App-V Deploy and Publish

Tools from TMurgent Technologies
Updated Aug 5, 2010 – Version 1.8

Introduction:

The deployment of Virtual Applications in the simplest way possible, without the need for complicated backend infrastructures, has been a goal of mine for App-v for quite some time.

While the App-V runtime is a fantastic technology enabling the virtualization of application packages, and combined with the full (dedicated) back-end infrastructure it has the most feature complete implementation for application virtualization out there. But what I looked forward to with the acquisition of Softricity by Microsoft was componentization of the product. Microsoft has to a large extent broken some of the technology apart, and offers three distinct ways to deploy virtual applications. Yet none of these three – especially on Terminal Servers – matches the simplicity and feature set that I think Enterprises desire. We want to fix that.

This document describes the free version of a set of tools TMurgent is making available for enterprises, AppV_DeployApp, AppV_DeployContent, and AppV_PublishApp. These two tools enable the deployment of applications virtualized using App-V 4.5, and above, in a new and simple way. This deployment method is an alternative to the three main ways that Microsoft provides for virtual application deployment, the App-V Server using RTSP/S, SCCM, or using MSIs. This method allows for more flexible deployment and features than the last two methods, and much less overhead (in the form of back-end servers) than the first two.

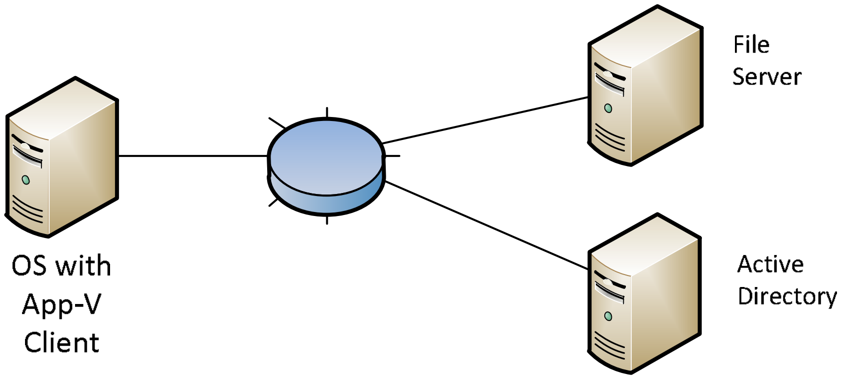
These tools address the main pain points of Terminal Server deployment of App-V. Fully scriptable, these tools are the simplest way to administer your virtual applications, while leveraging your organizations ability to perform application assignment using Active Directory. These tools also have applicability to desktop deployments, especially in a VDI style deployment.

These free tools should be sufficient to conduct a new large scale App-V deployment. Use of the free tools requires adhering to a fairly simple and straightforward naming scheme to tie the package folders on the file share to that of the authorization groups in Active Directory. Recognizing that companies with existing deployments or unusual needs may not want to revisit the naming they have used in the past, TMurgent can offer a customized version of the tools for a fee.

The free tools may be downloaded from the tools section at www.tmurgent.com . The website also has a 9 minute video demonstrating use of these tools.

Purpose:

These tools are for use on App-V Client machines. The primary intent is to provide a new way to deploy virtual applications on shared use computers, such as Terminal Servers or certain VDI and HotDesking scenarios, without the use of either the App-V dedicated server or SCCM. This simplified way of providing virtual applications only requires a very minimal back end; a file share for deployment and updates, and active directory for per-user publishing.



The App-V clients are configured for “standalone client mode with updates”. The central file share to hold the packages is the same as would be used by the App-V Management Server, or SCCM Site Server. It is a simple file share with folders storing the virtual applications as output by the App-V Sequencer. There is no need to create the optional App-V package MSI, nor is there any need to use a management console to configure the packages. Just copy the files onto the share, configured with read-access by users.

We also use a designated Active Directory Organizational Unit containing groups matching the names of the packages. These “application groups” are used by the publishing tool to determine which users should have shortcuts published to them.

The tools use client side registry settings to store references to these locations. These registry settings are established on install, but may be pushed out using Group Policy Preferences also. Adding the publishing tool to the user’s login script (possibly via Group Policy) is also recommended.

Note: Customers using other publishing tools, such as Citrix XenApp, Quest, or RES, will find value in using the deployment tools of this package to get the virtual applications deployed, and then use their existing publishing tools to make the packages available to users, instead of publishing tool in this package. Active Directory is not needed for the deployment tools of this package, only the file share. XenApp customers are also directed to the “XenApp Publishing Extensions 2.0” tools that TMurgent released on the Citrix Developer Network “Code Share” program for a better way to publish App-V applications to a XenApp farm.

Using these tools, and this simple back-end, you can achieve application deployment with the following features:

- Per-user virtual application package publishing
- “Active updates”

Overview:

Providing users virtual applications involves two steps, deployment and publishing of access (shortcuts and file type associations). Because of the existing tools available, we tend to think of deployment and publishing as a single step. These tools are specifically designed for each of the two steps.

- **Deployment**: Deployment involves configuring the App-V Client system for the existence of one or more virtual application packages, and fully pre-loading into the App-V sft cache.

- **Publishing**: Publishing involves the placement of application shortcuts and file type associations necessary for a user to run the virtual applications in the package.

There are three tools in this toolset. Two are for deployment, and one for publishing.

- **AppV_DeployApp** is a tool for deploying a single virtual application package.
- **AppV_DeployContent** is a tool for deploying all virtual application packages stored under a referenced folder.
- **AppV_PublishApp** is a tool for publishing the shortcuts and file type associations for deployed packages that are assigned to this user in Active Directory.

The AppV_DeployApp and AppV_DeployContent tools also perform publishing to the user account that performs the deployment, to ease testing. These tools also have an option to publish to all user accounts on the client in this step, mimicking the behavior of the MSI or SCCM deployments today.

The deployment tools have a GUI, but may also be run silently using command line parameters, making them fully scriptable. The Publishing tool is a command line tool intended to be part of a user's logon script. It has a debugging option to output to a log file for testing.

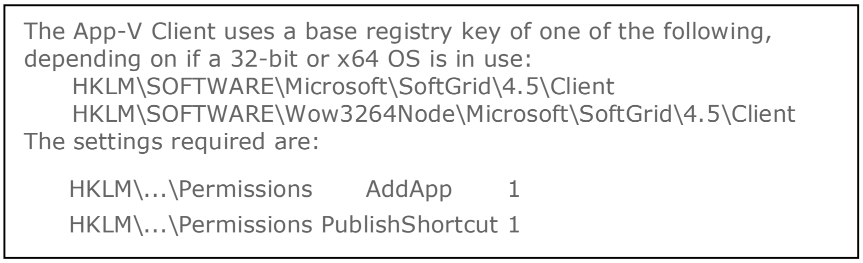
Setup/Installation:

First, set up the App-V Client in "stand-along mode" with updates. See the TMurgent White Paper "***Microsoft App-V 4.5 Client in Stand-Alone Mode***" for details regarding installing the client for stand-alone with streaming mode (www.tmurgent.com/WhitePapers/Microsoft_AppV_Stand-Alone.pdf). As a short summary, the following should be set:

```
The App-V Client uses a base registry key of one of the following,
depending on if a 32-bit or x64 OS is in use:
    HKLM\SOFTWARE\Microsoft\SoftGrid\4.5\Client
    HKLM\SOFTWARE\Wow3264Node\Microsoft\SoftGrid\4.5\Client
The settings required are:
   HKLM\...\Configuration AllowIndependentFileStreaming 1
   HKLM\...\Configuration RequireAuthorizationIfCached    0
   HKLM\...\Network        AllowDisconnectedOperation     1
   HKLM\...\Network        LimitedDisconnectedOperation   0
```

In addition to this installation instructions, in order to achieve per-user publishing, permissions must also be granted in the App-V client to allow users to publish applications. This permission is not normally set.

These permission settings are shown below:



```
The App-V Client uses a base registry key of one of the following,
depending on if a 32-bit or x64 OS is in use:
    HKLM\SOFTWARE\Microsoft\SoftGrid\4.5\Client
    HKLM\SOFTWARE\Wow3264Node\Microsoft\SoftGrid\4.5\Client
The settings required are:

    HKLM\...\Permissions      AddApp      1
    HKLM\...\Permissions PublishShortcut 1
```

A sample script (AppV_ClientSetup_For_AppV_PublishApp.cmd) suitable for a silent install and configuration of the App-V Client is included in the download package for these tools. Keep in mind that although the provided script is a good base, this script likely requires modification for your environment.

Next, run the installer for this toolset. Version 1.6 now has an installer. The setup.exe will ensure proper pre-requisites are met and then install the tools using the external setup.msi file. To install, the installation wizard will ask the following:

- Which tools should be installed.
- UNC path to present as the default folder in the deployment tool GUIs.
- Active Directory Organizational Unit to contain application package groups.
- Directory for installation.

Note that adding AppV_PublishApp to the user logon script must be manually performed.

Content Folder

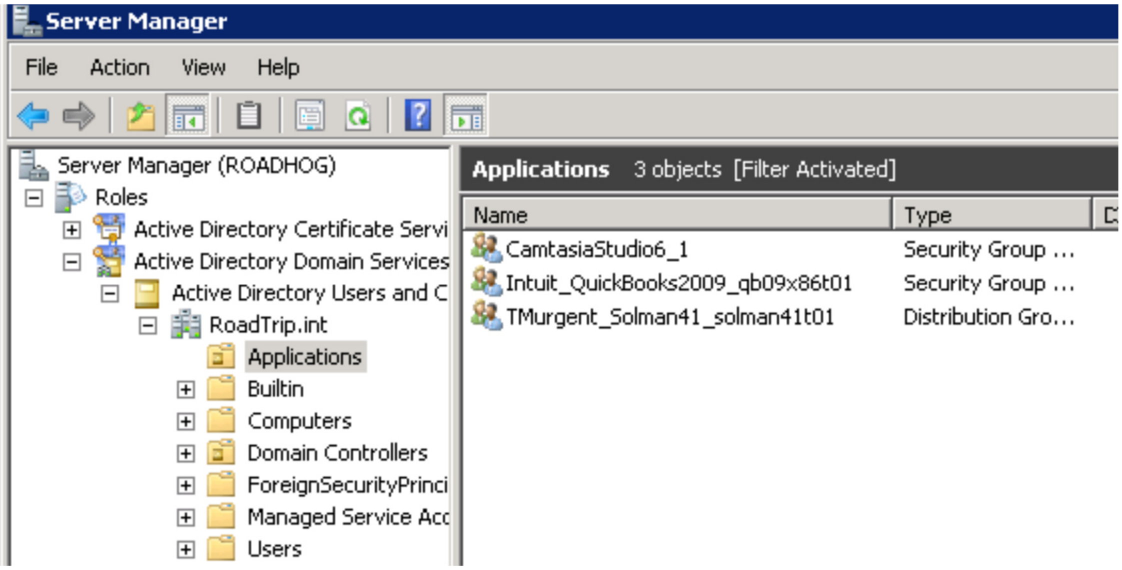
Ultimately, you need a file share to hold the virtual application assets. Organization of the packages under this share may use any scheme desired. It is usually advisable that each package be stored in its own folder, but additional organizational structuring may also be performed. Unlike when using the App-V Management Server, this storage structure does not need to match the “PATH” entry configured for the package when sequencing. To make it easier to manage, we recommend that packages are stored in a folder using the name of the package. Thus, for an example scenario,
we might find the following layout for three packages:

```
\\roadhog\Content
        CamtasiaStudio6_1\
        CamtasiaStudio6_1_manifest.xml
        CamtasiaStudio6_1.sft
        Intuit_QuickBooks2009_qb09x86t01\
        Intuit_QuickBooks2009_qb09x86t01_manifest.xml
        Intuit_QuickBooks2009_qb09x86t01.sft
        TMurgent_Solman41_solman4t01\
        TMurgent_Solman41_solman4t01_manifest.xml
        TMurgent_Solman41_solman4t01.sft
        TMurgent_Solman41_solman4t01_2.sft
```

Note: When multiple sft files exist in a folder, the file date of the files will be used to determine the newest sft in the folder.

Active Directory

To use the publishing tool, you will need an Organizational Unit (OU) in Active Directory that will provide a means to authorize users and groups of users the right to use the applications in a package. The OU might look like this example, (taken from a Server 2008 setup):



- Server Manager (ROADHOG)
  - Roles
    - Active Directory Certificate Servi
    - Active Directory Domain Services
      - Active Directory Users and C
        - RoadTrip.int
          - Applications
          - Builtin
          - Computers
          - Domain Controllers
          - ForeignSecurityPrinci
          - Managed Service Acc
          - Users

Applications 3 objects [Filter Activated]

Name	Type
CamtasiaStudio6_1	Security Group ...
Intuit_QuickBooks2009_qb09x86t01	Security Group ...
TMurgent_Solman41_solman41t01	Distribution Gro...

The groups may be either security or distribution groups. The names of the groups must exactly match both the content folder name and the package name.

As users, or groups of users, are assigned as members of these groups, they will be permitted to have the applications in the package published. Members of these groups may either be individual members, or other groups that exist in the domain.

In the free version of the tools, publishing is done only on a per-package basis, not per-application, and the group name must exactly match that of the package name.

Administrator Roles

While the Deployment aspects above, short of assigning users, would be carried out by a more central IT administrator, typically many administrators will have the need to assign users to applications. Because this is a simple operation using Active Directory, they already have the skills to perform those actions.

Even when more complicated systems, like the App-V Server provide their own assignment mechanisms, we see over and over again that enterprises by-pass those mechanisms and use Active Directory in a very similar way.

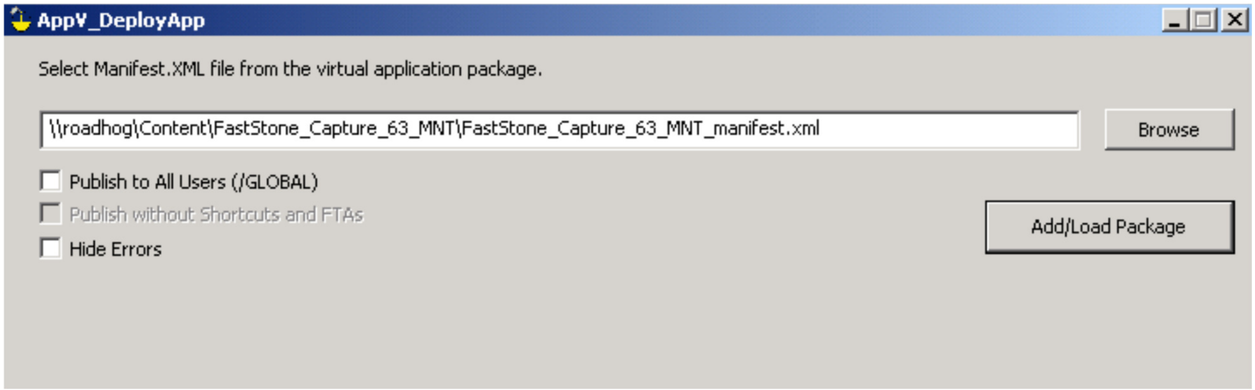
Administrator's Job to Deploy Packages

After the setup, the administrator logs into the terminal server using an Admin Account.

Using one of two tools, the administrator will deploy the applications to the PC, filling the cache, but not yet publishing to the user. The AppV_DeployContent tool can be used to point to a unc folder, such as the content share folder itself, and it will walk the directory structure searching all packages with a valid manifest file to deploy. Alternatively, AppV_DeployApp can be used to deploy a single package. Either tool may be scripted (see the “About” help of the GUI for command line details).

Using AppV_DeployApp from the GUI

1. Launch AppV_DeployApp.
2. Browse to the content folder desired and select the manifest.xml file.
3. Ensure the “Publish to All Users (/Global) checkbox is unchecked.



AppV_DeployApp

Select Manifest.XML file from the virtual application package.

\\roadhog\Content\FastStone_Capture_63_MNT\FastStone_Capture_63_MNT_manifest.xml

□ Publish to All Users (/GLOBAL)

□ Publish without Shortcuts and FTAs

□ Hide Errors

4. Click Add/Load package.

This will perform the same actions that the App-V MSI would have done, using sftmime to add the package and then load the sft, except:

- It will be set for streaming (for updates)
- No /global flag (unless selected)
- No Add/Remove programs

At this time, the administrator can run and test the package on the terminal server, but nobody else can.

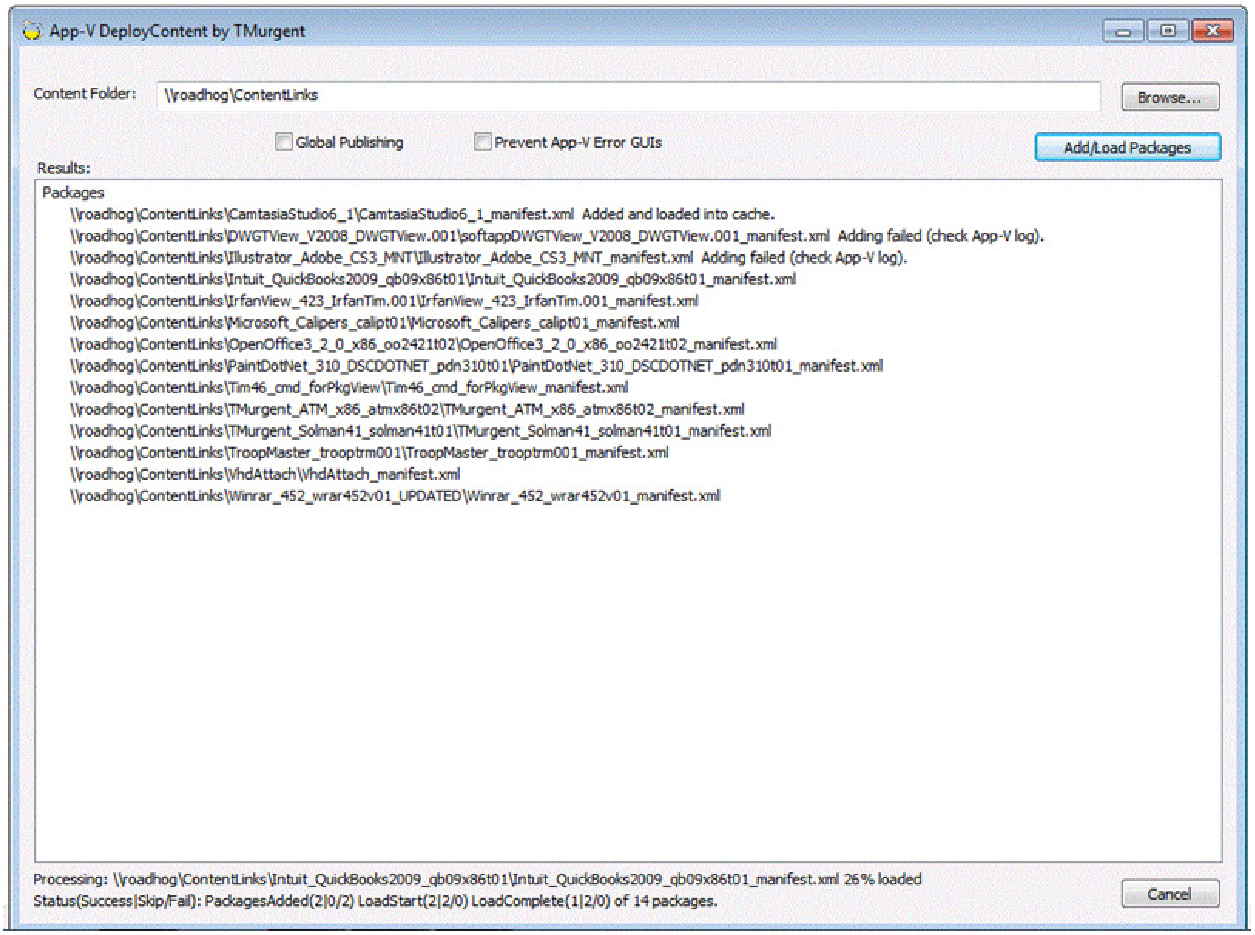
Using AppV_DeployApp from Command Line

AppV_DeployApp can be run without the GUI from the command line, making it scriptable also. Run the command “AppV_DeployApp.exe /?” in a command prompt for latest information on the command syntax.

Using AppV_DeployContent

As an alternative to AppV_DeployApp, you may also use the AppV_DeployContent tool. This tool will search through a folder structure to deploy all of the packages it finds. Shown next is a screen shot of this tool in action:

(Note that the screen shot, two packages failed to load. The reason was that we were deploying on Windows 7 and those packages were marked to only work on Windows XP. This is an expected failure which is easily diagnosed in the App-V event log).



Content Folder: \\roadhog\ContentLinks

□ Global Publishing □ Prevent App-V Error GUIs

Results:

```
Packages
    \\roadhog\ContentLinks\CamtasiaStudio6_1\CamtasiaStudio6_1_manifest.xml  Added and loaded into cache.
    \\roadhog\ContentLinks\DWGTView_V2008_DWGTView.001\softappDWGTView_V2008_DWGTView.001_manifest.xml  Adding failed (check App-V log).
    \\roadhog\ContentLinks\Illustrator_Adobe_CS3_MNT\Illustrator_Adobe_CS3_MNT_manifest.xml  Adding failed (check App-V log).
    \\roadhog\ContentLinks\Intuit_QuickBooks2009_qb09x86t01\Intuit_QuickBooks2009_qb09x86t01_manifest.xml
    \\roadhog\ContentLinks\IrfanView_423_IrfanTim.001\IrfanView_423_IrfanTim.001_manifest.xml
    \\roadhog\ContentLinks\Microsoft_Calipers_calipt01\Microsoft_Calipers_calipt01_manifest.xml
    \\roadhog\ContentLinks\OpenOffice3_2_0_x86_oo2421t02\OpenOffice3_2_0_x86_oo2421t02_manifest.xml
    \\roadhog\ContentLinks\PaintDotNet_310_DSCDOTNET_pdn310t01\PaintDotNet_310_DSCDOTNET_pdn310t01_manifest.xml
    \\roadhog\ContentLinks\Tim46_cmd_forPkgView\Tim46_cmd_forPkgView_manifest.xml
    \\roadhog\ContentLinks\TMurgent_ATM_x86_atmx86t02\TMurgent_ATM_x86_atmx86t02_manifest.xml
    \\roadhog\ContentLinks\TMurgent_Solman41_solman41t01\TMurgent_Solman41_solman41t01_manifest.xml
    \\roadhog\ContentLinks\TroopMaster_trooptrm001\TroopMaster_trooptrm001_manifest.xml
    \\roadhog\ContentLinks\VhdAttach\VhdAttach_manifest.xml
    \\roadhog\ContentLinks\Winrar_452_wrar452v01_UPDATED\Winrar_452_wrar452v01_manifest.xml
```

```
Processing: \\roadhog\ContentLinks\Intuit_QuickBooks2009_qb09x86t01\Intuit_QuickBooks2009_qb09x86t01_manifest.xml 26% loaded
Status(Success|Skip/Fail): PackagesAdded(2|0/2) LoadStart(2|2/0) LoadComplete(1|2/0) of 14 packages.
```

Add Publish to User's Login Scripts

Just add AppV_PublishApp.exe (no arguments) to the login script of users. The user will have all appropriate shortcuts and file type associations added to their account each time they log in. It is trivial to set up this scripting using Group Policy.

It is that simple. Should anything ever go wrong, errors from SFTMIME will be logged as normal App-V errors, so check the event log for issues. You can also manually run AppV_PublishApp /D from a command prompt in the user's login to see rich details of the processing steps to further isolate any mistakes made in deployment. Try "AppV_PublishApp /?" for more on the syntax of the command.

Removals of packages are not supported by the tool at this time. Use the App-V Client Management Console to remove packages and applications.

Regarding Updates to Packages

One of the benefits of the stand-alone client with updates is the ability to centrally update a package and have clients receive the update automatically when the application is next run. And this is supported when using these tools. There are some things to consider, if you want to automatically update previously deployed packages.

- When the original package is deployed to the client, the location of the source sft was recorded by the client in the App-v portion of the local machine registry. When a client starts an application, the source sft file location is checked to see if the file has changed, using the file datetime-stamp.
- If the sft has changed, the difference will be streamed to the client.
- On the App-V 4.6 TS Client, the difference is only streamed if no users are currently using that package on this terminal server. This has a tendency to delay implementation by a day on some applications.
- The default in the sequencer for updating a package for updates (as opposed to branching) is for the new sft to have a version number appended to the name. Although our deployment tool will detect the newer SFT automatically for new deployments, the App-V client will not detect the new sft is the file name and location is not identical. We recommend one of two solutions:

    1. Rename the SFT and copy over the old one (preferably during a maintenance window).
    2. If Server 2008 R2 or Windows 7 is in use for the content share and client, the use of a secondary share using "soft links" is recommended. In this case, the upgraded package is placed in a new content folder (instead of overwriting), and the link is changed. The file name of the new sft must still match, but at least the link may be changed at any time.

  Note that the sequencer has a configuration option to not modify the SFT filename when an upgrade is created.

Because it is often difficult to determine when a package is not currently in use, when Server 2008 is used for the file share, we recommend deploying using a secondary "shadow share", rather than having the clients reference the content share directly. This secondary file share does not contain a copy of the packages, but rather a set of "symbolic links" (created using the mklink command) to their counterpart folders in the content share. It is completely safe to copy the updated package into a new folder in the content share, and then modify the symbolic link to point to the new folder. Performed this way, the update will have no effect on clients currently running the package. If the "shadow share" is not used, you should only update the packages during a maintenance window.

Regarding Metering Usage

Without the server, there is no standard way to monitor application usage. But just as in the case of SCCM deployment, if you need this you can make use of the Clients WMI Provider or Remote Registry capabilities in a tool such as System Center Operations Manager. The most detailed reporting could also be accomplished via event log trolling, however we have never seen a customer do that.