

WHITE PAPER

# Best Practices for Building Virtual Appliances



**Table of Contents**

Objectives..... 3

Design Principles ..... 3

Completely Encapsulated..... 3

Optimized Virtual Disk and Operating System ..... 4

Initial Configuration..... 4

'Standard' Components..... 5

Command Line Interface ..... 5

SSH Support..... 5

Web Management Interface ..... 5

External Management..... 5

Virtual Machine Configuration ..... 6

Updates and Ongoing Maintenance ..... 6

Packaging and Publishing ..... 6

## Best Practices for Building Virtual Appliances

A virtual appliance is a pre-installed, pre-configured operating system and software solution delivered inside a virtual machine. Deploying a software solution as a virtual appliance enables you to build a complete turnkey package that customers are able to download and immediately deploy. Thus, customers skip the time-consuming and often support-intensive task of installing and configuring the appliance. This lets customers focus all their energies on trying or using your solution rather than struggling to get it to run.

This document describes the best practices for building a virtual appliance. It covers high level design principles as well as low level details for building virtual appliances ready for certification under the VMware Certified Virtual Appliance program. In turn, virtual appliances built according to these standards will allow your customers or prospective customers to test or use your virtual appliance with all the VMware virtualization platforms.

### Objectives

- Deliver ready-to-run solutions to your customers and prospective customers.
- Package your software with an operating system and services optimized and configured to provide only the functionality required by your solution.
- Decrease support costs associated with traditional software installation and configuration.
- Enable customers to run your solution on their standard hardware.
- Allow customers to leverage VMware Infrastructure 3 to provide management, fault-tolerance and continuity services without having to install additional agents inside your solution.

Virtual Appliances represent an evolutionary step in the way ISVs deliver solutions to their customers. They enable Enterprise Software as a Service (ESaaS) that can be deployed instantly.

60-70% of support calls relate to initial installation and configuration of a software application

### Design Principles

#### *Completely Encapsulated*

A virtual appliance contains all of the components required to run your solution on top of a virtualization layer. Since virtual appliances are designed to run a specific solution, customers should not need root access to the operating system inside your appliance. Typically, users get limited access to a console-based command line interface with either a limited set of commands or an interactive menu. In some cases access to the console is not permitted at all (nor required). Ideally, end-users will only get to configure a virtual appliance through a browser-based interface. Wrapping configuration files inside a Web interface ensures that customers do not corrupt those files and will ensure that your virtual appliance is always configured with expected values.

Locking the user out of the operating system that drives your virtual appliance means that you need to provide the complete solution stack. If your appliance requires a database, it should either be embedded inside your solution (pre-installed and pre-configured to run your virtual appliance) or you should provide a mechanism for connecting to an external database and setting that database up with the schemas required by your solution (via ODBC or JDBC). Similarly, your appliance should provide support for tools and technologies typically used by enterprise-class server solutions. You should support internal and external (LDAP) user stores, internal and external (remote syslog) logging, and internal and external (SNMP, NTP) management and monitoring tools. In the default case, your appliance should be able to run in an isolated mode. Nonetheless, for production deployments, customers may prefer to externalize certain aspects of your solution (database, logging, security and/or management). Typically, your solution should be flexible enough to support externalization of services like these to support a customer's need to scale your solution or to centralize data or log information to comply with their internal data backup/retention/security policies.

When users opt to run your virtual appliance in a completely self-contained form, you should provide services that allow them to manage the disk requirements of the solution. It is imperative to provide facilities to export and purge log files and database tables to ensure that the appliance does not fill up its virtual disk. A nice option for shipping smaller virtual appliances is to allow users to add an additional virtual disk, sized according to their need (possibly affording them longer online access to log files or other features of your solution). Your appliance can check the virtual hardware for the presence of a new, unformatted virtual disk and take advantage of that disk for user files (you can format the new, unformatted virtual disk and add it to your root file system so your appliance can write its log files or store its database on that resource).

### Optimized Virtual Disk and Operating System

The basic design principles used for building a virtual appliance are similar to those used to build physical appliances. The major difference between physical and virtual appliances is that a virtual appliance is not delivered inside a piece of server hardware. As such, the developer of a virtual appliance will typically take additional steps to optimize the entire stack so it only contains the absolute minimum footprint required to deliver the desired functionality. Physical appliance developers are less concerned with disk size and will often ship a “fatter” OS because they are shipping a server that typically has a big enough physical disk to install the largest operating systems without any concern for space. Because virtual appliances can be packaged for download over the Internet, it is best to decrease the total footprint of the solution.

The effort required to optimize the footprint has additional benefits. Shipping less code inside a solution typically means that the solution is more secure—less code means fewer possible bugs and vulnerabilities. Further, a virtual appliance with an optimized OS will require fewer patches (no need to apply an OS patch that addresses a security flaw in a Web browser if your server-class solution doesn't even include those modules). Rather than building your virtual appliance on a heavy-weight, general-purpose OS, your virtual appliance should ship with “Just enough OS” to support your solution. This concept of JeOS (pronounced “juice”) is typically achieved with a modular OS like Linux. Nonetheless, there are tools that can be used to tailor a Microsoft operating system down to a light-weight JeOS. An extreme example of JeOS can be seen in BEA's Liquid VM. The BEA Virtual Appliance deploys the BEA J-Rocket Java Virtual Machine on a custom thin layer that translates low-level system calls made by the JVM to instructions that are directly executed by the VMware virtual machine. The JVM is no longer handicapped by a general purpose operating system that has not been optimized to run Java applications.

Part of optimizing the operating system is to ensure that only those services and ports required by your solution are used. If your solution does not need to allow external users to write data to its file system, services like an FTP daemon should not be included and the ports typically used by this service should not be opened. Typically, your appliance will open a port so external users can connect to the Web management interface of your appliance (port 80 and/or 443 for standard HTTP traffic or some solution specific port as desired). If your appliance supports a command line interface, you should consider supporting SSH to provide secure, remote access to that interface (run an SSH daemon and open the corresponding port – typically port 22).

### Initial Configuration

Ideally, ongoing use of your solution will be “lights out”. Once your virtual appliance is setup, users should not have to manage it on a regular basis (“set it and forget it”). As such, it is important to think through the first boot experience of your virtual appliance. You should allow users to perform all the necessary configurations the first time they turn on your virtual appliance. The appliance itself should support DHCP for getting an IP address and should still boot if a DHCP server is not found (allow fail-over to a static IP address that can be reached by the host). At a minimum, the virtual appliance console should print a message for the user that displays the URL for the management interface (<http/https + IP address + port>).

Once the virtual appliance has an IP address, the user will be able to connect to the Web management interface you provide. Your virtual appliance should support the initial configuration process through either the command line interface or the Web management interface (or a combination). Some of the initial configurations your appliance should support are:

- Establish a password for an administrative account (ideally not the root account)
- Change the network configuration for your virtual appliance (change from DHCP to static IP and set the hostname for the appliance)
- Setup the SNMP configuration for managing and monitoring your virtual appliance
- Setup the log levels and log-rotation schedules for the various services your virtual appliance supports
- Generate an SSL certificate request that can be used to generate a new SSL certificate for your virtual appliance
- Upload a new SSL certificate to secure communications to and from the virtual appliance
- Change the system date and time or connect with an NTP server
- Reset to “factory” original settings

## 'Standard' Components

While each virtual appliance will have its own primary application that needs to be configured and maintained, there are some standard components that all virtual appliances should have. These standard components may be provided by a pre-packaged Virtual Appliance Framework or may be custom code that your developers write. Among these standard services are:

- Just Enough OS (JeOS as discussed above)
- Command line interface (CLI)
- SSH support
- Web management interface (WMI)
- External management support (SNMP or XML SOAP interface)

### Command Line Interface

Most virtual appliances will not provide a GUI as they provide some kind of server-side functionality. As such, the console experience with a virtual appliance will likely be a simple character-based interface. Rather than leave the user to a standard Linux shell, it is advisable that you provide a custom CLI. You can either provide custom grammar that parses commands that are specific to your application or a menu interface where users select among the functions your CLI supports. Some of the functions you may want to provide commands for are:

- **Network Configuration:** Allow the user to change from dynamic to static IP, set the virtual appliance host name, add static routes, etc.
- **Network Display:** Allow the user to query the appliance for its network address and detailed network information.
- **SSL Setup:** Generate certificate requests, upload new certificates, and generate self-signed certificates.
- **Process Monitoring:** Your own version of 'top' or similar utility
- **User Account Management:** Reset passwords, add users, and/or connect the appliance to LDAP.
- **Lifecycle Operations:** Power off, reboot, restart network, etc.

### SSH Support

If your virtual appliance supports a rich CLI, it is advisable to allow administrators a secure way of accessing that CLI remotely (so they don't have to be at the console itself). Running an SSH daemon is the simplest way of allowing secure remote access to your CLI.

## Web Management Interface

While your virtual appliance may support a robust CLI that allows users to do everything they need to configure, manage and monitor the appliance, most users prefer to configure a virtual appliance with a Web browser. Thus, your virtual appliance should provide a WMI that allows users to graphically set up the virtual appliance configuration, maintenance and monitoring functions. Some of the popular Web-based functions include:

- **Network Configuration.** Allow the user to change from dynamic to static IP, set the virtual appliance host name, add static routes, etc.
- **Network Display.** Allow the user to query the appliance for its network address and detailed network information
- **SSL Setup.** Generate certificate requests, upload new certificates, and generate self-signed certificates
- **Process Monitoring.** A graphical version of "top" that shows load on your appliance and may provide a live histogram showing various performance data
- **User Account Management.** Reset passwords, add users, and/or connect the appliance to LDAP
- **Lifecycle Operations.** Power off, reboot, restart network, etc.
- **Log Management.** View, purge, export logs for the various services running on your virtual appliance
- **NTP Setup.** Configure the NTP client to synchronize time settings from a specific NTP server
- **Solution Configuration.** Pages that control the actual application you are providing as a virtual appliance; often your application will have its own set of custom configuration options that users need to setup (Providing a Web-based mechanism for editing those configuration options will make it easy for you to guide your users through the process.)

### External Management

While WMI provides an excellent way to configure, manage and monitor virtual appliances, users often prefer to use standard management tools to manage all of their applications and appliances. As such, your appliance should support SNMP. Your appliance should also be able to send messages to an SNMP listener, support MIB II based messages that external tools can use to query the state of your appliance and make supported configuration changes or generate lifecycle requests (reboot, power off, etc.) Modern applications often support these same capabilities through an XML SOAP based interface, which allow users to write their own management applications in tools that they prefer.

## Virtual Machine Configuration

Since a virtual appliance is built inside a virtual machine, it is important to create and configure the virtual machine according to best practices. Most virtual appliances are built to deliver a specific server-class application. As such, the virtual machine itself does not need access to a wide variety of virtual hardware. It is best practice to remove any virtual hardware that your appliance does not need (floppy, CD, USB, etc.). If you do choose to support devices like virtual CDs (for updates to your appliance), it is best to configure your appliance to start with those devices disconnected.

Additionally, there are some best practices that relate to the portability and performance of your virtual appliance. The most critical of these is that your virtual appliance should be build using SCSI virtual disks. It is also important that your virtual appliance includes VMware Tools. VMware Tools provides optimized drivers for VMware virtual hardware and management tools that can monitor and manage the virtual appliance with VMware VirtualCenter.

## Updates and Ongoing Maintenance

Rather than having your customers patch their own OS, you patch and test the OS along with your appliance. This gives you the control to schedule and test patches and lets you decide which ones are critical (or even relevant) for your appliance. By controlling the entire process, you decrease the burden on your support organization if customers call with issues. In addition, your customers will get an inherently more reliable appliance because the entire stack of the virtual appliance has been through your QA process. With more traditional software, your customers patch the OS according to their own standards and without regard for the potential impact on your application. In turn, your support team is on the hook for figuring out what caused a specific problem. You can see how although "owning" the OS appears to be a cost, in the long run, spread out over multiple customers, retaining ownership is a benefit.

There are a few things to consider when providing patches for your appliance. Ideally, your appliance should be able to apply a patch file with minimal disruption and downtime. Properly configured virtual appliances should enable users to keep the appliance up-to-date with the latest application and operating system patches. Your appliance should support the ability to connect to an online repository to query for new patches (in the event that the appliance is configured with proper network access to support this model) or to an offline repository (when the appliance is not given Internet access). You may even want to deliver your patch repository as its own mini-virtual appliance that customers can install inside their network. Regardless, downloading and applying a patch to a virtual appliance should be something that users can do from the virtual appliance's Web management interface.

## Packaging and Publishing

Once you have built and tested your virtual appliance (for evaluations and/or production use), you can package it and make it ready for download. Start by copying the virtual machine into a clean folder. Remove any unnecessary files (typically you only need a .vmx and .vmdk files) and add a Getting Started Guide to the directory. When you have finished cleaning up the folder, use a compression tool to package the folder into a single file (.zip and/or .tar). The file is now ready to post for downloading. For security reasons, you may want to consider generating SHA1 signatures for each file inside your ZIP (or for the ZIP file itself). This will allow end-users to verify the integrity of the files they download from your site.

Once your virtual appliance is packaged and ready for download, you may list it on the [VMware Virtual Appliance Marketplace \(VAM\)](#). Each appliance receives its own unique page that provides general information and allows visitors to access your appliance.

If you have incorporated the best practices for creating a virtual appliance presented here and on the VAM, you should submit your virtual appliance to our labs for certification. After certification, you can advertise your appliance as a “VMware Certified Virtual Appliance”, with an accompanying VMware certified logo. Customers look for this logo to ensure that the virtual appliance was build according to VMware best practices and supported for production use. Details of the VMware Certified Virtual Appliance Program can also be found on the VAM.

Finally, once you have built and published your virtual appliance, enjoy the shortened sales cycles and simpler customer support!





VMware, Inc. 3401 Hillview Ave Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 [www.vmware.com](http://www.vmware.com)

© 1998-2007 VMware, Inc. All rights reserved. Protected by one or more U.S. Patents Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022, 6,944,699, 6,961,806, 6,961,941, 7,069,413, 7,082,598, 7,089,377, 7,111,086, 7,111,145, 7,117,481, 7,149, 843 and 7,155,558; patents pending. VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

