SCCM 2012 APPLICATION DEPLOYMENT METHODOLOGY BEST PRACTICE
Application Deployment Methodology Best Practice

By Shawn Cardamon

Contents

Introduction ............................................................................................................................................. 3
New Deployment Options in 2012 ........................................................................................................ 4
Be Careful What You Wish For .......................................................................................................... 5
Focusing On Use-Case .......................................................................................................................... 6
Available Yet Required ....................................................................................................................... 6
What If.................................................................................................................................................. 7
Conclusion ............................................................................................................................................ 7
Introduction

This whitepaper is intended to share knowledge gained from the field, utilizing the Application Management model within System Center Configuration Manager 2012 (ConfigMgr). The main focus will be placed on deployment availability and how that is managed within the software lifecycle management tools provided in Configuration Manager 2012.

In providing an understanding of what the long-term effects deployment availability options have on an enterprise environment, we can begin to assess tactical options open to us. Focus will be held tightly on one seemingly simple option within the Deploy Software Wizard. This option being “Required” or “Available”: 
New Deployment Options in 2012

With Configuration Manager 2012 came the birth of Applications aside from legacy Package/Program and with it, a significant change in how IT organizations manage their managed application catalog. Applications are now more of the focus when addressing the needs of software distribution rather than Configuration Manager being a vehicle for sending a “configuration” to a target device.

Microsoft has done a fantastic job of separating out the needs of the direct configuration manager and application manager between the Applications and Configuration Settings modules. These two tools can help you deploy software in a state-based manner with the introduction of Deployment Types. You can also configure business specific custom application needs, post installation, via Configuration Baselines. This strengthens ability to support applications within an IT enterprise as in most cases administrators are able to provision applications to managed devices using the standard vendor configuration. Then user-targeted and/or machine-targeted configurations may be enforced post install. This is very important when working directly with software vendor support as we all know from past experience this is the first major hurdle to overcome. This hurdle being, “did you customize the installation in any way?” then spending hours proving your simple customization has nothing to do with an application issue.

These new strengths along with the ability to provide pre-requisite applications (even further enabling maintaining standard application content for delivery), supersedence, and proper retirement construct takes Configuration Manager 2012 light-years ahead of where Configuration Manager 2007 left off.

Apart from the provisioning enhancements, we have a new approach to deployment proper. Users are now a big part of the deployment process where in the past, administrators had to rely on Active Directory discovery population within the database to provide materials for WQL querying of UserID data and correlating to machine inventoried Top Console or Last Logged On information. Configuration Manager 2012 now manages user specific inventory apart from machine inventory and linkages are native to the deployment tools provided via the console.

So now that we know this, there is a paradigm change to be addressed. Where once the Configuration Manager toolkit provided a set number of reliable avenues for targeting software to devices (and many avenues of which we all used but since regretted), we now get Application Management which goes so much further than Configuration Manager 2007 did but with all these new shiny features, we have a bit of room to get lost.

I have been working with this new model since the V-Next days where I was lucky enough to work with the Configuration Manager 2012 development team alongside colleagues during the Technology Adoption Program (TAP). So many options were open to us from a software distribution prospective, where it now reached over into the software lifecycle management side of things, and this was intentional. Microsoft looked at this new model the right way, ensuring applications are managed as a catalog, providing dynamic targeting and methods to present applications to end users. No more ready, aim, fire – we now have Configuration Manager truly living up to its name in this space as you can set up an application with line of sight to most all management configuration.

This is a lot to get your head around and transpose into a reliable, consistent, and repeatable set of operational processes. Depending on the demands of the business within your organization, you may utilize the Applications node a bit differently than others, which is the nature of Configuration Manager to a certain degree. What this white paper aims to provide is best practice when approaching deployment and availability.
Be Careful What You Wish For

We have the tools to drive greater business value and ensure application presence is pervasive and can even follow users wherever they go if there is a business need. This is fantastic for core critical applications where consistent availability of tools is very important but what about when the application was needed but when needs change and it is no longer used? Is its presence still core critical? Additionally, for most environments, core critical applications only make up a fraction of the managed application catalog.

A great example of these two types of applications would be comparing Microsoft Office which the business has stated must be on all devices and ELA licensing supports this demand with Adobe Acrobat Standard, which is a per seat license and only required as needed. An application which is noted as needed by many business units but the use case is often not one of being needed all the time to guarantee productivity. However we need to ensure those who need it, have it and that use case can, and will change.

With both examples, we have capability within the toolset to setup software within the Applications node and ensure their configuration is managed for each use case within Configuration Settings along with deploying as “Required” to all applicable devices. This ensures high availability ongoing, even if somehow MS Office gets removed, Application Enforcement will ensure the application is again installed. But what about Acrobat?

Since I wanted to ensure the proper targets have this application without forcing end users to approach Software Center (or the Application Catalog if user targeted) to obtain this title. I have deployed this application with my only option for distributing to identified targets, which is “Required” to meet with business-generated deployment requirements. This was done because the business is used to Configuration Manager 2007 where the end user productivity impact of searching for an application in one of two places did not exist. The concept of a consumerized experience was not desired by the business and, therefore, they did not want any end-user involvement/time consumption.

Now that I have set up Acrobat to deploy in this manner. In both cases, I have to rely on my support organization for basic remediation of application issues. Here, I cannot provide an option to my end users which gives them the ability to uninstall and directly reinstall the application from Software Center should issues arise, we called this “self-remediation”.

After this main realization, we start to see a number of other challenges, which include:

- Will all support representatives requested by end users to remove or upgrade an application be required to have console access to alter targeting?
- Will all applications need to be targeted in a way where individual targets may be removed in order to ensure application enforcement does not reinstall applications where not desired, if so, how do we manage this selective targeting? Additionally if we do allow for removal or use of “Exclusions”, how do we ensure retargeting is successful?
- Do I need to grant end users the ability to uninstall applications within Windows so they may self-remediate, and is expecting an end user to trigger the evaluation function or waiting for the next interval acceptable to the business? Is this acceptable when it comes to other apps?
- How do I handle an initial deployment of a software title which is not core critical in applications node which will just install without user interaction but can be removed later if need arises?
- How do I address software waste in this model?

There are more challenges than these listed above but I believe this paints a typical scenario. The Applications node is a fantastic advance forward but with this advancement comes some critical questions you need to ask yourself from an operational process and application lifecycle prospective.
Focusing On Use-Cases

Available Yet Required
So aside from the main construct discussed, there is the topic of application deployment and ensuring proper availability. To manage these tasks, a fundamental question must be posed to the business or demand generator for the software in question. “Is this application a core critical application driving end user productivity?”

This question of persistent targeting long term should also be qualified with some basic logic. If demand is calling out for an application to be core critical, then this demand is also qualifying that:

- Any new device or user which qualifies for persistent targeting must have this application at all times in order to be fully productive
- Entitlement model supports target audience and projected growth
- Support and CM administrator must be engaged should this application require removal
- No matter if the application is used regularly or not, it should never be removed from the device.

If the application is detected as removed, it will need to be reinstalled (within an SLA which meets with client settings for Application Eval).

If all these conditions are present then the base Application composition and deployment targeting with install behavior of “Required” is what you are looking for as this deployment will be state based. In other words, this will be treated like a WSUS deployment.

Should any of the questions above result in anything other than an unequivocal yes, then another option is required. In most all interactions with the business, an initial rollout period will be desired and users searching Software Center/Application Catalog will many times not be part of that desire since this consumes end user productivity time.

Additionally, if the above-mentioned model is used to deploy software within your environment which does not meet this base definition of core critical, software waste is all but guaranteed. Software will be deployed and enforced whether it is used or not permanently.

There are a few ways to ensure non-core critical applications get the core-critical deployment treatment while clearing the way for Software License Optimization efforts and end-user self-remediation without leaching away productivity. This can be achieved through a simple process of altering deployment availability at the right time or adding query logic. In this paper, we will speak to timing.

To best describe this process we will take on a set of deployment requirements for Adobe Acrobat Standard:

- All users of Adobe Acrobat Professional are supplied Standard as a mandatory deployment
- End users need the ability to self-remediate the application as plugins with home-grown applications is problematic with high support call volumes and resolution simply involves uninstall/reinstall
- After initial deployment this application needs to be available for on-demand installation going forward
- If software is identified as not used, it can be uninstalled/reclaimed for license transferral or reduction in business demand.

To achieve this setup the answer is simple. Initial deployment efforts are set to “Required” and, once deployment activities have concluded to business satisfaction, a single new deployment is established where install behavior is set to “Available” leveraging the same principal target audience and all initial deployment(s) are deleted. It is also required that the uninstall
configuration is populated within the application setup to meet the abovementioned requirement set.

This does mean that deployment reporting will be removed once this action is taken so if there is a need to archive this data, you will need to extract this information pre-deletion. The instituting of an “Available” deployment will still report on which targets have the application installed and will call out where it is available once devices evaluate policy just as the “Required” deployment did.

What If...
Any time a best practice is suggested, no matter how complex, a number of repercussions can exist when looking at end to end management of an application. The following is a list of potential caveats to this practice and how they have been addressed:

- Applications which are set to available but desired as part of a base OS load but long term it should not be set to “Required” to avoid software waste - this can be addressed via establishing an additional deployment which is set to “Required” with the proper target audience with an addition of a query which will only target devices which have been loaded in the last 30 days. That is if you are not just calling for direct installation within a Task Sequence, but in the interest of keeping OSD content light, this is an option.
- If a user uninstalls the application, how can they get it back? Native to the “Available” model (when uninstall information is populated in the Application) is the ability to install again after uninstalling.
- How are deployment issues handled where clients are not able to install during initial deployment phase? This can be handled by the end-user adoption of Application Management in Configuration Manager 2012. If this process is used and end users are encouraged to search for needed software in an on-demand fashion, they will know to use Software Center and the Application Catalog as a pre-support landing to see what is self-addressable.

Conclusion

This approach has been devised though personal approach to Application Management in Configuration Manager 2012 and engaging many other Configuration Manager administrators who are regularly challenged by the business to deliver IT value in a way which meets with demand. This is not the only way to handle deployment and availability within Application Management but is thus far best practice when looking to ensure Applications are properly provisioned with the entire software lifecycle in mind.