Chapter No. 8

"Integrating App-V with System Center Configuration Manager 2012"
In this package, you will find:
A Biography of the author of the book
A preview chapter from the book, Chapter NO.8 "Integrating App-V with System Center Configuration Manager 2012"
A synopsis of the book’s content
Information on where to buy this book

About the Author

Augusto Alvarez is a computer geek and a fan of the latest IT platform solutions. He has been working with Microsoft TechNet and Microsoft Academics since 2006, collaborating on different technological events.

He is currently working as a Sr. System Integration Consultant in Dell LATAM where he is in charge IT services delivery and quality assurance directly to the datacenter, deploying multi-vendor technologies and solutions within Microsoft, Citrix, VMWare and Open Source. These solutions deal with the major business processes used to manage the overall enterprise. The tasks involved in his work usually requires integrating hardware, processes, methodologies, software, and the customer.

He has vast experience not only in App-V, but also in several Microsoft technologies for Windows Server: deployment, Active Directory, Hyper-V, virtualization solutions, clustering, System Center suite, security implementations, and so on.

For More Information:

You can find out more info about him in his blog: http://blog.augustoalvarez.com.ar/ or following him on Twitter: @augustoalvarez.
Microsoft Application Virtualization Advanced Guide

Microsoft’s Application Virtualization technology has been emerging as one of the main platforms in application virtualization. This is mainly because more and more organizations are thinking about application virtualization as a valid implementation and also because App-V probably represents the most attractive solution. App-V introduces a new and efficient way to improve operating systems and application life cycles in organizations. However, if we want to maximize our investment, we must learn that every scenario and application differs from another, as does complexity.

With the Microsoft Application Virtualization Advanced Guide, administrators will find detailed topics for understanding App-V components, architecture, and implementation models, as well as step-by-step guidance to master App-V processes. There is also guidance on integrating App-V with other robust platforms, such as VDI, SCCM 2012, and even virtualizing server applications.

The Microsoft Application Virtualization Advanced Guide will start by giving readers the means to discover unfamiliar topics in App-V, as well as understanding App-V’s architecture. By doing so, it will provide the baseline needed before shifting to advanced topics in App-V, such as complex scenarios for sequencing and deploying applications.

This book details scripting within App-V as well as handling the platform using the command line; there is a complete review of troubleshooting installations, sequencing, and deploying applications.

Integration is another key component that we will cover by introducing App-V in VDI and SCCM 2012. Server applications are not forgotten with Server App-V. Using this book, the reader will have a detailed understanding, with step-by-step instructions on how to virtualize server applications.

What This Book Covers

Chapter 1, Taking a Deep Dive into App-V: In this chapter, we will review App-V architecture and detailed components as well as some advanced options available in the App-V environment.

Chapter 2, Sequencing in Complex Environments: In this chapter, we will review detailed and advanced options available in the sequencing process, handling applications with unsupported configurations, and using package accelerators.

For More Information:
Chapter 3, Deploying Applications in Complex Environments: In this chapter, we will understand how to deploy applications in scenarios where it normally is not an easy process.

Chapter 4, Handling Scripting and App-V Command Lines: This chapter will be your guide to using scripting in OSD files as well as handling command lines available in App-V to automate all processes.

Chapter 5, Troubleshooting App-V: In this chapter, we will review all the scenarios to understand and solve problems within App-V installations and application sequencing, as well as deploying virtual packages.

Chapter 6, Scaling Up App-V Implementations: In this chapter, we will cover the scenarios for designing your environments on a large scale.

Chapter 7, Integrating App-V with Virtual Desktop Infrastructure (VDI): In this chapter, we will learn about VDI environments and how to integrate them with App-V. We will also use the shared cache feature.

Chapter 8, Integrating App-V with System Center Configuration Manager 2012: In this chapter, we will cover the new System Center Configuration Manager platform and how to integrate it with App-V.

Chapter 9, Integrating Server App-V with Private Clouds: In this chapter, we will review, the new concept appearing as Server App-V, as a possibility to improve our private cloud environment.

Appendix, Reviewing App-V Microsoft and Third-party Tools: In this chapter, we will take a look at the tools available from Microsoft and third-party vendors to complement and improve App-V scenarios.

For More Information:
Integrating App-V with System Center Configuration Manager 2012

Optimizing our infrastructure and operations is a concept that we have revisited several times within this book. System Center Configuration Manager (SCCM) is a platform vastly used in several organizations with the purpose of this optimization, and with its new version, we now have several improvements in management and operations.

App-V integration is also an important topic that we must consider, analyze, design, and implement, if we already have the Configuration Manager platform in place. If we choose App-V as the desired solution for handling the application's lifecycle, we must consider SCCM as an important part in this solution.

Even though these two platforms were designed not only for integration but also to be deployed as highly scalable platforms, there are some considerations that we should carefully review in the process. And since SCCM appears in the application's deployment process, we must understand some of its capabilities and how they differ from an App-V implementation without any coexistence.

With the analysis completed, we can easily evaluate it, as this integration is quite a powerful solution. SCCM 2012 and its new enhancements tremendously facilitate application management and enhance user experience, not only for the IT administrator but also the final end user.

For More Information:
In this chapter, we are going to cover the following:

- Understanding SCCM 2012 and its new functionalities
- Reviewing the benefits of, and considerations regarding, the SCCM+ App-V integration
- Naming the components involved in this integration and the application delivery methods present
- Implementing the SCCM + App-V integration (starting with the SCCM 2012 installation and how to distribute the App-V Client to clients)
- Creating and deploying virtual applications using SCCM 2012

**SCCM 2012**

*System Center Configuration Manager 2012* is the new version of this robust and scalable solution from Microsoft.

SCCM is in charge of defining your organization structure in order to improve most of our operational tasks in the IT platform, such as software deployment, compliance settings management, and comprehensive asset management of servers, desktops, laptops, and mobile devices.

The basic functionality of SCCM is to work with client agents (discovered initially and commonly using Active Directory), which are deployed to users’ devices. These components are then put in an inventory for both hardware and installed software. This data is stored in a central database, and that data is then used to target application deployments to collections of client computers or users.

Some of the common tasks we can perform and integrate with SCCM are software lifecycle management, operating systems deployment, patch management, guaranteeing clients security compliance by integrating it with the *Network Access Protection* (NAP) platform, and so on.

**SCCM 2012 improvements**

System Center Configuration Manager 2012 introduces significant changes to its predecessor, SCCM 2007 R3. To summarize, the main focus of the new features is *the user*, not only the administrator in charge of managing the platform. SCCM 2012 now contains a far simpler UI, and the final user has an improved interaction with the SCCM resources available.

For More Information:

Let’s review the main improvements in SCCM 2012:

- **Improved Management Console**: This feature is likely to surprise you the most, at first, since it has a fresh, new, and considerably different design, as compared to what we’ve seen in earlier versions.

  The Management Console is divided into four sections: **Administration, Software Library, Monitoring, and Assets and Compliance**. The new console also adds a nice-looking ribbon at the top with all the basic operations, and most of the basic tasks have been regrouped in order to facilitate usage.

  We can assure you that the new SCCM 2012 console will give administrators an incredibly improved experience.

- **Improved setup options**: The installation steps are configured to run with simplified options, but the main improvement here is the option **Site Recovery** included in Setup Wizard.

- **Including Software Center as the main end-user interaction interface**: SCCM 2012 includes the SCCM client in a portal where users can handle their resources, provided by Configuration Manager. Using **Software Center** users now have more control of the software they have available in all of its devices (laptops, desktop or smartphone). Within Software Center, users can also request new applications.

For More Information:

Integrating App-V with System Center Configuration Manager 2012

- **New and enhanced options for applications management**: There are several new features included in all the lifecycles of applications, for example, multiple deployment instances of the same application—one application can include different ways to deploy it—handling a virtual package for one device but full binaries in another device.

- **Retirement**: SCCM 2012 uses application retirement for all applications. This "retirement" option was only available for virtual applications in SCCM 2007 R2/R3.

- **Requirements and dependencies**: We can set different types of requirements to deploy an application, for example, number of processors available, memory, free disk space, existing registry key, and so on. We can now also use other applications as dependencies.

- **Supporting smartphones natively**: SCCM includes the possibility of managing smartphones in our organization, not only for the Windows Phone series but also Symbian and Nokia devices. This new feature (essential for several organizations) appeared basically because Microsoft offered, without much success, a previous version for smartphone management—System Center Mobile Device Manager. This platform is included in SCCM 2012, natively.

- **Infrastructure and roles improvements**: There are several changes in the infrastructure itself in SCCM 2012. It is now simpler than the previous version. Some of the changes are as follows:
  - Central Administration sites can't have any users assigned. This role is only used to manage replication and for management hierarchy tasks between sites.
  - Sites are no longer configured for mixed or native mode, instead you secure client communication endpoints by configuring individual site system roles to support client connections over HTTPS or HTTP.
  - Client agent settings are now defined at the collection level, instead of at the site level.
  - Content distribution is now the responsibility of SQL Server replication.

- **SCCM 2012 is a 64-bit only platform**: It can only run in Windows Server 2008 64-bit version or Windows Server 2008 R2. For the database instance, SQL Server 2008 or SQL Server 2008 R2 in 64-bit are supported (to be discussed later—the SP and update level of each).
The System Center Configuration Manager distribution points role can still run in 32-bit.

As we can see, there are tons of improvements in this new platform and several others that we don't have the chance to review just yet. For more information about these changes, please read What's New in Configuration Manager at http://technet.microsoft.com/en-us/library/gg699359.aspx.

Reviewing SCCM and App-V integration

Integrating the System Center Configuration Manager with App-V is much simpler with the new versions of these two platforms. Let's review some of the benefits, considerations, delivery methods, and components required.

Benefits of integrating SCCM and App-V

There are several benefits of using SCCM and App-V as an integrated platform instead of separate technologies. Let's review some of the most important ones:

- **Optimizing your infrastructure**: If we have already implemented SCCM in our environment, not integrating with App-V could translate into larger costs for management, troubleshooting, complexity, and hardware. Since we will need to implement the Streaming Server functionality separately from distribution points, this role in Configuration Manager can fulfill the streaming process without acquiring big changes in our implementation.

- **Improving client targeting**: System Center Configuration Manager brings you the possibility of deploying normal and virtual applications with an enhanced level of targeting, depending on collections and capabilities of the systems involved.
  
  There are several possibilities of "client targeting" we can include while deploying virtual applications. For example, dependencies on other applications and client requirements, such as memory available, processor type, and so on.

- **Complementing App-V with SCCM assessments**: App-V includes user targeting for their packages. By integrating with Configuration Manager, we can combine these possibilities with software metering, asset intelligence, and Wake-on-LAN features for the deployment of the virtual applications.

For More Information:

• **Virtual applications delivery as a complement in Operating System Deployment:** One of the most important features in SCCM is Operating System Deployment (OSD), which can be combined and scaled up with other features, such as Software Updates, software and hardware inventory, targeting for implementing operating system drivers, and so on. Using App-V, we can deliver applications as soon as the operating system is deployed, saving considerable time to deliver a ready-to-go operating system.

• **Background delivery of App-V applications:** In unstable or slow networks, BITS protocol can be leveraged, allowing application delivery as network connections permit. The SCCM client performs the download of the App-V application into the SCCM cache from where it is then imported into the App-V cache. This offers much more flexible application delivery but comes with a storage penalty on the client. The application will exist in both the SCCM and App-V client cache and cannot be purged from the SCCM client cache. This means in this delivery model that there is at least a doubling of the storage required on the client for this delivery model.

**Some considerations about the integration**

SCCM does not provide the exact functionalities we can find in the native App-V components implemented. The idea of SCCM integration is to complement the App-V platform. But, because of the Configuration Manager architecture, there are some considerations we should review.

Most of the integration features remain undocumented by Microsoft at the moment, but analyzing the infrastructure using the Beta and Release Candidate version of SCCM 2012 appears to maintain basic architecture definitions better than the previous version—SCCM 2007 R2/R3. Here are some of the considerations we can find so far:

• **We must re-advertise an application when there's an Active upgrade:** As we've mentioned before, Active upgrade is the process that we run in an App-V package to update the application using a service pack or any other type of modification. The App-V Full Infrastructure model automatically delivers the new version to clients. SCCM does not handle updates in the same package as a delta that must be delivered to clients, so we will need to make a new advertisement every time there's a virtual application update.

• **Reduced reporting:** App-V Full Infrastructure provides a very important set of reports about our virtual application that we can execute and retrieve; Configuration Manager does not provide the same level of reporting. Using Local Delivery (as used earlier) as the preferred method for delivering applications, it is not possible to report on how many times an application has been used.

For More Information:

Targeting applications for Remote Desktop Services requires users to log off and log on in their sessions: This is not only a limitation for virtual applications; it applies to all Configuration Manager Clients' user targeted and/or user interaction with the SCCM client. The SCCM client only allows software distribution to the console session of a terminal server system (mstsc.exe /console). Therefore, if application delivery targets users that are using a remote session on the terminal services system, they will not be able to execute the advertisement.

Asset Intelligence (for reporting and inventory features): This feature in SCCM requires Feature Block 1 present in the virtual application to be streamed to clients. Asset Intelligence cannot inventory virtual applications that co-exist with the same version of an application installed locally. As we mentioned, virtual applications live within their own environment, making it possible for the same application to work as locally installed and virtually deployed. If that's the scenario, Asset Intelligence won't inventory this App-V application.

In order to use Dynamic Suite Composition (DSC) in virtual applications, both interconnected packages must be advertised and registered with the App-V Client. That's why using the delivery method Local Delivery (downloading and execution), to be discussed later, is the recommended option when we are using DSC.

We can also use dependencies in SCCM to guarantee that both packages can be delivered normally.

Components involved
In this integration, we must understand which components are interacting in these two platforms:

- **App-V Sequencer**: The process of capturing it is the same, and we don't need to introduce any changes in that phase
- **SCCM Site Server**: In charge of managing and handling the actions performed by the SCCM Distribution Points
- **SCCM Distribution Point**: Storing and distributing the App-V applications
- **SCCM client**: This client agent communicates with System Center Configuration Manager and receives virtual applications
- **App-V Client**: SCCM Client and App-V Client work together—the SCCM Client delivers the virtual application to the App-V Client, which has the responsibility of executing it

For More Information:
Understanding delivery methods

Using SCCM to deploy virtual applications includes two types of delivery methods that we can use to fit any specific scenario. These two delivery methods are the same that appeared in the App-V + SCCM 2007 integration (covered in my previous book Getting Started with Microsoft Application Virtualization 4.6).

The delivery methods remain the same, but SCCM 2012 adds some twists to the configurations that we can achieve, using some options that we are going to review later, such as Persist content in the client cache and Enable peer-to-peer content distribution.

Applying the different delivery methods will basically depend on the type of connectivity the client machine has with the SCCM distribution point, converting this integration into a highly scalable one since we can discriminate the type of user with a specific type of streaming delivery.

Let's go through each type; they are Streaming Delivery and Local Delivery (downloading and execution).

Streaming Delivery

This delivery method represents the one similar to that which we discussed in Chapter 1, Taking a Deep Dive into App-V about Electronic Software Distribution (ESD). As we discussed, this is a role we can install on a server that will execute nothing more than delivering applications; this server can be a System Center Configuration Manager Server configured for Streaming Delivery.

When using this delivery method, the App-V Client will be configured to receive applications using HTTP/HTTPS (Standard Distribution Point) or SMB (Branch Distribution Point) streaming.
Delivery in the streaming mode occurs as shown in the following screenshot:

In this process, we can evaluate the entire workflow in the Streaming Delivery, from the moment the application is sequenced. These set of steps should be familiar to us if we already know and understand how a Streaming Server works with virtual applications, but note the following:

- The App-V Client does not stream an application until one of the shortcuts of this application is double-clicked
- Once the streaming process has started, the same behavior occurs at first—Feature Block 1 is delivered from the SCCM Distribution Point to the App-V client cache
- Once the application is running, the rest of the package is streamed down by the App-V Client

The Streaming Delivery method must be considered when the clients and servers live on the same LAN; remember that the streaming process requires high-bandwidth connections. Another good example of using this method is when we have applications that are constantly updated, and the updates occur in the Distribution Point that delivers the new package to clients. Avoid using this method when we have several offline users.

Local Delivery (downloading and execution)

The Local Delivery (downloading and execution) method is self-explanatory. The initial task is executed by the client who downloads the application and complete package, and then executes it.

In this downloading process, the application is delivered to the SCCM client cache, and then the SFT file is streamed from the SCCM clients' cache into the App-V clients' cache. Basically, the SCCM client works as a local Streaming Server for the App-V Client.

Delivery in this mode occurs as shown in the following screenshot:

In this process workflow, we can clearly see how the SCCM Client is in charge of downloading the entire content of the package as soon as it's advertised. The App-V Client streams the application to the cache from the SCCM Client cache, and then it completes the launch process only when the user clicks on any of the shortcuts from the application.

The application stays in the App-V cache, ready to be launched, as long as the advertisements in Configuration Manager are maintained.
Local Delivery is the best approach when we are using slow networks between servers and clients, and of course for offline users, who can work normally with the application, even when they are not connected to the network.

Local Delivery also needs a considerable amount of storage, three times the size of the application package: one for the SCCM Client, another for the App-V Client, and the third one is stored for calculating differentials when the application receives an update.

**App-V Client and the OverrideURL setting**

As we reviewed, in both delivery methods the SCCM client streams down the published applications to the App-V client cache; to accomplish this, the App-V Client includes a new registry value `OverrideURL`.

This value can be changed to use an alternate server in charge of delivering the virtual applications. All of this process is transparent to the user; the value is changed by the SCCM client and the streaming process is redirected to the Configuration Manager Distribution Point in charge of the delivery.

This is a simple diagram where we can see the interaction between the SCCM Client and the App-V Client:
Integrating App-V with System Center Configuration Manager 2012

Even though this concept appeared in the integration of App-V and SCCM 2007 R2, the basic functionality remains in this new version SCCM. By reviewing the deployment of App-V applications using SCCM, we can confirm that the OverrideURL behavior is maintained. There's an example of the Winamp virtual application (deployment reviewed later in this chapter) and the registry settings in the App-V Client.

The OriginalURL setting displays the configuration used in the OSD, and the OverrideURL uses the SFT location in the SCCM Client cache:

<table>
<thead>
<tr>
<th>Name</th>
<th>REG_SZ</th>
<th>Winamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>OriginalURL</td>
<td>REG_SZ</td>
<td>RTSP://appv-server:554/Winamp.sft</td>
</tr>
<tr>
<td>OverrideURL</td>
<td>REG_SZ</td>
<td>FILE://c:\windows\ccm\cache\content_5dc317e8-23f1-44cf-9d5-5c7a2bbf7c16f91.system</td>
</tr>
<tr>
<td>PkgURL</td>
<td>REG_SZ</td>
<td>FILE://c:\windows\ccm\cache\content_5dc317e8-23f1-44cf-9d5-5c7a2bbf7c16f91.system</td>
</tr>
</tbody>
</table>

Requirements for the SCCM + App-V integration

In order to accomplish this smooth integration without adding significant load to our operations, we must understand each of the requirements involved.

The requirements do not differ much from what we've seen earlier in SCCM 2007 R2, which is an important advantage if we already have this platform implemented and are considering migrating to SCCM 2012.

The two important requirements we are going to analyze are: platform requirements in SCCM 2012 and storage requirements for sizing the SCCM client cache.

Platform requirements in SCCM 2012

Even though it could sound pretty obvious at this point, always remember that having a healthy environment in SCCM is an important requirement before starting with changes in the environment.

Regarding System Center Configuration Manager 2012 components, the basic platform requirements are as follows:

- **Primary site**
- **Site Server**: A Site Server with the following roles installed:
  - Site System
  - Site Server
  - Component Server

For More Information:
• **Distribution Points**: At least one available and working Distribution Point, with IIS and streaming enabled for BITS application delivery, is required. This server will be in charge, of course, of the package distribution.

• **Clients**: SCCM clients must be installed and work properly. We'll review how to push clients' agents in SCCM 2012, later in this chapter.


**Storage requirements**

The storage requirements for SCCM 2012 + App-V remain the same as the SCCM 2007 integration.

These considerations depend primarily on the type of the delivery method chosen in the environment. As for the App-V environment, we must size the storage considering clients, App-V cache, and server distribution points in SCCM. A general guideline for the storage requirements is as follows:

• **SCCM Client cache** must be configured considering the full size of the App-V packages to be distributed.

• It is recommended to size the App-V client cache considering the SCCM client cache defined. The App-V client cache should be configured with a free disk space threshold option, adding 1 GB more to the SCCM client cache value.

  When using an SCCM Client cache with 4 GB, the App-V Client cache should be configured with a free disk space threshold of 9 GB.

• **SCCM Distribution Points** should allocate space considering the size of the package multiplied by three. The triple-sizing consideration, as said, depends on the delivery method, current version of the package, upgrade version, App-V client cache version, or differential files while constructing an upgraded version of the package.

For More Information:

Integrating App-V with System Center Configuration Manager 2012

In the official guide provided by Microsoft, *Virtual Application Management with Microsoft Application Virtualization 4.5/4.6 and System Center Configuration Manager 2007 R2*, we can find a complete reference about all the necessary requirements, including the storage considerations. We can download this guide from the Microsoft Download Center, at http://www.microsoft.com/downloads/en/default.aspx.

To configure the SCCM Client cache, we can check the following official article by Microsoft: *How to Manage Clients in Configuration Manager* (http://technet.microsoft.com/en-us/library/gg712288.aspx). See the section titled *Configure the Client Cache for Configuration Manager Clients*.

Implementing SCCM and App-V integration

Finally, after reviewing all the benefits, requirements, and considerations about the integration, it is time to get our hands into the SCCM 2012 and App-V integration.

As mentioned earlier, the scope of this chapter will not focus on understanding how to use the System Center Configuration Manager platform, but we’ll take a closer look at the basic operations tasks, such as handling SCCM client installations.

In this section, we are going to review the following topics and step-by-step procedures:

- Installing System Center Configuration Manager 2012
- Distributing the App-V Client with SCCM 2012
- Using virtual applications in SCCM
- Creating virtual applications
- Deploying virtual applications
Installing SCCM 2012

This section will consist of an overview of SCCM 2012 installation using the Release Candidate bits of SCCM.

SCCM 2012 requirements

Configuration Manager is a robust platform and must be analyzed and sized carefully in every environment. In this example, we will be using a minimalist scenario.

For the primary site, the following configuration is required:

- Active Directory and DNS platform in place and working properly
- Installing SCCM requires an Active Directory schema extension (the same extension used in SCCM 2007)
- If the current AD platform was already extended for SCCM 2007, there's no need to extend it again

![For more information about this matter, please see the article Determine Whether to Extend the Active Directory Schema for Configuration Manager, at http://technet.microsoft.com/en-us/library/gg712272.aspx.](image)

- Windows 7 client machines
- For the SCCM 2012 Primary Site, general requirements for installation are the following:
  - Memory: 2 GB minimum; 4 GB or more recommended
  - Windows Server 2008 R2
  - IIS role installed with the following features:
    - Common HTTP features: Static Content, default document, directory browsing, HTTP errors, and HTTP redirection
    - Application Development: ASP.NET, .NET Extensibility, ISAPI Extensions, and ISAPI Filters
    - Health and Diagnosis: HTTP logging, logging tools, and Request Monitor Tracing
    - Security: Windows Authentication and Request Filtering
    - Performance: Static Content Compression
    - Management Tools: IIS Management Console, IIS Management Scripts and Tools, IIS 6 Metabase Compatibility, and IIS 6 WMI Compatibility

For More Information:

The following features also have to be added:

- .NET Framework 3.5.1
- Background Intelligent Transfer Service (BITS)
- Remote Differential Compression
- SQL instance available with one of the following characteristics:
  - SQL Server 2008 SP2 with Cumulative Update 6
  - SQL Server 2008 R2 SP1 and Cumulative Update 3

IMPORTANT:

These requirements are based on a simple and minimalistic SCCM 2012 implementation; depending on the installation type and SCCM roles involved, the requirements at this stage will vary.

There are other optional components available, which are not present because of the scope of this book. For example, Windows Server Updates Services (WSUS).

For more information about requirements in SCCM 2012, please visit the following Microsoft TechNet link: Supported Configurations for Configuration Manager (http://technet.microsoft.com/en-us/library/gg682077.aspx).

Installing SCCM 2012

Let's review the step-by-step process for installing System Center Configuration Manager 2012 Beta 2:

1. Launch the Install option in the SCCM 2012 setup window.
2. The option we are going to use is Install Configuration Manager Primary Site server.
3. In the setup process, SCCM 2012 (as well as its predecessor) requires a folder where the platform will download the latest updates for Configuration Manager.
4. Configure the site information necessary for the Primary Site configurations—Site Code and Site Name. Also, leave the option for Install Admin Console selected.

5. Since this is the first SCCM Server and there’s no existing Central Site, select the option, Primary site will be installed as a standalone site.
Central Sites do not process any client data, do not accept client assignments, and only have some certain functionality related to coordinate sites replications.


6. Configure the database server. Remember that the only SQL instances supported in this version are SQL Server 2008 SP2 with Cumulative Update 6 and SQL Server 2008 R2 with SP1 and Cumulative Update 3.

There have been some changes between the SQL supportability in the SCCM 2012 Beta versions and the Release Candidate. There could be some changes for the RTM version also, but there is no official word from Microsoft about it.

7. Enter the NetBIOS name for the server where the SMS provider will be installed; in this case, it is the same server.

8. In **Client Computer Communication Settings**, choose for client machines to be able to use HTTP and HTTPS for communication. Remember that HTTPS communications require a healthy PKI infrastructure in all servers and clients. However, that topic goes beyond the scope of this book.

---

For More Information:

9. Make sure also that **Management Point** and **Distribution Point** are using **HTTP** for communication.

10. Complete the wizard to start installation. The process will take a while, so be patient.

Once the installation is complete, we will be able to appreciate a refreshed SCCM console with some great UI improvements and also standardize the entire System Center suite to fit with the same configuration.

SCCM 2012 includes four panes from which we can handle the entire suite: **Administration**, **Software Library**, **Monitoring**, and **Assets and Compliance**.

Within **Administration**, we can find all the options for managing the SCCM infrastructure (sites, client options, security, distribution points, and so on). Some of the initial tasks we can perform are about the discovery options from which clients will be presented.

For example, using **Active Directory System Discovery**, SCCM will find client machines by searching for computer objects in AD Domain Services.

---

For More Information:
Under **Site Operations**, we can review all the sites we have configured in our platform, accessing each site’s properties we can configure general parameters involved in that scope, including for example, the ports used by Configuration Manager to communicate with clients.

![Configuration Manager 2012 Beta 2 Test Drive Site Properties](image)

For More Information:
Within **Site Operations**, there are several client configurations available; one important one is **Client Installation Settings**, from which we can define how the **Configuration Manager Client** agent (necessary in SCCM) will be installed.

If you wish to install the client agent as soon as the computers are discovered, check **Enable automatic site-wide client push installation**. Take note that, if we have several machines deployed in our **Active Directory** domain, this action can be network-intensive.

For manual deployments of the Configuration Manager, we can use the **Assets and Compliance** pane. In **Devices**, we have all the default collections created, and in **All Systems**, we can find the discovered computers.

In this example, we will be using a created collection, **Windows 7**, which contains one computer: **W7-CLIENT**.

For More Information:  
By selecting **Install Client**, we will receive a new wizard to confirm the SCCM Client installation. In this example, we will be using the specific site of **Configuration Manager 2012 Beta 2** (pre-configured VHD available from Microsoft).

![Install Client Wizard](image)

This task can usually take several minutes to complete, depending on several factors. If you've had the chance to work with SCCM previously, you'll know that this kind of task is never completed immediately.

Still, we can review the latest status of each client in the same console. In the **Push Information** section, we will retrieve the latest status of the client installation.
Integrating App-V with System Center Configuration Manager 2012

Distributing the App-V Client with SCCM 2012

Since the SCCM 2012 platform is installed and the client agents are deployed successfully, we are going to learn how to install the App-V Client in a given SCCM Collection, in this section.

SCCM 2012 provides a new experience in UI for handling applications and packages, but basically, the steps for deploying the App-V Client will not differ much if we have already completed them in SCCM 2007.

In this example, we will use the setup.exe file, which installs the prerequisites in the same process. The file setup.msi can be used separately, but the prerequisites must also be installed separately.

Distributing the App-V Client using SCCM is not necessary for the integration to take place; this is an optional step and should not be considered if the App-V Client is already installed in all clients.

Once the SCCM client is installed for all the virtual applications distributed with SCCM, the necessary configurations will be completed automatically.

Let's look at the process for adding and installing the App-V Client package:

1. Within the Software Library pane, we have several options available, since this "application" contains several files necessary to completing a successful installation. Under Application Management, right-click Packages and select Create Package, as shown in the following screenshot:

2. Specify the package name, and enable the This package contains source files option.

For More Information:
3. In Set Source Directory, select the UNC path for the installation files.

For More Information:
One great improvement in SCCM usability is how we use the Programs assets, which are basically the command lines to be used for deploying packages or scripts.

Using SCCM 2012, as soon as we try to add a package, we will be asked to insert a new program (previously, this program had to be created separately).

4. In Program Type, select Standard Program.

5. Select the program command line and other parameters. Since I’m using the .exe file, the command line used is `setup.exe /s /v"/qb-1 SWICACHESIZE="6144".`

6. In Requirements, we can configure several parameters prior to running this program, for example, the client platforms to be installed or the estimated disk space.

7. Complete the wizard.

For More Information:
8. As soon as the package is added, we can select the option **Deploy**, in the SCCM 2012 console for that package.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍎</td>
<td>App-V Client</td>
</tr>
<tr>
<td>🕵️‍♂️</td>
<td>Manage Access Accounts</td>
</tr>
<tr>
<td>📚</td>
<td>Create Program</td>
</tr>
<tr>
<td>🔄</td>
<td>Refresh F5</td>
</tr>
<tr>
<td>🗑️</td>
<td>Delete Delete</td>
</tr>
<tr>
<td>🌐</td>
<td>Create Prestige Content File</td>
</tr>
<tr>
<td>☐</td>
<td>Deploy</td>
</tr>
<tr>
<td>📦</td>
<td>Distribute Content</td>
</tr>
<tr>
<td>🔄</td>
<td>Update Distribution Points</td>
</tr>
<tr>
<td>🔄</td>
<td>Move</td>
</tr>
<tr>
<td>🕵️‍♀️</td>
<td>Set Security Scopes</td>
</tr>
<tr>
<td>📋</td>
<td>Properties</td>
</tr>
</tbody>
</table>

9. In the **Deploy** option, we will receive a new wizard. The first option will let us select the **Collection** to be used for this deployment.

For More Information:
10. In **Deployment Settings**, we can configure how to deploy this package. **Available** is used to let the package be available to the client and let the user decide when to install it, whereas **Required** will automatically install the client.

11. In **Scheduling**, we can configure the schedule options for this package to be installed. In my example, I'm using **As soon as possible**.

12. In **User Experience**, we can configure options about the possibilities each user receives when the installation or assignment is taking place.
13. In the next window, we will receive the options regarding the type of delivery methods, whether Streaming Delivery or Local Delivery (downloading and execution):

- **Download content from distribution point and run locally** represents the Local Delivery method.
- **Stream content from distribution point** represents Streaming Delivery.
- Note also that we can configure a different type of delivery, depending on whether the client is connecting from the same LAN or using a slow connection.

14. In **Summary**, review the deployment settings for the package and finalize the deployment configuration.

For More Information:
Integrating App-V with System Center Configuration Manager 2012

Using the Monitoring pane, we can review the current status of the deployment, including Completion Statistics, which retrieves a simple and easy report overview of the deployment without requiring running any specific report.

Also, in this section, we can modify the deployment properties configured recently for the same package.

Using virtual applications in SCCM 2012

Creating, handling, and deploying virtual applications within SCCM 2012, has been simplified compared to what we needed to configure when we were using SCCM 2007 R2. Virtual applications are supported by default without requiring any execution change in the platform.

In this section, we will learn about the following:

- Creating virtual applications in SCCM 2012
- Deploying virtual applications in SCCM 2012
- Reviewing the deployment of virtual applications in clients

For More Information:
Creating virtual applications

The concept of "importing" virtual applications does not apply any more in SCCM 2012; "creating" is the right word in this case. However, the basic steps remain the same. To create a virtual application, we need the basic component: the manifest (XML file).

Let's look at the process step-by-step, as follows:

1. In Software Library | Application Management, right-click on Applications and select Create Application.

2. Select the path for the application manifest using the UNC. In this example, we will be using the **Winamp** virtual application.
3. Complete the information about the application: name, version, and so on.
4. Once the wizard is completed, the application will be added into SCCM.

As soon as the application is added, we can configure its deployment using the same section in the console.

**Deploying virtual applications**

The deployment process of App-V packages is pretty much the same used in SCCM 2007, but some interesting options are added in the process.

Some of these new capabilities are not actually new, but the usability of these features has been simplified in order to optimize our deployments. Some of the interesting capabilities are: handling deployment types or generating alerts depending on success and/or fail rate.

Let's take a look at the deployment process in SCCM 2012 for virtual applications:

1. In the *Software Library* section, using the applications list, we can right-click the application we would like to deploy and select the relevant options.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>🛠</td>
<td>Foxit</td>
</tr>
<tr>
<td>🛠</td>
<td>ImgBurn</td>
</tr>
<tr>
<td>🛠</td>
<td>Skype</td>
</tr>
<tr>
<td>🛠</td>
<td>Winamp</td>
</tr>
<tr>
<td>⬤</td>
<td>Revision History</td>
</tr>
<tr>
<td>⬤</td>
<td>Update Statistics</td>
</tr>
<tr>
<td>⬤</td>
<td>Create Prestage Content File</td>
</tr>
<tr>
<td>⬤</td>
<td>Create Deployment Type</td>
</tr>
<tr>
<td>⬤</td>
<td>Reinstate</td>
</tr>
<tr>
<td>✗</td>
<td>Retire</td>
</tr>
<tr>
<td>⬤</td>
<td>Export Application</td>
</tr>
<tr>
<td>⬤</td>
<td>Copy</td>
</tr>
<tr>
<td>⬤</td>
<td>Refresh</td>
</tr>
<tr>
<td>⬤</td>
<td>Deploy</td>
</tr>
</tbody>
</table>
2. Select the **Collection** of devices where we would like to deploy this app, and also use the associated distribution point from which the clients will retrieve this application. In this example, the Distribution Point **TESTDRIVEB2** is used with the **SCCM 2012 Beta 2** installed:

![Deploy Software Wizard](image)

3. In **Deployment Settings**, we have the same options seen earlier—**Available** and **Required**.

4. In **Scheduling**, as seen before, we can define when this application will be available for deployment on client devices.

---

For More Information:  
5. In User Experience, we have similar options to the ones we've seen in the App-V Client package. In this section, we can configure it to **Hide all notifications** for the users. This option is not selected as we will be reviewing the installation process in the client later.

6. Deploying software also includes the possibility of managing the alerts regarding this process. We can configure this deployment to elevate alerts for SCCM and/or System Center Operations Manager (SCOM).

For More Information:
In the SCCM section, we can configure the following options:

- Warning if the deployment success rate is below the selected percentage.
- Warning if the deployment failure rate is below the selected percentage.
- And for SCOM, we must enable the option to generate alerts in Operations Manager. By enabling this option, the SCCM client will communicate with the SCOM agent in the same machine to elevate this warning.

7. Review the **Summary** and complete the wizard.

Once the steps are completed, in the **Software Library** section, we can also review the **Deployment Types** records existing for the applications.

<table>
<thead>
<tr>
<th>Applications 5 items</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Icon</td>
<td>Name</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Front</td>
</tr>
<tr>
<td>![Icon]</td>
<td>ImgBurn</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Skype</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Winamp</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Winamp-copy</td>
</tr>
</tbody>
</table>

**Winamp**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Priority</th>
<th>Name</th>
<th>Dependencies</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>1</td>
<td>Winamp - Microsoft Application Virtualization</td>
<td>No</td>
<td>App-V</td>
</tr>
</tbody>
</table>

Double-clicking the deployment type, we will get the parameters configured earlier, plus a few more we should consider. Let's take a look:

- In the **Content** tab, some of the options available are as follows:
  - **Persist content in the client cache**: This option is used when we want to store this application in the cache and prevent an automatic deletion (which must be configured manually) for the files used.
° **Enable peer-to-peer content distribution**: This option is used for client machines to distribute the content with other client machines that are near. This parameter is not yet documented by Microsoft, so we cannot confirm how it works exactly.

° **Load content into App-V cache before launch**: Again, this option is not documented by Microsoft yet, but what we assume is that the package is loaded completely into the App-V cache prior to launch, instead of just Feature Block 1.

In the following options, as we saw earlier, we can find the behavior of the delivery types—Streaming Delivery or Local Delivery (downloading and executing).

- In the **Requirements** tab, we can configure the dependencies for the deployment. In this case, there's only **Operating System**, but there are several other options, such as client's memory, disk space, processor, existing registry path, and so on.

---

For More Information:  
Here’s an example of creating a *Global Condition* (selecting a *Custom requirement*) to be included as a requirement in application deployment—we will be using it to be assessed by the client prior to deployment; if the file exists, the deployment continues:
As a reference, we will be using the Program Files App-V Client default installation folder and selecting the App-V Client Management Console (SftCMC.msc).

- In **Dependencies**, we can configure an existing application as a dependency to deploy the application.
- Each dependency can be configured with a selected **Priority**. This option is used with an **Auto Install** parameter. The application with higher priority will be installed first.

We cannot configure an existing SCCM package as a dependency. The only possibility is a previously added application.

### Deployment Types

The **Deployment Types** are basically a deployment profile configured for an application. Using this option in SCCM, we can configure different parameters for an application deployment to take place.

Take note that these Deployment Types are used to set parameters in the process of deployment behavior, but not parameters in the application settings.

For every application, we can have different deployment types, which can be configured by just right-clicking the application and selecting **Create Deployment Type**.

Within the wizard, we will get the chance to configure all the necessary parameters for deployment. After completing this option, we will have different Deployment Types. Thus, we can be certain to fit each deployment process in every scenario.

Here are some examples:

- We might need to choose different **Requirements** for 32-bit and 64-bit clients. For example, the App-V Client installs on a different Program Files folder, and adds different drivers, in each case.
• For roaming users, we would like to use Local Delivery (downloading and execution) for client machines, guaranteeing that users can run the application by loading locally and not depend on a Streaming Server.

Deploying applications in clients

Once we have configured the application and the deployment process, we just need to wait till that package is deployed to the SCCM client.

SCCM 2012 includes Software Center for every client machine deployed. Using Software Center, we can retrieve the latest status of applications available and installed for the client machine, as well as information about each package.

In this example (shown in the following screenshot), the client machine shows the virtual applications that are already installed as well as virtual applications available for installation. For those applications marked as Available, we must manually select Install, to complete installation.

For More Information:
Integrating App-V with System Center Configuration Manager 2012

The installation of the package depends on the package size and the deployment parameters we've configured.

<table>
<thead>
<tr>
<th>NAME</th>
<th>PUBLISHER</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winamp</td>
<td></td>
<td>Installed</td>
</tr>
<tr>
<td>Skype</td>
<td></td>
<td>Waiting to install</td>
</tr>
<tr>
<td>Foxit</td>
<td></td>
<td>Installed</td>
</tr>
</tbody>
</table>

Since we had configured the system to not suppress notifications earlier, we should receive all the messages about the deployment processes.

Also, we can review the current applications deployed in the App-V Client console. We can verify that the package URL for each application deployed using SCCM must be directed to the local SCCM cache (the default location is C:\Windows\ccmcache).

For More Information:
With those verifications, we can confirm that our **SCCM + App-V integration** is working properly for deploying applications.

If we need to troubleshoot virtual application deployment, the SCCM client also includes a log file dedicated to virtual apps. This log file can be found, along with the rest of SCCM logs, in `C:\Windows\CCM\Logs\VirtualApp.log`.

We've completed the SCCM 2012 and App-V integration successfully. There are other possibilities within this integration, such as understanding **Asset Intelligence** in the SCCM console, that are beyond the scope of this chapter.

**Summary**

In this chapter, we've completed a comprehensive analysis of the App-V integration with System Center Configuration Manager 2012, not only analyzing the main benefits, requirements, and considerations, but also putting our hands into the deployment of SCCM 2012.

Integrating App-V application with System Center Configuration Manager 2012 is much simpler than what we've seen with SCCM 2007. There are a few suppressed steps (such as enabling virtual application support in SCCM or adding programs manually to the console, to deploy a package), and the UI improvements facilitate the work of all of IT administrators.

For More Information:
In this chapter, we've looked at the basic concepts in SCCM and App-V integration—components involved and the delivery methods existing for distributing virtual applications. This last topic is an important matter when we are designing our infrastructure, since an incorrect delivery method could result in several infrastructure problems as well as provide bad user perception.

With the considerations completed, we had the chance to complete the necessary steps to implement SCCM 2012 in a minimalistic scenario. And with that, we also reviewed the process involved for deploying the App-V Client component.

As a final step in this chapter, we had a closer look at the management of the virtual applications in SCCM 2012. We had a look at creating and deploying virtual applications to our clients and also understood how the Deployment Types appear with an important role in this integration.

In the next chapter, we are going to review another type of App-V integration—private clouds and System Center Virtual Machine Manager 2012. This time, we are not going to consider desktop virtual applications, but server virtual applications.
Where to buy this book


Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.