

# Best Practices for Building Virtual Appliances



## Best Practices for Building Virtual Appliances

rPath's rBuilder® produces virtual appliances for all leading virtual machine formats, including:

- VMware® Virtual Machine
- Citrix® XenServer XVA
- Microsoft® Hyper-V
- Amazon® Machine Image
- Virtual Iron® Virtual Machine
- Parallels® Virtual Appliance

### Introduction

The growing acceptance of virtualization technology is revolutionizing the way IT consumers deploy applications. A major part of this shift is being driven by software vendors who provide virtual appliances — an application integrated and packaged with just enough operating system (JeOS) that can be easily installed in a virtual machine “container.”

One of the most significant advantages of a virtual appliance is that IT consumers are not required to integrate and configure application and operating system components into a working solution themselves. However, the process of constructing and maintaining virtual appliances is poorly defined today, making it difficult for software vendors to deliver them.

rPath's rBuilder® solves this problem by providing a tool for software developers to create virtual appliances that deliver the best possible end user experience and the least possible ongoing maintenance cost. rBuilder incorporates several virtual appliance best practices including:

- No command line interaction should be required
- Appliance management should be web-based
- There must be a rollback mechanism for update failures
- The appliance must have the minimum possible install size
- The appliance must be secure by default
- There must be an automated method for updating both the appliance image and the field unit

This paper will explore these best practices in detail, along with the rationale for adopting each one.

### Why Build A Virtual Appliance?

Virtual appliances represent an opportunity to streamline and simplify software distribution for virtual environments. With a virtual appliance, software vendors can ship thoroughly tested, standard configurations that require minimal installation effort. End-users are spared the complications of installing multiple supporting systems, since everything is contained in the virtual appliance package — the operating system, services and applications. This dramatically decreases potential support issues related to incorrectly configured software or hardware. Users can simply download your virtual appliance and run it with virtualization software such as VMware ESX, Citrix XenServer or Microsoft Hyper-V.

### How to Build a Virtual Appliance

There are several ways to build virtual appliances ranging from manual to fully automated. Many developers and organizations already create working virtual appliances.

As the leading provider of tools that allow the efficient creation of virtual appliances, rPath has identified several characteristics which are shared by successful virtual appliances. We built our tools with the following best practice guidelines in mind.



## Best Practices for Building Virtual Appliances

### Simple Installation

Although this would seem to fall into the category of obvious, it bears stating. A virtual appliance is no good if it can't be started quickly and easily in a virtual machine. To that end, the installation should happen automatically or require minimal end user interaction in the form of choosing configuration options through a simple web interface, preferably a wizard.

When building an appliance with rBuilder, a developer can easily include pre-built configurations for all application and operating system components. This allows installations to occur without manual intervention. Once configurations are made, they can be moved forward to new versions of application and operating system components with little or no manual effort. Ongoing maintenance of the virtual appliance image becomes vastly simpler than in other packaging models.

In addition, rPath provides the rPath Lifecycle Management Platform™, a web based system configuration tool which includes a configuration wizard that runs on initial installation. The rPath Lifecycle Management Platform interface can provide configuration management for any appliance.

### No Command Line Required

As a corollary to the above, all installation and configuration settings required for proper operation should be handled automatically by the appliance, or be configurable by the user with a web based interface. At no time should the user need to “drop to the command line” to give instructions to the appliance.

An intuitive web-based interface is the best way to optimize the users' experience with a virtual appliance. Given that there is plenty of room for error when someone is inputting commands directly, it is better from a support standpoint to circumvent command line access with an interface. An appliance building tool should offer the ability to include a web-based command console in each appliance.

Similarly, the appliance should start automatically when the system is booted and not require input from the user to begin operation. Not having a command line is an important feature of making the virtual appliance easy to deploy, maintain and use. It also allows the virtual appliance to be moved quickly from one virtual machine to another for rapid resource allocation or even disaster recovery.

As described above, rPath enables these capabilities through the rBuilder virtual appliance build process and the rPath Lifecycle Management Platform, not only for the initial installation process, but for ongoing system reconfiguration and use.



## Best Practices for Building Virtual Appliances

### Smallest Possible Footprint

An appliance encapsulates the application with the complete set of components it needs to run as a “plug and play” entity. Unfortunately, the way that most virtual appliances are built means that they contain a large number of packages and services that are not necessary for the execution of the core application. This is because they are built from the bottom up by installing a full size operating system and attempting to remove as many components as possible. The large number of package interdependencies defined in most standard operating systems limits the degree to which such an approach can shrink the installation footprint. Such bloated appliances not only take up precious system resources, but also complicate maintenance and updates with extended dependencies among components.

rBuilder uniquely allows developers to define the applications that are needed to run in the virtual appliance, and then automatically determine only the components that are necessary to run those applications. This top down approach, combined with fine-grained package dependencies allows virtual appliances constructed with rBuilder to install the minimal possible set of components. Working with the smallest possible footprint at the outset yields gains throughout the appliance lifecycle, in areas ranging from disk space to execution time to smaller update requirements.

### Secure By Default

Because a virtual appliance is designed to perform only a single function, it should be possible to limit its security vulnerabilities considerably as compared to a general purpose server. However, the mechanisms used to construct such appliances typically do little to avail themselves of this opportunity.

The mechanisms rBuilder uses to insure a small image footprint have the additional benefit of reducing the surface area that is open to attack. The result is not only having less code in the appliance, but also running only the services necessary as part of application functionality.

The process for configuring all application components to simplify the install and management process can also be applied to security configuration, allowing predefined firewall, networking, and specific security enhancements to be made at build time and preserved in a sustainable fashion over the life of the appliance.

### Automated Software Updates

Once an appliance is deployed, there should be a method of seamlessly updating it with new application features and maintenance streams. It makes most sense to do this automatically and simultaneously across the entire stack of components contained in the appliance. Thus, a new virtual appliance that provides a complete upgrade in a single step can be released.



## Best Practices for Building Virtual Appliances

While many end users will have adequately separated their local data from the virtual appliances and rely on full image updates and snapshots for ongoing maintenance, some will expect to run the same virtual appliance and incrementally update it over time. So, each deployed virtual appliance should be capable of receiving the same updates that are made to the virtual appliance image. These updates should be provided through a secure online mechanism that is controlled through a web administration tool.

In the event that there is an error or incompatibility in the update, the virtual appliance should have the ability to immediately roll back to a known good state.

rBuilder allows for the simple updating of virtual images to include new operating system and application components. Once these updates are made, they are available not only for the creation of new images for end users to deploy but also as incremental updates to existing deployed units.

The rPath Lifecycle Management Platform provides end users with a simple web interface for performing online updates and gives them a mechanism for rolling back individual changes if there is a failure during an update.

### Choosing The Best technology to Build a Virtual Appliance

Some organizations choose to build virtual appliances internally, often in an ad hoc fashion. The virtual appliance build is not standardized, the contents in the appliance package are not recorded and the results are not reproducible.

Others are choosing pre-certified software stacks that may limit their flexibility to choose the optimal operating system and services for the application. These pre-certified stacks may also lead to bloated appliances that include many more components than they need to run efficiently.

rBuilder is a comprehensive tool for assembling and packaging lean and standardized virtual appliances. rBuilder's automated analysis of the application ensures that only the minimal set of services and applications are included in the appliance. All dependencies are completely resolved without the need for user intervention. The build elements are preserved in rBuilder's unique repository system, allowing for a consistently repeatable process.

The rPath Lifecycle Management Platform allows the user to start and configure the appliance, as well as schedule updates. There is never a need for the user to access the command line to perform any function related to the appliance's operation.

Finally, rBuilder allows simultaneous online updates of the appliances in the field. The user is completely relieved of managing the maintenance stream, while gaining the benefits of the latest features and fixes.



## Best Practices for Building Virtual Appliances

### About rPath

rBuilder and the rPath Lifecycle Management Platform™ automate the creation, configuration, management and maintenance of application images for virtualized and cloud computing environments. By producing application images that are optimized for any hypervisor, rPath frees the application from the underlying hardware, and enables a more efficient, lower cost model for development, deployment and support. rPath's end-to-end technology simplifies the entire range of lifecycle management activities for application images, while promoting scalability in response to dynamic demand. rPath is headquartered in Raleigh, NC.

Corporate Headquarters:  
701 Corporate Center Drive, Suite 450  
Raleigh, NC 27607

+1 919.851.3984 Main    info@rpath.com  
+1 866.508.6200 Sales    www.rpath.com  
+1 919.851.3985 Fax

