.RegExp")
objRegEx.IgnoreCase = True
objRegEx.Global = True
objRegEx.Pattern = ","
set colMatches = objRegEx.Execute(strRadeApps)

' For each pathname, pre-deploy the application
Do until (colMatches.count < intLoop)
  **intend** = Instr(intStart +1, strRadeApps, ",")
  if intEnd <> 0 then
    strApp = Mid(strRadeApps, intStart, intEnd -intStart)
  else
    strApp = Mid(strRadeApps, intStart)
  end if
  **intLoop** = intLoop + 1
  **intStart** = intEnd + 1
  wshShell.run chr(34) & "C:\Program Files\Citrix\Streaming Client\radedeploy" & chr(34) & "/deploy:" & chr(34) & strApp & chr(34)
Loop
When the XenApp servers contained within the organizational unit starts, the script will execute reading in the personalization settings. The personalization string for RADE will pre-deploy the streamed applications to the server.

Pagefile

Many XenApp configurations have included a configuration change to move the pagefile to a partition that is different from the system partition (C: drive). In some situations, if the pagefile partition was on a completely different physical disk, performance improvements might be gained. If XenApp is delivered with Provisioning Services, this configuration should be reexamined.

The pagefile is undergoing constant modification by the operating system. As physical memory is paged, it is stored in the pagefile. Upon reboot, the contents of the pagefile are not important, and it is overwritten on subsequent startups. In a Provisioning Services environment, the pagefile will be located on the vDisk, but any change made to the file will be recorded in the write cache, because the vDisk image is read-only. Assigning the pagefile to a different partition within the vDisk is not recommended, because there is no benefit to the functioning of the XenApp server.

Multiple Partitions

Each XenApp environment is configured with many different partition configurations. Typical configurations are:

- One Partition Server:
  - Partition 1: Operating System, Pagefile, Applications

- Two Partition Server:
  - Partition 1: Operating System, Pagefile
  - Partition 2: Applications

- Three Partition Server
  - Partition 1: Operating System
  - Partition 2: Applications
  - Partition 3: Pagefile

 Provisioning Services can deliver an image with one or two partitions, but not three. Plus, based on the Pagefile section above, there is no gain from having a separate pagefile partition on a provisioned XenApp server as the pagefile changes will be stored in the write cache.

 Although two partitions are supported, it does increase the complexity of the environment and does not offer any performance benefit to the environment. As a general best practice, XenApp servers should be configured with a single partition.

Drive Remapping

In a few XenApp installations, the XenApp server drives are remapped to an alternative drive letter (C = M, D = N). The most common justification for this design consideration was for users to have their client drives mapped within their XenApp sessions as C and D. However, this approach should be re-examined. Remapping server drives is not considered a best practice for XenApp environments because:

- Drive remapping is no longer an option in XenApp 5 running on Windows 2008 Server.
- Giving users access to their local C and D drives within a XenApp session is considered a security risk. If the applications are hosted on the XenApp servers, the data should be in the data center for best application performance.

Web Interface
In many scenarios, a site’s Web Interface servers are identical except for system-level settings like server name and IP address, which makes the Web Interface server a candidate to provision with Provisioning Services. Because Web Interface is the users’ main point of presence into a XenApp environment, it is critical--regardless of server--that Web Interface provides the same responses to users. To properly provision Web Interface, a common base Web Interface role should be created with all pertinent consistent configurations. This will help keep all Web Interface servers synchronized and help protect the server as changes will not be saved during reboots.

However, for high-security environments, provisioning Web Interface does bring about a best practices that should be taken into account:

- Certificates: The common image for the Web Interface must contain a server certificate if communication between the end-point -> Web Interface or Access Gateway -> Web Interface is to be encrypted. As multiple servers can be delivered from a single image, certificates pose a challenge that can be solved by:
  - Installing multiple server certificates on the Web Interface vDisk. Each server certificate will correspond to a potential server name defined within Provisioning Services. If there will be three Web Interface servers, then the vDisk should contain all three server certificates.
  - Using a wildcard certificate that can be used across all provisioned Web Interface servers. The wildcard will dictate the need for a common naming standard for the Web Interface servers.

**Data Collectors**

The data collector is responsible for application enumeration and designating the appropriate server to host user connections. Just like Web Interface, organizations often dedicate a primary and potentially a secondary data collector for fault tolerance. Unless users are logging on, logging off or disconnecting, the data collector remains fairly unused. Configuration differences between a primary and secondary data collector are very minor. By provisioning the data collector, a single role could be used for all data collectors.

Even though data collector provisioning is possible and recommended, the following are areas that should be taken into account:

- **Real Time Metrics**: When servers are provisioned, the Resource Manager (XenApp 4.5) real-time metric information is lost upon reboot because a common image is used. Resource Manager summarizes this information on the local server daily.

- **STA**: The integration utility automatically makes the STA unique because MAC addresses are unique. However, if the MAC address changes because the vDisk is streamed to a new physical/virtual server, the STA on the provisioned server will also change. To avoid disruption, the STA settings within Access Gateway must be updated to contain the latest STA ID value.

- **Election Preferences**: One important difference between multiple data collectors is the election preference levels. The primary data collector should be set to Most Preferred and the backup data collector should be set to Preferred. Even though these two servers will be delivered from the same vDisk, these changes will be kept between reboots because this information is stored within the XenApp data store.

**Application Tuning**

Every XenApp server provisioned with Provisioning services has the exact same registry settings. This is one of the core benefits of Provisioning services for XenApp… server consistency. However, certain applications experience issues with this type of a configuration, although the number of applications is minimal. The following subsections offers options for dealing with these types of applications so they function appropriately with Provisioning Services.

- **Machine Specific Registry Keys**
Machine Specific Registry Keys

A few applications rely on machine specific identifications in order to function properly. Typically, these applications are, but it is not limited to, systems management software. Incorporating the application within a provisioned XenApp server, either incorporated into the vDisk or delivered via XenApp Application streaming results in the application failing because another instance is running on the network using the same ID. However, these applications can still function by using one of the recommended options:

- ID Reset: Third party tools are available that can reset the machine-based ID after the server has booted. A startup script can be executed during server startup, which will dynamically create a new ID for the application. Even though the ID does not persist between server reboots, the script executes each time the server starts and completes in an extremely short amount of time. The script can either be incorporated into the vDisk image or applied to a set of servers by using an Active Directory Group Policy specifying a machine startup script.

- Personality: Provisioning services includes a personality feature allowing unique settings to be passed to the target device during startup. When applied to a target device within the Provisioning services console, the personality settings are delivered to the running server as a text file on the root folder. A script, integrated into the vDisk, can read this information and populate the information into the registry, file system or anywhere else. This allows for the creation of unique but static information that persists between server reboots.

- Differential Disks: Provisioning services allows many XenApp servers to be based on the same vDisk image, but still retain settings upon server reboots. By using a differential disk, the unique machine ID can persist between server reboots, unless the base vDisk image is modified. Although this is an option, the first two are the preferred methods.

Maintenance

With a single image being used across numerous end points, processes must be followed in order to properly integrate new updates without impacting users or systems. The following section focus on simplifying maintenance of images.

Automatic Updating of vDisks

Manually updating vDisks is not a difficult task. However, if done on a regular basis (daily, weekly, monthly) on one or more vDisks, the time required to maintain vDisks can quickly balloon to problematic levels. The Provisioning Server 5.0/5.1 admin guide (page 109) gives detailed instructions on how to perform the process manually. On a high level, the process goes as follows:

- Load a machine with a copy of the production vDisk in private mode
- Make updates
- Shut down and put the vDisk into standard mode
- Increment the version number

However, with Workflow Studio, the process can be automated to be entirely hands-free antivirus updates and operating system patches. Workflow Studio (WFS) is designed to reduce repetitive tasks into easy-to-manage workflows that can be run either on-demand or on a scheduled basis. Using Workflow Studio and Provisioning services built-in CLI, a script can be used to automate the entire process of updating vDisks, allowing for easy nightly or weekly scheduling with less chance for human error. An example script can be found at the Citrix Developer Network, which can be utilized as-is or modified to meet the needs of most deployments. The script can also be used without requiring WFS, but some additional functionality will need to be implemented in order to replace WFS’s capabilities. Please see the script for additional information.
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Change Description</th>
<th>Updated By</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Initial document created</td>
<td>Daniel Feller – Sr. Architect</td>
<td>February 17, 2009</td>
</tr>
<tr>
<td>1.0</td>
<td>Document finalized and released</td>
<td>Daniel Feller – Sr. Architect</td>
<td>March 11, 2009</td>
</tr>
<tr>
<td>1.1</td>
<td>Updated App Cache and Event Logs sections</td>
<td>Daniel Feller – Lead Architect</td>
<td>June 10, 2009</td>
</tr>
<tr>
<td></td>
<td>Added Machine ID section</td>
<td>Michael Bogobowicz – Sr. Consultant</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Added Automating vDisks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>