

rPath® Versus Other Software Appliance Approaches

white paper



Executive Summary

Building a virtual appliance is not as easy as it may sound in your engineering department. Although plenty of smart engineers probably can build a one-off appliance, building it the same way again is a different story. If you think you might gain on short-term results by building a virtual appliance in-house, consider the longer-term price in terms of deploying, managing and maintaining the appliance. So, before you authorize your engineering department to build a virtual appliance, ask some important questions:

- How will you ensure the repeatability and consistency of the appliance build over time?
- Are you ready to invest in additional quality assurance and testing to support multiple appliance formats?
- Do you have a way to track changes to appliance components and provide a maintenance stream to field units?
- Will you be developing a graphical interface to perform tasks such as configuration and backup?
- How will you provision and manage licenses for the appliance?

Because it provides a ready solution to all these questions, and more, rPath's rBuilder® is the best way to build virtual and software appliances. In this paper, we intend to prove that point by highlighting the specific ways that rBuilder and the rPath® Appliance Platform are different from home-grown approaches to building virtual appliances. We'll also show that rPath's solutions are significantly more robust than any other appliance building tools on the market today. We won't name names - our "competitors" know who they are - but you'll see that rPath takes a unique and innovative approach to building, deploying and maintaining software appliances.

For purposes of comparison, we refer to what you would encounter when using a general-purpose OS solution, a snapshot tool solution and rPath's solutions. A general-purpose OS solution refers to a top-down process of stripping down an operating system into a slimmed down version of its former self that is more suitable for inclusion in an appliance. A snapshot tool refers to a platform that provides rudimentary automation for appliance generation and minimal, script-based assistance with assembling the various software components.

rBuilder - A Complete Appliance Development Tool

Smaller Appliance Footprints with Just Enough Operating System (JeOS)

Smaller appliances are more secure, more stable and higher performing. The fewer components they have, the less will go wrong or be vulnerable to attack and the faster they can go. Why load an appliance with more than it needs? The key to keeping an appliance small is the concept of just enough operating system (JeOS). This means that the appliance contains only those pieces of operating system that are required to run the application.

General purpose OS: Starting with a general purpose OS each time and stripping it down for the appliance is a recipe for disaster. A knowledgeable engineer must remove the components that are not essential for the appliance, while preserving the ones that are - along with the various bits and pieces those components are dependent upon to work properly. Even a minimal install of a general purpose operating system is typically larger than an entire appliance built with rBuilder.

Simple snapshot tool: These vendors provide you with a minimal install of a general purpose operating system pre-loaded into a virtual machine. Some snapshot tool vendors imply that this is JeOS, but it's really the same as the general purpose OS scenario - you wind up with a non-reproducible build with more components than you actually need to run your application. Real JeOS - reproducible and supportable - requires rBuilder.

rBuilder: rBuilder automatically determines the dependencies among software components needed for the application. During appliance creation, it uses this information to ensure that only the components required are included in the appliance. This bottom-up approach to JeOS ensures that appliances built with rBuilder are smaller in size with fewer software components to maintain. In practice, this smaller footprint can reduce the maintenance burden for the appliance by as much as 90%.

Automated and Consistent Build Process

Most talented engineers can build an appliance - but can they build it the same way each time? A consistent and repeatable build process is crucial to maintaining the integrity and operability of the appliance over time.

General purpose OS: If the appliance is going to be updated in the future, its components must be stored and tracked. This is typically a time intensive and manual process and may involve identifying and locating components stored in code silos throughout the organization. How will your engineer know whether any of those pieces of code have been updated by others? Manually assembling components into an appliance requires several steps that leave it open to errors and inconsistent results.

Simple snapshot tool: A snapshot tool may provide some automation assistance with the assembly of the appliance or the stripping of the operating system. Unfortunately, it does not track software

components contained in the appliance, so like the general purpose OS, cannot reproduce an appliance consistently.

rBuilder: rBuilder is designed around a centralized repository model. rBuilder stores and versions all the components contained in the appliance in the repository. This gives you the ability to rebuild the appliance in a consistent and repeatable way using the correct version of each component. rBuilder also automates key portions of the appliance-building process, reducing errors and ensuring that appliances are consistent across the organization.

Validation, Auditing, and Reporting on Appliance Builds

Quality control, license auditing and cryptography filings are your responsibility to assure compliance with corporate policy and export laws. It is essential to know what is included in your software and be able to report on it as needed.

General purpose OS: If you choose to build an appliance yourself from a general purpose OS, your development and QA teams are responsible for quality control and must construct the appropriate test case scenarios. Any policies and rules in place in the enterprise must be manually enforced through testing. Further, your developers must manually keep track of the individual licenses of the appliance's components. You'll be leaving yourself open to liability if you can't produce the correct license information for your appliance.

Simple snapshot tool: As with a general purpose OS, these offerings do not provide any inherent quality control, license auditing or cryptography filing capabilities. You're basically on your own.

rBuilder: rBuilder comes with over 50 built-in policies that check for problems with the appliance components; you can also write your own and have rBuilder enforce them. These automatic validation steps ensure a higher level of quality for the appliances you generate. rBuilder also reports on the licenses contained in the appliance. This greatly facilitates legal review of your appliance and can help ensure compliance with the numerous open source licenses. Finally, rBuilder generates a report on the cryptography contained in the appliance to assist with the U.S. export filing process.

Automatically Generate Multiple Appliance Formats

As you expand your appliance offerings, you'll want to give your customers a choice of appliance formats. Perhaps they want to run an appliance on bare metal or in one or more virtual environments. As the number of choices increases, so does the need to engineer and maintain different versions of the appliance.

General purpose OS: If you are building your appliance yourself, each new image format that you select adds complexity to your appliance generation process. Your engineer must maintain multiple appliance definitions to support the different formats. This introduces more error prone steps to the appliance process and requires more validation and

testing of the final images. Then, what happens when you need to update all those image formats?

Simple snapshot tool: Snapshot tool vendors typically target only one or two virtualization platforms. Creating an appliance with one of these developer kits limits your potential market or forces you to convert the image manually after the fact. Simple snapshot tools do not allow you to create images suitable for installation on physical hardware.

rBuilder: rBuilder automates the generation of multiple virtual appliances by supporting virtualization formats from VMware, Microsoft, Citrix/XenSource, Amazon EC2, Parallels, Red Hat, Novell, and Oracle. rBuilder uses your appliance definition to create all these virtual images, as well as appliance images suitable for installation on physical hardware. The use of one appliance definition ensures you get the same appliance regardless of the end format. rBuilder also knows to automatically include format-specific tools or files each time the appliance is updated. So a Xen appliance will only receive Xen tools and not VMware tools.

Automated Maintenance Process

Most appliances won't be used just once and thrown away. If you're producing appliances for commercial use, then being able to track and incorporate changes over time is crucial.

General purpose OS: Maintaining an appliance built from a stripped down operating system requires determining which pieces have been updated and integrating those updates into the appliance image. In the absence of a repository, this involves a manual tracking effort that could span multiple code silos across the enterprise. Is engineering time really well-spent looking for code changes each time an appliance is rebuilt?

Simple snapshot tool: These offerings don't currently include a tracking capability either. Just as with a general purpose OS, the process is manual, time-intensive and annoying. You're engineers will be sending a lot of emails to find those updated pieces of code.

rBuilder: All appliance components, builds and versions are stored in rBuilder's centralized repository. Maintaining an appliance is as simple as adding the updated code to the repository. All appliance definitions that include that component will account for the update when the appliance is rebuilt. This process is seamless to the developers; they see only the most current version when they need it.

rPath Appliance Platform - An Ideal Appliance Foundation

Web-based Appliance Management

As an appliance provider, you want the easiest possible way for your end users to interface with the appliance to perform maintenance, set configurations and establish network connections. Graphical management tools are preferred over using a command-line for interaction with an appliance.

General purpose OS: You must develop your own web-based interface, or instruct end users on how to use the command line for maintenance and configuration tasks. Do you have adequate resources to devote to interface development or user training?

Simple snapshot tool: These tools may offer a web-based management interface that lets you control and configure appliance settings. However, these interfaces cannot be branded or extended to create a unique appliance experience for the end customer.

rPath Appliance Platform: The rPath Appliance Platform is a set of services that provide common functions needed in any appliance. It is also extensible and can be customized to provide an interface that matches the needs of your specific appliance. In particular, the Configuration Service provides a web-based management console to adjust appliance configurations such as networking and services; date and time; logging, status and notification; and users and passwords. No interaction with the command line is necessary.

Complete Online Updates

Software updates can be complex migrations, total reinstalls or ongoing incremental hassles as component changes in one area break another area. Software appliances minimize some of this problem by making the provider responsible for the integrity of the entire software stack. However, deploying updates to field appliances can still be difficult.

General purpose OS: You must develop an update capability yourself or choose an existing online update manager where you populate a server with the correct update packages. Unfortunately, these systems do not support file level updates. This means that if you change one file in a component, the entire component must be sent to the appliance. This can cause large and unwieldy update packages, perhaps even to the point of having to replace the whole appliance. The lack of an update capability also means that deploying the updates to field appliances will require manual intervention because a user must download and install the updates, test the appliance and hope it still works.

Simple snapshot tool: Snapshot tools available today don't currently include an update capability. Your engineers will be spending time installing updates and coaching users through the process.

rPath Appliance Platform: The Update Service allows you to maintain deployed appliances by facilitating the connections needed to check for updates. All updates to any components on the appliance are downloaded and applied in a single transaction, eliminating the problems and potential downtime that could occur if components get out of sync. The Update Service optimizes time and bandwidth by using file level differentials - dramatically reducing the update file size to 10% of the size of a typical software update. Incremental updates can also be performed from an update CD or DVD created by rBuilder for “dark” or disconnected sites. The appliance provider has complete control over the update process, thereby minimizing the risk of failure and reducing support incidents.

Rollbacks and Fail-safe Updates

Even under the best circumstances, errors can occur that compromise the integrity of a software update. In that case, it could become necessary to revert to a known good state to maintain functionality. Rollback is the process of reversing an update step-by-step until you reach that known good state. Fail-safe functionality gets you back to a prior image in one transaction.

General purpose OS: Unless you develop your own rollback capability, you will have to rely on the capabilities of the underlying packaging system; typically RPM or dpkg. Neither of these packaging systems is capable of update rollbacks. Nor are they capable of reverting to the prior appliance image. You’d have to reinstall it from scratch.

Simple snapshot tool: Since all the snapshot tool offerings are based on RPM or dpkg, they also do not have rollback capabilities. The best alternative to a fail-safe function would be to snapshot the virtual appliance before each update - a manual process that your customers would have to perform.

rPath Appliance Platform: If the end user of an appliance encounters problems with an update, the rPath Update Service can rollback the update. This is accomplished by storing a journal of the changes made during an update and reversing those changes during the rollback. Rollbacks done in this fashion are robust and preserve the integrity of the appliance.

Similarly, an appliance built with rBuilder can be configured to accommodate multiple appliance images. With the fail-safe features, when an image update is applied, the Appliance Platform automatically configures the appliance to boot into the new image. If the new image does not perform as expected, the appliance returns to the prior image.

Appliance Backups

We hardly need to instruct you on the importance of backups, both for applications and data. You must do it. Period.

General purpose OS: You must develop your own backup functionality or license a third party offering. Again, this requires development time and resources or management of a third-party relationship and support of that product in the software stack.

Simple snapshot tool: These tools don't currently include a backup capability. The best alternative would be to snapshot and archive images of the appliance, a manual process for your customers.

rPath Appliance Platform: Hardware or other system failures can render an appliance unusable. The Backup Service allows the appliance to push critical configuration and application data onto permanent storage outside the appliance. The backup process includes executing scripted actions that can, for example, perform a database backup operation. In the case of a critical system failure, the appliance can be restored from this backup.

Entitlement Management

If you are selling a commercial appliance, you want customers to pay for it before they receive access to appliance functionality, maintenance and updates. Similarly, if they don't pay, you may want to restrict their access and not provide maintenance and updates.

General purpose OS: You must develop your own entitlement functionality or license a third party offering. As with backup, using a third party offering for entitlement will introduce another disjointed management function for deployed appliances.

Simple snapshot tool: These offerings don't currently include entitlement capabilities. You must devote time and resources to developing (or buying) a solution.

rPath Appliance Platform: The Entitlement Service gives you a ready solution for license and maintenance management and enables a subscription based business model for your appliance. At first deployment, the Entitlement Service provisions a license key. It then checks before each update that the appliance still has a valid entitlement. Various actions can be taken if an appliance entitlement is not valid, such as cutting off access to future appliance updates or disabling the appliance. As the appliance provider, you have centralized control over the appliances in the field.

Summary

We think these arguments are pretty compelling justification for building your appliance with rPath's rBuilder and rPath Appliance Platform. Faced with the alternative of building an appliance in-house or using another method, rPath's solutions are just good sense. Why spend time developing in-house expertise and technology that rPath already spent years in the making?

As you move forward with your appliance strategy, remember to consider how you will address the following:

- Repeatability and consistency of appliance builds
- Supporting multiple appliance formats
- Providing maintenance and updates to field units
- Interfacing with the appliance to perform administrative tasks
- Provisioning and managing licenses

rPath's solutions address these concerns and more. But if you need more convincing, give us a call for a demonstration. Or, try building an appliance yourself - we think you'll quickly realize the advantages of doing it the rPath way.



1.866.508.6200
info@rpath.com

For software companies that want to accelerate license growth, expand into new markets, and reduce support and development costs, rPath's rBuilder transforms applications into virtual appliances. A virtual appliance is an application combined with just enough operating system (JeOS) for it to run optimally in any virtualized environment. Virtual appliances eliminate the hassles of installing, configuring and maintaining complex application environments. Only rPath's technology produces appliances in multiple virtual machine formats, simplifies application distribution, and lowers the customer service costs of maintenance and management. The company is headquartered in Raleigh, North Carolina.

© Copyright 2008, rPath, Inc. rPath, rBuilder and the rPath logo are registered trademarks of rPath, Inc. All other trademarks are property of their respective owners.