Guide to Building and Implementing Virtual Appliances

VMware Studio 1.0

Guide to Building and Implementing Virtual Appliances
Item: EN-000013-01

You can find the most up-to-date technical documentation on the VMware Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About This Book

This *Guide to Building and Implementing Virtual Appliances* provides information about developing virtual appliances using VMware® Studio 1.0. This book is intended for VMware developers who are creating, testing, distributing, and revising virtual appliances created by VMware Studio.

Revision History

This book, *Guide to Building and Implementing Virtual Appliances,* is revised with each release of the product or when necessary. A revised version can contain minor or major changes. Table 1 summarizes the significant changes in each version of this book.

Table 1. Revision History

Revision	Description
20080905	First version of the Guide To Building and Implementing Virtual Appliances 1.0.
20090205	Similar version with minor corrections throughout the text.

VMware provides several different SDK products, each of which targets different developer communities and target platforms. To view documentation for the various SDK products, and the current version of this guide, go to http://www.vmware.com/support/pubs/sdk_pubs.html.

Intended Audience

This document is intended for independent software vendors (ISV) and hardware appliance vendors (HAV) who are creating virtual appliances using VMware Studio. For information on deploying virtual appliances with VMware Studio, see the companion manual, *Guide to Building and Implementing Virtual Appliances*.

Document Feedback

VMware welcomes your suggestions for improving our documentation. Send your feedback to docfeedback@vmware.com.

Technical Support and Education Resources

The following sections describe the technical support resources available to you. To access the current versions of other VMware books, go to http://www.vmware.com/support/pubs.

Online and Telephone Support

To use online support to submit technical support requests, view your product and contract information, and register your products, go to http://communities.vmware.com/community/developer.

Support Offerings

To find out how VMware support offerings can help meet your business needs, go to http://www.vmware.com/support/services.

VMware Professional Services

VMware Education Services courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. Courses are available onsite, in the classroom, and live online. For onsite pilot programs and implementation best practices, VMware Consulting Services provides offerings to help you assess, plan, build, and manage your virtual environment. To access information about education classes, certification programs, and consulting services, go to http://www.vmware.com/services.

Overview of VMware Studio 1

This chapter provides an overview of VMware Studio, and includes the following topics:

- "VMware Studio Interfaces" on page 7
- "Virtual Appliance Management Infrastructure (VAMI)" on page 9

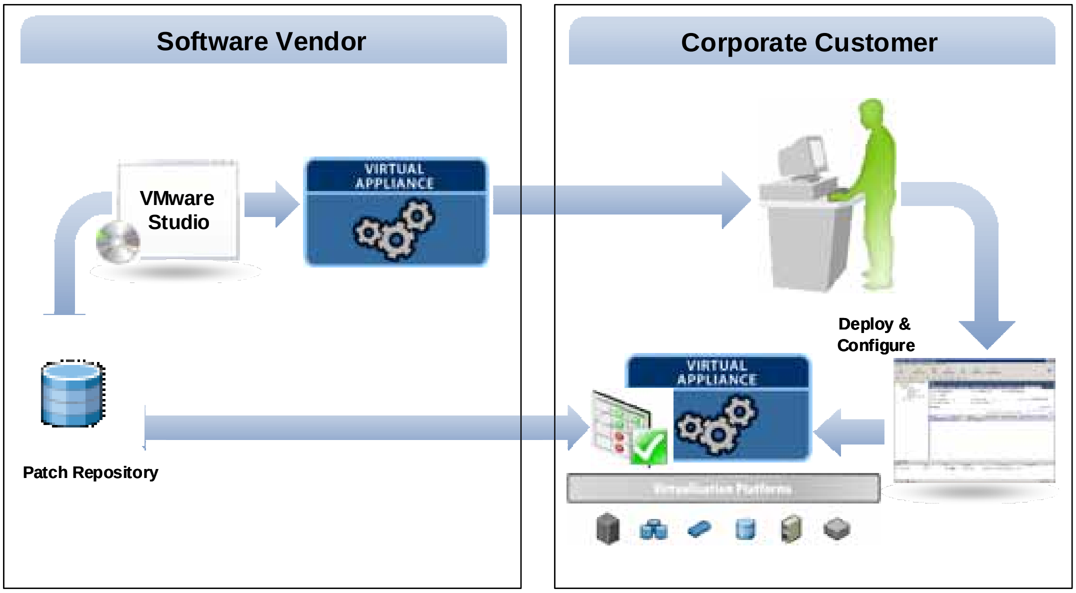
VMware Studio Interfaces

A virtual appliance is pre-built software containing one or more virtual machines that are packaged, updated, maintained, and managed as a unit. Customers can easily install and deploy these pre-integrated solution stacks in a virtual environment. Virtual appliances simplify application development, distribution, and management.

VMware Studio helps you to configure, build, deploy, customize, and maintain virtual appliances. Independent Software Vendors (ISVs) and hardware appliance vendors (HAVs) can use VMware Studio to build and maintain production-ready virtual appliances for their applications.

Figure 1-1 illustrates the workflow of using VMware Studio. A software vendor assembles and packages the virtual appliance, then distributes the virtual appliance to end users, such as corporate customers. Users can deploy the virtual appliance on their own virtualization platform, such as VMware ESX or VMware Workstation, and run the applications packaged in the virtual appliance. You can also include an optional update repository URL into the appliance so that thevirtual applicance can check for updates to your software.

Figure 1-1. Overview of VMware Studio



VMware Studio provides both a Web interface and a command-line interface to build virtual appliances. The Web interface offers ease of use and online help. The command line interface can integrate with of existing source control or configuration management systems for fully automated builds of virtual appliances.

VMware Studio is itself a virtual appliance built using its own framework. See Chapter 2, "Running VMware Studio," on page 11 for information about installing the VMware Studio virtual appliance and using it to create new virtual appliances.

VMware Studio appliances allow integration with VMware vCenter Server (formerly VirtualCenter) and the VMware Update Manager.

Virtual appliances created with VMware Studio are:

- Highly portable among virtual machines.
- Reproducible due to ordered specification of standard RPM or DEB packaging.
- Field-updatable, when an update repository is enabled.

Not all virtual appliances are interchangeable with those created by VMware Studio. Some virtual appliances are merely customized virtual machines packaged in a compressed format. VMware Studio can manage and update only those virtual appliances that were created with VMware Studio.

To build a virtual appliance with VMware Studio

1. Define a virtual appliance with the features you want.

   VMware Studio allows you to configure a variety of options for your virtual appliance, using either a Web-based graphical interface or a command-line interface. VMware Studio stores the definition of a virtual appliance in a build profile.

   These are a few examples of the many items you can configure in the build profile for your virtual appliance:

   - Select the operating system to include in the virtual appliance.
   - Select the application packages to include in the virtual appliance.
   - Configure the welcome screen.
   - Provide application and vendor information to include in the virtual appliance.
   - Configure first boot and subsequent boot scripts. For example, if you would like to pre-seed the application with metadata, you can achieve this by supplying a first boot script.
   - Configure virtual disks or virtual network cards for the virtual appliance.

   Currently, VMware Studio allows you to build single virtual machine virtual appliances.

2. Build the virtual appliance.

   VMware Studio relies on VMware Server 1.0 to pull together the files required to build your virtual appliance. The process of constructing the virtual appliance is known as provisioning.

   To provision the virtual appliance, VMware Studio and VMware Server communicate by way of an SSH connection. Therefore, an SSH daemon on the VMware Server host is a requirement of VMware Studio. See "Requirements" on page 12 for information about deployment options of VMware Studio.

   VMware Studio can produce virtual appliances in either Open Virtualization Format (OVF) or ZIP format. OVF files conform to the OVF 0.9 draft specification. For information about the OVF standard and available tools, see http://www.vmware.com/appliances/learn/ovf.html.

3. Deploy the virtual appliance to the desired systems.

   Once your virtual appliance is created and tested, you can distribute the VMware Studio output files to your customers. A virtual appliance can be deployed either on VMware's hosted platforms, such as VMware Workstation and VMware Server, or on VMware ESX/ESXi hosts.

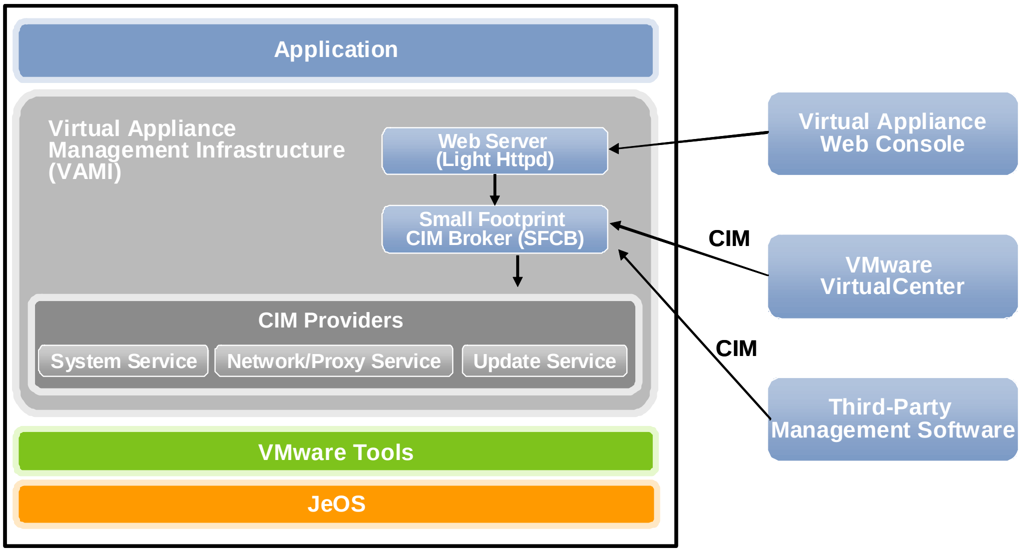
4. Maintain and update the virtual appliance.

Once your virtual appliance is deployed, end users can configure and manage the appliance from the VMware Studio Web console or from VMware vCenter Server. If an update repository is provided, end users can also check for and install updates through the Web console.

Virtual Appliance Management Infrastructure (VAMI)

Virtual appliances built by VMware Studio carry an in-guest management component, called the Virtual Appliance Management Infrastructure (VAMI). All management services provided by VAMI are implemented as Common Information Model (CIM) Providers using Small Footprint CIM Broker (SFCB) as the CIM Manager. The CIM standard is from Distributed Management Task Force (DMTF); see http://www.dmtf.org for details.

Figure 1-2. VMware Studio Components


Application
Virtual Appliance Management Infrastructure (VAMI)
Web Server (Light Httpd)
Small Footprint CIM Broker (SFCB)
CIM Providers
System Service
Network/Proxy Service
Update Service
VMware Tools
JeOS
Virtual Appliance Web Console
CIM
VMware VirtualCenter
CIM
Third-Party Management Software

VAMI includes the following components:

- **Web-based Management Console** – Allows end users to configure, manage and monitor their virtual appliances. For example, users can change network settings, check for and install updates, and change update settings to enable automatic installation of updates. The web interface is an AJAX application and communicates with the management agent using CIM APIs. A lightweight resource-constrained Web server (lighttpd) is installed in the virtual appliance to host the Web interface application.
- **VAMICLI** – Provides a command line to configure, manage, and monitor virtual appliance.
- **Small Footprint CIM Broker (SFCB)** – CIM manager, acts as a management agent for virtual appliances.
- **Published CIM APIs** – Integrates VMware Studio services with central network management software such as Tivoli and HP Openview.
- **Management Services** – Implemented with CIM providers:
  - Update service – Simplifies the maintenance of a virtual appliance by automatically or manually applying updates from a remote update repository. VMware Update Manager integrates with the update service to allow centralized management of virtual appliance updates.
  - Network service – Allows end users to configure their network and proxy settings.
  - System service – Provides basic system information, and shuts down or reboots the virtual appliance.

Running VMware Studio 2

This chapter describes how to install and configure VMware Studio, and contains the following topics:

Basic Configuration of VMware Studio

VMware Studio is distributed as a virtual appliance, similar to the virtual appliances you will create using VMware Studio. The VMware Studio virtual appliance uses Ubuntu 7.04 as its operating system. To use VMware Studio, you must deploy the VMware Studio appliance into either VMware Workstation, VMware ESX/ESXi, or VMware Server.

Figure 2-1. A Basic Deployment of VMware Studio

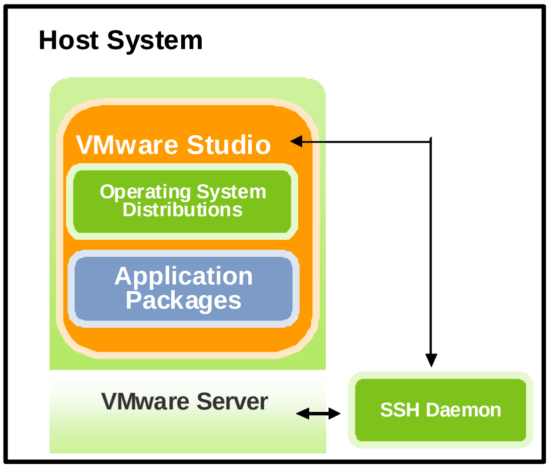


Figure 2-1 illustrates a typical configuration of VMware Studio. For simplicity of configuration, you can deploy VMware Studio within VMware Server. The SSH daemon should reside on the same host system as VMware Server to communicate and transfer files between VMware Studio and VMware Server during the provisioning operation when building virtual machines. For better performance during provisioning, you can store the application packages and the operating system distribution media in the VMware Studio appliance's file system. Alternately, the operating system distribution media (the ISO image) can be mounted by way of NFS from within VMware Studio. This facilitates sharing the ISO image between multiple instances of VMware Studio.

Requirements

To use VMware Studio, the following are required:

- VMware Server 1.0.4, 1.0.5, or 1.0.6 installed and running on a host in your environment. VMware Studio requires VMware Server 1.0 to provision virtual appliances. Afterwards, a virtual appliance built with VMware Studio can be deployed on any of the supported virtualization platforms.
- The VMware Studio appliance software. Obtain this software from the VMware website, or from a copy on physical media such as CD or DVD. Although the default is 512MB, for performance reasons it is best to allocate 1GB of virtual memory for the VMware Studio virtual machine.
- An SSH daemon running on the host where VMware Server is installed. If you are running VMware Server on Linux, then ensure that SSH is enabled. If you are running VMware Server on Windows, then see Appendix A, "Installing SSH on Windows," on page 37.
- Access to the operating system distribution media (an ISO image) for the guest OS of your virtual appliance. For improved build speed, it is best to download and store the OS distribution on a file system of the VMware Studio appliance. See "Operating System Distribution Media" on page 19 for more information on supported operating systems.
- Either Internet Explorer 6, Internet Explorer 7, or Firefox 2. You use this browser to access the Web-based graphical user interface of VMware Studio.
- If a firewall is between the VMware Studio appliance and its provisioning engine (VMware Server), VMware Studio should have the following ports open:
  - 22 ssh tcp
  - 69 tftp udp
  - 80 http tcp
  - 5480 https tcp

  Also in the case of a firewall, VMware Server should have the following ports open:
  - 7 echo tcp, udp
  - 22 ssh tcp
  - 902 vix tcp (If 902 was in use at install time, VMware Server suggests port 904 instead for VIX.)

Deploying VMware Studio

The simplest configuration is to deploy VMware Studio into VMware Server, as follows:

1. On the host where VMware Server is running, extract the VMware Studio appliance software (ZIP file) into a directory. For example on a Linux host, **/var/lib/vmware/VirtualMachines/VMware_Studio-1.0.0**.
2. In the VMware Server Console application, choose **File > Open** from the menu bar.
3. Click **Browse** and navigate to the directory where you extracted the VMware Studio software.
4. Select **VMware_Studio-1.0.0.vmx** and click **Open**.

   VMware Studio appears in the Inventory pane.
5. Click **Start this virtual machine** in the Commands section.

   VMware Studio displays its startup messages. When you first load VMware Studio, you are presented with the End User License Agreement. Press the **Space** bar after each page of the license displays. At the end of the license agreement, accept the agreement by entering **yes** and pressing **Enter**.

   Once you accept the license agreement, you will be asked to enter a new password for the root account. Enter a password, then type the password again at the second prompt.

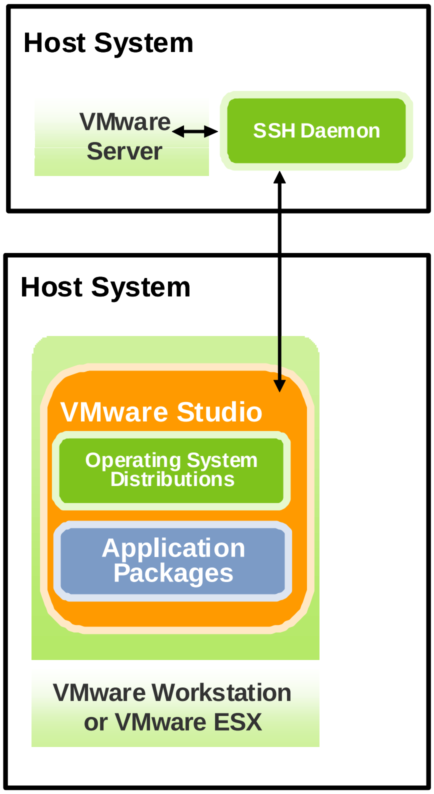
   NOTE VMware Studio requires the root (superuser) account to build virtual appliances.

6 The VMware Studio welcome screen displays. By default, VMware Studio tries to acquire an IP address from a DHCP server. If you prefer a static IP address, see "Configuring Static IP Addresses" on page 14. Take note of the IP address that appears in the sentence "To manage your appliance please browse to https://<IP.address>:5480." You will use this URL to access the VMware Studio Web console.

Other Configurations

VMware Studio can be deployed on VMware ESX Version 3.0 and later or on VMware Workstation Version 6.0 and later. To deploy VMware Studio on ESX/ESXi, use the OVF version of the VMware Studio appliance. The recent VI Client provides a dialog, **Deploy OVF Template**, to guide you through importing and deploying VMware Studio. See the VMware ESX/ESXi documentation for more information.

Figure 2-2. VMware Studio Remote Host Configuration



To deploy VMware Studio on:

- **ESX/ESXi 3.5** – Use the VI Client installed from the ESX/ESXi host to import the OVF template of the VMware Studio virtual appliance into the host.
- **ESX 3.0.x** – Use VMware Converter 3.0.3 or the OVF Tool to import the VMX format of the VMware Studio virtual appliance into the ESX host.
- **Other product platforms** – Download the virtual appliance in ZIP format. In the **File > Open** dialog, locate and run the **VMware_Studio-1.0.0.vmx** file, as in "Deploying VMware Studio" on page 12.

NOTE VMware Server is required for provisioning virtual appliances, but you can run VMware Studio on an ESX/ESXi host or on VMware Workstation and provision over the network using an SSH connection.

Modifying the Network and Time Zone Settings

After installing VMware Studio, you can configure several options. Use the VMware Server Console tab for VMware Studio to specify configuration options, as described in the following sections. The welcome screen displays the following menu selections at the bottom of page:

```
Login
Configure Network
Set Timezone (Current: UTC)
```

Choosing Between DHCP and Static Network Addresses

When the VMware Studio appliance starts, the appliance attempts to discover a DHCP server on the network to request a temporary IP address. Because the address is temporary, it might be different each time the virtual appliance starts. Most server appliances should be configured to a static address that remains the same after the virtual appliance is restarted. This allows all users of the virtual appliance to be certain of its network address (and associated host name) so they can predictably connect to it.

Configuring Static IP Addresses

By default, VMware Studio retrieves its network address from a DHCP server. To configure a static network address for a virtual appliance, perform the following procedure:

1 From the virtual appliance welcome screen, select **Configure Network** and press **Enter**.

2 Enter **no** to the "Use a DHCP server instead of a static IP address" prompt.

3 Enter the addresses for the following and press **Enter** after each entry:

- IP Address
- Netmask
- Gateway
- DNS Server 1
- DNS Server 2

4 Enter **yes** or **no** at the "Is a proxy server necessary to reach the Internet?" prompt.

If you answered no, press **Enter**.

If you answered yes, press **Enter**. Type the IP address and port number of the proxy server. Press **Enter**.

A review of your entries displays.

5 Type **yes** and press **Enter** if the values are correct.

If the values are incorrect, type **no** and press **Enter** to perform the procedure again.

Configuring DHCP

If you have configured the virtual appliance to use a static network address and you wish to change the configuration to retrieve its network address from a DHCP server, perform the following procedure:

1 From the virtual appliance welcome screen, select **Configure Network** and press **Enter**.

2 Enter **yes** to the "Use a DHCP server instead of a static IP address" prompt.

3 Enter **yes** or **no** at the "Is a proxy server necessary to reach the Internet?" prompt.

If you answered no, press **Enter**.

If you answered yes, press **Enter**. Type the IP address and port number of the proxy server. Press **Enter**.

A review of your entries displays.

4 Type **yes** and press **Enter** if the values are correct.

If the values are incorrect, type **no** and press **Enter** to perform the procedure again.

Setting the Time Zone

To change the time zone for a virtual appliance, perform the following procedure:

1. From the virtual appliance welcome screen, select **Set Timezone** and press **Enter**.
2. At the "Do you want to change that?" prompt, enter **y** and press **Enter**.
3. Enter the number of the area containing the time zone you want and press **Enter**.
4. Enter the name of the time zone you want and press **Enter**.

The time zone is set and the Welcome screen displays.

Accessing the VMware Studio Web Console

To start the VMware Studio Web console, perform the following:

1. In the section "Deploying VMware Studio" on page 12, the URL to access the VMware Studio Web console is displayed in Step 6 of the process. Enter that URL into one of the supported Web browsers.
2. Type **root** as the User name. Type the password you specified in Step 5 of the process "Deploying VMware Studio" on page 12. Press **Enter** or click **Login**.

Creating Virtual Appliances With the VMware Studio Web Console 3

This chapter documents use of the VMware Studio Web console.

Overview of Creating Virtual Appliances

VMware Studio supplies a Web console for creating the build profile and for building the virtual appliance. This section describes the general procedure for creating a virtual appliance. Subsequent sections describe each configuration option in detail.

To create a virtual appliance

1 Obtain the operating system distribution media (ISO image) for one of the supported operating systems. See "Operating System Distribution Media" on page 19 for information about obtaining operating system distribution media.

2 In the Appliances tab of the Web console, click **Create Appliance**.

3 In the Create a New Appliance dialog, specify a name for your virtual appliance, select a profile from the OS templates, and click **Create New Appliance**.

Pre-configured templates for each supported operating system are supplied with VMware Studio. Once you create your own virtual appliances, they become available as templates in this dialog for creating derived virtual appliances. Specifying a template from a previously created virtual appliance creates a new build profile with the same contents as the original template build profile.

4 The Editing screen appears, where you can edit the build profile of your virtual appliance.

The appliance editing screen organizes the build profile into the following tabs:

- **Description** – specifies general information about the virtual appliance. See "Description Tab" on page 20 for details.
- **Hardware** – specifies hardware configuration, such as CPU, memory, and disks, for the virtual appliance. See "Hardware Tab" on page 22 for details.
- **OS** – specifies information about the operating system for the virtual appliance. See "OS Tab" on page 22 for details.
- **Application** – specifies the application packages and operating system packages to load into the virtual appliance. See "Application Tab" on page 23 for details.
- **Output** – specifies the type of distribution format (OVF or ZIP) for the virtual appliance. Also, optionally specifies the location of an update repository for the virtual appliance. See "Application Tab" on page 23 for details.
- **Setup** – specifies setup of the build environment, such as connection information for VMware Server. See"Setup Tab" on page 25 for details.

You can use the **Next** and **Back** buttons to navigate between the tabs until all required fields are specified. If you attempt to navigate to the next page and a required field has not been specified, an error message appears in red next to the field, and you must fix the problem before continuing.

Each field in the build profile has a help icon to the right of the field (a question mark inside a circle). Clicking the help icon toggles between display of contextual help and a list of frequently asked questions.

There is also a set of navigation links above the build profile tabs. These navigation links allow you to go to previous screens in VMware Studio. Navigating away from the editing screen causes a dialog to appear asking if you wish to discard your current configuration changes and navigate to another screen.

5 Once you have configured your build profile, you can perform one of the following actions by clicking on the corresponding buttons on the right-hand side of the window.

- **Validate** – After completing your build profile, you can validate the configuration before building the virtual appliance. Click the **Validate** button on the right side of the window to check the following items, ensuring that the build profile is valid and will build properly.
  - Check the build profile schema.
  - Connect to the VMware Server through the VIX connection.
  - Check that SCP is available to the provisioning host, and available to the update repository host (the path must exist).
  - Check the appliance version syntax.
  - Check the appliance minimal memory size.
  - Check that the ISO URL is specified.
  - Check that user names and passwords are specified for the appliance.
  - Check that the appliance user name and password are not duplicates.
  - Verify accessibility of paths to the application logo, application packages, and update repository.
  - Check for duplicate disk IDs, disk filenames, and OS packages.
  - Check for valid minimum disk sizes.
  - Check for valid VMDK filenames.
  - Check other specifications, such as controllers, disks, partitions to ensure they are valid.
- **Save and Build** – Saves your build profile and displays the build summary dialog. To build your virtual appliance, click **Build Appliance** to begin the provisioning process. A progress icon displays and a build log is available to view the status of the build as it progresses.

- **Save and Close** – Closes the editing session and saves your build profile. The build profile appears in the Appliance List, and you can choose to edit or build the virtual appliance at a later time.
- **Close Without Saving** – Closes the editing session and discards any changes. A warning appears asking if you are sure you wish to exit the editing session without saving changes.

Operating System Distribution Media

VMware Studio supports multiple operating systems. To improve the speed of building virtual appliances, you should download and store the OS distribution media on the VMware Studio appliance.

The following is a list of supported operating systems, URLs for locating the distribution media, and MD5 checksums that you can use to ensure that the distribution media is valid.

- SLES 10 SP2

  http://www.novell.com/linux/

  SLES-10-SP2-DVD-i386-GM-DVD1.iso

  MD5: 2b91efeb8a62e2824302bf9269080252

- Ubuntu 7.04

  The Ubuntu 7.04 ISO image has been removed from the Ubuntu releases archive. Locate old CD media or contact the Ubuntu communities if you want to retrieve the ISO image.

  MD5: ca609edf086eea0c821ba34a5c0a709d

- Ubuntu 7.10

  http://cdimage.ubuntu.com/releases/gutsy/release/ubuntu-7.10-dvd-i386.iso

  MD5: b5d9aaa45af862b4c804530734216a15

- Ubuntu 8.04

  http://cdimage.ubuntu.com/releases/hardy/release/ubuntu-8.04-dvd-i386.iso

  MD5: 088b5d9e656dd89483dbb1b845b96fb9

- CentOS 5.0

  The CentOS 5.0 ISO image has been removed from the CentOS download archive. Contact the CentOS organization if you want to retrieve the ISO image.

- CentOS 5.2

  http://ftp.usf.edu/pub/centos/5.2/isos/i386/CentOS-5.2-i386-bin-DVD.iso

  MD5: 1539bad48e984ae1441052dc074c0995

- RHEL 5.1

  This OS is not available for free distribution.

  rhel-5.1-server-i386-dvd.iso

  MD5: 4380325824d389e2c202477c9d5b2855

Application Packages

VMware Studio uses the native packaging mechanism for the chosen operating system to install and update software in the virtual appliance. Any application installed on the virtual appliance must be packaged in the appropriate format supported by the virtual appliance operating system. RPM packages are used in RHEL, CentOS and SUSE. DEB packages are used in Ubuntu. VMware Studio also includes a command-line tool for creating packages, called `mkpkg`. See Appendix B, “Creating Application Packages,” on page 39 for more information.

Before creating your virtual appliance, make certain that you have all of the appropriate operating system and third-party packages that your application requires. You can install multiple packages, and you can control the installation order to resolve any package dependency issues.

Here are some considerations for assembling appliance packages and determining dependencies:

- Obtain any software packages that your application depends upon and are not available on the operating system distribution media. This includes system security updates and third-party packages. If your application depends upon a specific version of an OS package, you might have to obtain it. For example, your application might depend upon the latest version of Apache, but the OS distribution media that you are using might contain an older version. Place the latest package version in the application packages directory when configuring your virtual appliance so that the correct version is installed.
- You should be aware of package dependencies. VMware Studio installs appliance packages in the order specified in the Application Package Repositories list (see Table 3-4). You can specify either a directory or an individual package name. VMware Studio automatically resolves the installation order of packages in a directory, so it is easy for you to copy all packages into the default `ISV/appliancePackages` directory.
- For additional OS packages, ordering is not important because packages are obtained from the OS ISO.

The build profile contains a list of Application Package Repositories with packages to be downloaded over the Web and installed in the virtual appliance. This is done over the Web so you can include external Web servers. To include additional Application Package Repositories, specify either the URL of a directory containing one or more packages (VMware Studio resolves installation order), or the URL of a RPM or DEB package filename.

By default the Application Package Repositories list contains one URL that refers to a local directory on the VMware Studio appliance. The URL contains `[VADK.localIP]` to represent the IP address of the VMware Studio appliance. THis URL and local directory refer to the same location:

- URL of the application package repository – `http://[VADK.localIP]/ISV/appliancePackages`
- Local directory on VMware Studio appliance – `/opt/vmware/www/ISV/appliancePackages`

You can place your application packages in a different local directory on the VMware Studio appliance, or on an entirely different HTTP server. If you decide to use a non-default directory, be sure to specify its correct URL in the Application Package Repositories list or in your build profile. If you supply the URL of a directory or package on a different HTTP server, make sure it is accessible and remember to enable Directory Indexing in the HTTP server (for example, specify `Options +Index` in the Apache configuration file).

Importing Build Profiles

You can import build profiles into VMware Studio in the following circumstances:

- When you upgrade to a newer version of VMware Studio
- When you manually create a build profile
- When using a build profile created by another user on another VMware Studio appliance.

If you have an existing build profile that you want to import into VMware Studio, perform the following:

1 Place the build profile on the host where the Web browser runs.

2 Open the VMware Studio Web console to the **Appliance List** page and click **Import Appliance**.

3 A dialog appears where you must enter the name of the virtual appliance and a location for the build profile. Click the **Browse** button to locate the build profile on your computer.

4 Click **Import Appliance**. VMware Studio creates a new build profile based on the imported build profile. The imported build profile is added to the Appliance List. You can then select the virtual appliance and build or edit the virtual appliance.

Description Tab

Table 3-1 describes the configuration properties available on the **Description** tab.

Table 3-1. Description tab configuration properties

Field	Description
Appliance Information	
Appliance Name	Name for your virtual appliance. The appliance name is used in the virtual appliance Web console and the Welcome screen. If you enter a name that is 40 characters or longer, the UI resizes and alters the layout elements in the window.
Version	Version number for this release. The version number is used to track updates and must be in the format of major.minor.build.revision. Numbers can be in the range of 0-65535. For example, 1.0.0.0 for a new virtual appliance.
Full Version	This field allows you to enter further descriptive text for the version. For example, if the virtual appliance is in the alpha release state, you might enter “1.0.0.0 alpha”.
Vendor Name	Enter your company name.
Description	Descriptive text about the virtual appliance.
End User License Agreements	
EULA	The full text of your end user license agreement (EULA). The user is presented with this license when running the virtual appliance for the first time and must accept this license agreement. If no EULA is defined, then the virtual appliance boots without user intervention. You can add more than one EULA by clicking the Add EULA link on the top right of this field. If you wish to delete a EULA, click the Remove EULA link under the EULA you wish to remove.
Welcome Text	Enter information that provides a brief introduction to your virtual appliance. You may wish to supply instructions to the user on how to access your application on the virtual appliance. This text is displayed in the Welcome screen after the virtual appliance boots up. You can enter up to 16 lines of text with a maximum of 79 characters in each line. Characters over the maximum line length or lines after the 16th are not displayed.
Visual Customization	
Logo Path	Location of the image file on the VMware Studio appliance for the icon that displays in the Web console for this virtual appliance.
Logo Link URL	The URL to use when the logo is clicked in the Web console. You can specify your company’s Web site or a site that contains information about the virtual appliance. The link should be an http(s) URL. For example, http://www.example.com.
Application Link Label	The label for the URL in the Application Link URL field. This is the label that appears in the Web console for your application. For example, if your application is named “Acme”, you may wish to use the label “Acme Home” to point to the home address of the Acme application.
Application Link URL	URL for the site you wish to use for the Application Link Label field. This URL should be to the application on this virtual appliance. Replace host name with `${app.ip}` to resolve to the virtual appliance IP address at runtime. For example, use `http://${app.ip}/` to install a link to the application on the virtual appliance.
Boot Customization	
First Boot Script	A script that contains any application-related configuration that should occur when the virtual appliance is first booted up. For example, populating a database, configuring system data that the application needs, or starting various application components. This script is executed only on the first boot up of the virtual appliance.
Subsequent Boot Script	A script to execute on the second and subsequent boot ups of the virtual appliance. You may wish to enter commands for application startup in this field.
SSL Certificate Information	
Common Name	Server certificates are issued to a common name. In most cases, this is the full DNS name (fully qualified domain name) used to navigate to a website. This information is used to create a self-signed certificate for the virtual appliance Web console during the first boot.
Organization	The full legal name of your company. This name is used to sign the certificate.

Table 3-1. Description tab configuration properties (Continued)

Field	Description
Unit	The group or department within your company requesting the certificate.
Country Code	Enter the two-digit code for the country where your company is located. For example, US for the United States.

Hardware Tab

Table 3-2 describes the configuration properties available on the **Hardware** tab.

Table 3-2. Hardware tab configuration properties

Field	Description
Virtual CPU and Memory	
CPUs	Select the number of CPUs (1 or 2). The number must be less than or equal to the number of CPUs in the VMware Server used to create the virtual appliance.
Memory (MB)	The amount of memory (in megabytes) allocated to this virtual appliance. Enter a value in 4 MB increments (for example, 4, 16, 256, and so on).
Virtual Network	
Network Adapters	Number of virtual network cards to be configured for this virtual appliance. The VMware studio management agent provides console and Web console interfaces to manage only the first network adapter (eth0) if more than one adapter is defined.
Virtual Hard Disks	
Name	Name of the virtual disk. The disk name is used when naming VMDK files.
Size (GB)	Size in gigabytes of the virtual hard disk. If you require only one hard disk, enter the size of the disk and do not alter the default values in the Partitions section. By default, the first disk layout is locked. Adding or removing partitions is not allowed. Click the **Modify Disk Layout** link to alter the Partitions specifications. If you require more than one hard disk, click the **Add Virtual Disk** link. You can remove virtual hard disks by clicking the **Remove Disk** link.
Partitions	
Mount Point	Mount point for the partition.
Maximum Size	Check the box to specify that this partition automatically uses the maximum size available. Only the last partition of a disk should be marked as Maximum Size.
Size (MB)	Size in megabytes of the partition.
Filesystem	File system format, either `ext3` or `swap`.
Bootable	Check this box if you want to be able to boot from the partition.

OS Tab

Table 3-3 describes the configuration properties available on the operating system **OS** tab.

Table 3-3. Operating system tab configuration properties

Field	Description
Operating System	
OS Distribution	A read-only field specifying the operating system corresponding to the template you chose when you created this virtual appliance.
ISO URL	The location of the operating system distribution media ISO image. For better performance, you should download the ISO image to the machine where VMware Studio appliance runs prior to building your virtual appliance and have the URL refer to the local VMware Studio file system or make the ISO image available to VMware Studio using NFS.

Table 3-3. Operating system tab configuration properties (Continued)

Field	Description
ISO MD5	The MD5 checksum for the OS ISO image. You can compare the value in this field with the checksum for the ISO image to ensure that the image is correct.
User Accounts	
Add User	If you want to add a user account, click the **Add User** link. If you add a user, fill in all fields below. You can specify one or more users for your virtual appliance, and you may distribute these user names and passwords to end users.
Root Password	Enter the password of the administrative user account for the virtual appliance and then confirm the password by entering it again in the Confirm Password field.
Username	Enter the name for this user account.
Password	Enter the password for user name and then enter it again to confirm the password.
Full Name (optional)	Enter the full name for this user account.
UID (optional)	Enter User ID of this user account. Generally, not specifying the UID and allowing the OS to determine the UID is recommended. If specified, make sure to have a valid UID number value. Advanced use cases may require specifying the UID.
Installation Customization	
Pre-Install Script	Script that runs before the installation of application packages during the virtual appliance provisioning phase.
Post-Install Script	Script that runs after the installation of application packages during the virtual appliance provisioning phase.

Application Tab

Table 3-4 describes the configuration properties available on the **Application** tab.

Table 3-4. Application tab configuration properties

Field	Description
Application Package Repositories	A list of Application Package Repositories. Use the **Move Up** and **Move Down** buttons to specify the installation order. Select an item in this list to and click the **Remove** button to remove an application package repository from the list.
Application Package URL	To add an application package or directory of application packages to your virtual appliance, enter the URL of the package or directory in this field and click **Add**. See "Application Packages" on page 19 for details about application repositories.
List of packages from OS install media	A list of packages from the OS installation media that will be included in the virtual appliance. These packages are from the OS ISO image. Select a package and click the **Remove** button to remove the package from the list. You cannot remove any package from the list that is required by VMware Studio.
Additional Packages	To add a package to the list of packages, enter the package name in this field, then click **Add**. In addition to the ISO image, some OS vendors put additional packages on a supplemental disk. You can add those packages in this field. To add more than one package at once, separate each package name with a comma.

Output Tab

Table 3-5 describes the configuration properties available on the **Output** tab.

Table 3-5. Output tab configuration properties

Field	Description
Distribution Format	
File Name	The name for the distribution file of your virtual appliance. This name is visible to users when they download the virtual appliance ZIP or OVF package.
Output Format	Select the type of file you would like to create for your virtual appliance. You can choose more than one output format. ■ OVF – Use this format if your virtual appliance will be deployed on virtualization platforms that support the Open Virtual Machine Format, such as VMware Virtual Infrastructure Client. ■ ZIP – Use this format if your virtual appliance will be deployed on virtualization platforms that support the VMX configuration file, such as VMware hosted products (VMware Server, VMware Workstation, and VMware Player).
Archive Directory Name	The directory under which all files related to your virtual appliance should appear within the generated output file.
Update Repository	
Enable Repository	Click the check box to create an update repository. If you publish a virtual appliance without enabling the update repository, users of the appliance will be unable to update to newer versions automatically. See Chapter 4, "Publishing an Update For Deployed Virtual Appliances," on page 27 for more information about update repositories.
URL	Location of the update repository. You might plan to use a staging server for internal testing and copy the contents of your update repository manually to an Internet accessible and externally visible site when the virtual appliance is ready to be shipped. The update repository URL provided in the build profile is embedded in the virtual appliance. The virtual appliance uses that URL when checking for updates. If you plan to provide updates for your virtual appliance, it is critical that the update repository URL points to an Internet accessible location.
Username (optional)	User name for the update repository. This user name is used to authenticate to a Web server at the URL for the update repository. To comply with licensing restrictions, this field is required for updating any virtual appliance based on the Red Had Enterprise Linux (RHEL) operating system. All customers use the same user name to download updates.
Password (optional)	Password for the specified user name. Enter the same password twice to confirm. This password is used to authenticate to a Web server at the URL for the update repository. To comply with licensing restrictions, these fields are required for updating any virtual appliance based on the Red Had Enterprise Linux (RHEL) operating system. All customers use the same password to download updates.
Update Description	Description for this update. For example, a version number, issues fixed, or the enhancements provided by the update. Update information presented here displays in the Web console of the virtual appliance when the update is made available.
File Transfer Protocol	A read-only field indicating use of SCP file transfer protocol.
File Transfer Hostname	The host name or IP address where the update repository is located. VMware Studio must have SSH/SCP access to the host that serves as the update repository. This is needed to publish the virtual appliance file to the update repository.
File Transfer Username	User name on the update repository host. Used by VMware Studio to copy files using the SCP protocol to the update repository staging server. This user name can be different from the user name that the virtual appliances uses to download updates.
File Transfer Password	Password for user on the update repository host. Type it twice to confirm.
File Transfer Path	Path used for publishing virtual appliance updates to the update repository. For the virtual appliance to find any updates, this path should translate to the URL of the update repository.

Setup Tab

Table 3-6 describes the configuration properties available on the **Setup** tab.

File Transfer

Table 3-6. Setup tab configuration properties

Field	Description
VMware Server Settings	
Hostname	The IP address of the host where VMware Server is running.
Port	The port for the VMware Server VIX interface. The default port is 902 for VMware Server 1.0.4 and 1.0.5. For VMware Server 1.0.6, the default port is 904.
Username	The user name for the host used to authenticate to the VMware Server for remotely controlling the provisioning of a virtual appliance.
Password	Enter the password for the specified user name in both fields.
Host Path	The temporary path used by VMware Server to register and control provisioning of the virtual appliance. This location stores the provisioning virtual machine for building the virtual appliance. For example, if a Linux system is the host of the VMware Server and `/tmp` is the provisioning directory, then enter `/tmp` in the Host Path field. As an alternate example, if a Windows system is the host of the VMware Server and `c:\tmp` is the provisioning directory, SCP uses the OpenSSH server in Cygwin, so the Host Path is `c:\tmp` but the File Transfer Path below should be `/cygdrive/c/tmp`.
Host Environment	Specifies the host operating system where VMware Server runs. If you select Windows or Other, then the following File Transfer items must be defined. See the descriptions of these items under the **Output** tab. ■ File Transfer Protocol ■ File Transfer Username ■ File Transfer Password ■ File Transfer Path – Used by VMware Studio for uploading the provisioning virtual appliance to the Host Path. To upload the virtual appliance during the provisioning phase from VMware Studio to the host running VMware Server, you might need to specify the File Transfer Path and Host Path differently for the `scp` utility, although they must refer to the same directory on the host running VMware Server. For example, if VMware Server runs on Windows, SCP uses the OpenSSH server in Cygwin, so the Host Path is `c:\tmp` but the File Transfer Path should be `/cygdrive/c/tmp`.
Appliance Provisioning Settings	
VM Network Mode	Specify one of the following options: ■ Bridged mode is used by default to connect the provisioning virtual appliance to the LAN. Bridged mode must be used when the VMware Studio appliance and VMware Server are running on different computers and are connected by a LAN. ■ Host Only mode can be used when VMware Studio is running under a local installation of VMware Server. Note that Host Only provisioning requires that the VMware Studio virtual appliance be configured to use Host Only networking by powering off the VMware Studio virtual appliance and editing its virtual network adapter settings.
VM Network Type	Specify either DHCP to retrieving IP addresses from a DHCP server, or Static to set a fixed IP address. If you specify static IP addresses, the following fields appear: ■ IP Address ■ Netmask ■ Gateway ■ Preferred DNS Server ■ Alternate DNS Server

Verifying the Build

Once you have completed a build profile, you can build your virtual appliance. As the build progresses, a status icon appears to indicate that the build is in progress and a build log appears to inform you of each completed step in the build process. To cancel a build in progress, click **Cancel** under the status icon.

VMware Studio informs you whether the build was successful or if it failed. If you encounter problems while building a virtual appliance, refer to "Troubleshooting Builds" on page 53 for more information.

You can view the build log to verify a build once it is complete. Clicking on the status link in the Status column of the virtual appliance list brings up the build summary dialog. This dialog has the following tabs:

- **Build Summary** – provides details of the build profile configuration.
- **Verbose Log** – provides a detailed log of each completed phase of the build process.
- **Debug Log** – provides a detailed log of each completed build phase, including debugging information.
- **Install Log**—provides a log of the installation messages as each OS and application package is installed into the virtual appliance. Reviewing the install log can help when troubleshooting package dependency errors when additional packages are required.

Each build is stored in the file system. You may want to delete older builds when they are no longer needed. See "Old Builds" on page 32 for more information.

Testing the Finished Virtual Appliance

Once you have successfully built a virtual appliance, you can obtain the virtual appliance distribution file by clicking on the ZIP or OVF link (depending upon which kind of output file you specified) in the Download column of the virtual appliance list.

You should download your virtual appliance distribution file and test the completed virtual appliance by deploying it in the intended environment and ensuring that the virtual appliance and any applications that it contains work properly. Ensure that your application starts correctly. Ensure that all packages are deployed properly and boot scripts run correctly.

See the *Guide to Deploying Virtual Appliances* for information about deploying your virtual appliance for testing.

Publishing an Update For Deployed Virtual Appliances 4

This chapter describes how to publish updates to your virtual appliance and includes the following topics:

Overview of Updates

Once a virtual appliance is built and released, you might need to publish an update for the virtual appliance.

CAUTION To receive updates, virtual appliances must be configured with the update repository. In your build profile, make certain that you specify the update repository information before building the virtual appliance.

VMware Studio defines an update as a set of RPM or DEB packages. The update repository contains the RPM or DEB packages and a manifest file that describes the contents of a particular version of the virtual appliance. In the virtual appliance model, you have full control of what an update installs into your virtual appliance. Updates can include one or more of the following:

- Operating System Updates – You should download OS updates from the OS vendor's Web site.
- Application Updates – You provide your own application updates.

To update a virtual appliance

1. Monitor updates from the OS vendor and from VMware.
2. Analyze the updates and select applicable updates for your virtual appliance.
3. Test these updates to ensure that your application continues to work as before in the virtual appliance.
4. Package your updates as RPM or DEB. See Appendix B, "Creating Application Packages," on page 39.
5. Update your existing build profile with a new version number and the updated application packages.
6. Build a new version of your virtual appliance with the updated build profile. The build process places the latest virtual appliance update into the update repository.
7. Copy the update files from the staging repository server to the production repository server.
8. Publish the update. See "Publishing an Update" on page 28.

Generating an Update

To generate an update from the VMware Studio Web console

1. Log into the VMware Studio Web console.
2. In the **Appliance List**, select the virtual appliance for which you want to generate an update.
3. Under Actions, click **Edit Appliance**.
4. Change the version number in the profile for the virtual appliance. This is the key indicator that an update is being generated for an existing virtual appliance.
5. On the Output page, try not to change the update repository settings for URL and File Transfer Hostname.

   NOTE If you change the URL in a new version, you must inform end users, in the update description, that they need to reboot the virtual appliance so that the VMware Studio in-guest agent can process the new URL. Place the update, which contains the new embedded URL, on the old update server as well as on the new update server. The new URL will be used for future updates.

6. If you copied the staging server update to an update repository at an external site, ensure that you point to that same staging server when generating an update. Then manually copy the entire contents of the staging server to the update repository at that external site.

VMware Studio creates a new manifest with the new version number specified in the build profile. Building a new version of a virtual appliance updates the repository for the appliance. VMware Studio marks the new version of the virtual appliance as the latest version and copies the changed and new packages to the update repository. If you remove a package, the newly generated manifest reflects the exact list of packages required for this new version of the virtual appliance.

NOTE Changing the virtual appliance name or the vendor name in the build profile while generating updates might affect integration with VMware Update Manager. Refer to "Staging an Update for Testing and Publishing Update Externally" on page 29 for more information.

Updates Requiring Reboot

The application might require a reboot after applying the update. If the update requires a reboot of the whole virtual appliance, you should provide that information in the update repository description field. This field is displayed to the end user before the update is downloaded on the deployed virtual appliance.

VMware Studio automatically detects if kernel packages are being updated. End users are notified during an update if that is the case. Users are asked to reboot their appliance after the update completes.

Publishing an Update

After the fields are modified in the build profile, click **Save and Build** to build your virtual appliance with the updates. As part of the build process, the update repository is updated with a new manifest. You can see that `manifest_latest.xml` is updated to reflect the update, and additional packages are copied to the repository.

Test the generated virtual appliance to ensure it works as expected with the updates.

Once you are ready to make the updates visible to end users, you can copy the entire contents to your external Web server that hosts the update repository. Because deployed virtual appliances at customer sites have the URL of this server embedded, the following actions can occur:

- Virtual appliances flag an update-available, if automatic check for update is enabled.
- The end user can manually check for updates to view any available updates.
- The end user can see the Update Description that describes this update, and react to critical information that it contains. For example, you could include an advisory message like this in the Update Description: "You must reboot your virtual appliance after applying this update."

Staging an Update for Testing and Publishing Update Externally

The repository information configured through the Web console persists in the RepositoryStagingServer element of the build profile. Additionally, VMware Studio can create a ZIP archive of each update when you enable Update Repository Export. Update Repository Export can be used to manually control publishing of updates by manually copying the archive and extracting the update files to a remote Web server.

To enable Update Repository Export, the build profile must be edited using a text editor. Build profiles can be found in the directory `/opt/vmware/var/lib/build/profiles/` on the VMware Studio virtual machine. You must add a new XML element called `vadk:RepositoryExport` into the `vadk:UpdateSection_Type` section of the build profile. The `vadk:RepositoryExport` element has two attributes:

- **vadk:format** – This must have the value `zip`.
- **vadk:filename** – The filename of the ZIP archive containing the software update files.

The following is a sample `vadk:RepositoryExport` XML addition that you can add to a build profile:

```
<vadk:RepositoryExport vadk:format="zip" vadk:filename="update"/>
```

The RepositoryStagingServer element should be correctly configured to a staging server that you can use for testing the update. During the build, VMware Studio creates the following update ZIP file:

```
/opt/vmware/www/build/<buildNumber>/exports/zip/<exportFile>.zip
```

To publish an update from the ZIP archive created using Repository Export, extract the contents of the file in the document root of the URL that was specified as the update URL. For example, if you set the update URL to `http://<your-server>/updates/` and set the document root in the Web server to `/var/www/htdocs/`, then you would extract the contents of the ZIP file into `/var/www/htdocs/updates/` on `<your-server>`. Virtual appliance users would then be able to retrieve and install the update from your Web server.

To publish an update from the ZIP archive

1 Connect with SSH to your production update server, in this example `support.example.com`:

```
ssh support.example.com
```

2 Run `scp` to secure copy from the VMware Studio virtual appliance `<vmwStudio>` into `/tmp`:

```
scp root@<vmwStudio>:/opt/vmware/www/build/<buildNumber>/exports/zip/<exportFile>.zip /tmp
```

If the production update server has HTTP access to your virtual appliance, you could instead run:

```
cd /tmp
wget --no-proxy http://<vmwStudio>/build/<buildNumber>/exports/zip/<exportFile>.zip
```

3 Create a directory for updates on your Web server:

```
mkdir -p /var/www/htdocs/updates/productXYZ
cd /var/www/htdocs/updates/productXYZ
```

The path specified must be in the Web server document root (in this example, `/var/www/htdocs`) and must correspond to the repository URL specified for the appliance during the build.

If this directory already contains files from a previous update archive, the previous update manifest XML file, `manifest-latest.xml`, should be renamed as a backup file. The name of this backup file is arbitrary. For example, you could embed its creation date in the format `yyyymmdd` for year month and date. To make your backup of the old update manifest file, run a command like the following. If you ever need to revert your update server to an older update version, you can overwrite the `manifest-latest.xml` file with the contents of a previous manifest backup file.

```
mv ./manifest/manifest-latest.xml ./manifest/manifest-<yyyymmdd>.xml
```

4 Run the following commands to extract the update archive contents into the `productXYZ` directory:

```
unzip /tmp/<exportFile>.zip
rm /tmp/<exportFile>.zip
```

These examples are for Linux-based Web servers. Instructions would be similar for OVF instead of ZIP export. For Windows repository staging servers, the SSH service must be installed and active.

Update Repository Format

The update repository has a specific format. Each time you publish an update to the update repository, new packages, and new versions of existing packages, are added to the repository. The `manifest-latest.xml` file is replaced by the new build's manifest, the old manifest is moved to `manifest-<version>.xml`, and `manifestrepo.xml` is updated with any new packages in the repository.

Examine the `manifest-latest.xml` file to verify that the update is published to the repository. The update repository file structure is the following:

```
<Update Repository URL>
    package-pool
        package1-version1.{rpm|deb}
        package1-version2.{rpm|deb}
        package2-version1.{rpm|deb}
        ...
    manifest
        manifest-latest.xml
        manifest-repo.xml
        manifest-<version1>.xml
        manifest-<version2>.xml
        ...
```

Supporting Central Update Management with VMware Update Manager

The updates that are published for an appliance can be applied by the end user using either the Web console of the virtual appliance or using the centralized update management provided by VMware Update Manager.

VMware Update Manager enables the end user to create a baseline to control updates to virtual appliances, create a baseline for a particular appliance or for appliances from a given vendor, and so forth. The baselines specified using the vendor name and product names are internally referenced in Update Manager by the VendorUUID and the ProductRID. Update Manager retrieves the VendorUUID and ProductRID from the virtual appliances. These identifiers can be defined by the build profile in VMware Studio for the appliance.

By default, the VendorUUID and the ProductRID fields are empty in build profiles and the VMware Update Manager uses the vendor name and the name of the appliance instead of the identifiers while referencing the appliances in the baselines. If the vendor name or the appliance name changes for future versions of the virtual appliance, then any baselines created by the end user would become invalid. To avoid invalid baselines, use VendorUUID and ProductRID.

- **VendorUUID** – The globally unique identifier for your company or organization. The UUID can be a Windows GUID (Globally Unique Identifier).
- **ProductRID** – The unique identifier for your product, which should be unique across all appliances from the vendor.

Perform the following steps to enter VendorUUID and ProductRID keys into the build profile:

1. After specifying your build profile, select Save and Close to save your build profile.
2. Open your build profile with an XML editor. Your build profile is located in the following directory:

   `/opt/vmware/var/lib/build/profiles/<profileName>.xml`
3. Locate the `ExtendedProductSection_Type` in your build profile.
4. In the `VendorUUID` element, enter the UUID for your virtual appliance.
5. In the ProductRID element, enter the RID for your update repository.
6. **Save and close** your build profile.
7. Open your build profile in VMware Studio Web console and build the virtual appliance.

Creating and Managing Builds From the Command Line 5

You can use the VMware Studio Web console to create and build virtual appliances. You can also use the VMware Studio command line interface to integrate with an existing application build system, automate the build process, and integrate with your source control system. For information about build profile files, see Appendix C, "Manually Editing Build Profiles," on page 43. This chapter describes the VMware Studio command line interface and includes the following topics:

- "Using the Command Line" on page 31
- "Managing Builds" on page 32

Using the Command Line

VMware Studio provides a command-line interface to build, query, and clean up virtual appliances. To use the command-line interface, log in to the VMware Studio console by accessing the VMware Studio welcome screen and selecting the Login option. Alternatively, you can use a terminal emulator to make an SSH connection to the VMware Studio appliance. After you log in as `root`, run the following command:

```
studiocli
```

Table 5-1 describes the options available for the `studiocli` command.

Table 5-1. Options to the studiocli Command

Option	Alternate	Description
`-p`	`--profile`	Specifies the qualified filename of a build profile. Generic template build profiles can be found under `/opt/vmware/etc/build/templates`. VMware Studio generated profiles are located at `/opt/vmware/var/lib/build/profiles`.
`-c`	`--createbuild`	Creates the virtual appliance based on a given build profile. Also generates a repository for updating the virtual appliance if a repository is specified in the build profile. For example, to build a virtual appliance, enter the following command: `studiocli --createbuild --verbose --profile /<path>/<yourbuildprofile.xml>`
`-i`	`--instance`	A unique tag for each build. The value must be less than 16 characters long and can be composed of the letters a-z, A-Z, the numbers 0-9, and the characters - and _. By default, a build instance will be generated by VMware Studio.
`-D`	`--debug`	Enables debug mode. Normally, after a build, the transient virtual machine is deleted from the provisioning engine. For debugging purposes, this option prevents the transient virtual machine from being deleted. This option also controls whether temporary files used in the build process are deleted after a build. If this option is set, no temporary files are deleted. This option also enables verbose debugging messages (-vv option described below).
`-t`	`--validateprofile`	Validates a given build profile by testing resource authentication, availability, and connectivity. For example, to validate your build profile before building your virtual appliance, run the following command: `studiocli --validateprofile --profile /<path>/<yourbuildprofile.xml>`

Table 5-1. Options to the studiocli Command (Continued)

Option	Alternate	Description
`-v`	`--verbose`	Increases descriptive output with the first invocation. Enables debug output with subsequent invocations (-vv).

`-b`	`--buildstatus`	Displays the status and log messages for a specified build instance or all instances per build profile. For example, to list all build instances:
		`studiocli --buildstatus`
		To list the status of build instance myVA with extra verbosity, run:
		`studiocli --buildstatus --verbose --instance myVA`
		To list the status of all build instances for build profile myBuildProfile.xml, run:
		`studiocli --buildstatus --verbose --profile myBuildProfile.xml`

`-h`	`--help`	List the command line options of studiocli.
	`--cancelbuild`	Cancels the build currently in progress. If this virtual appliance is currently being provisioned on the VMware Server, it is deleted from VMware server.

`-d`	`--deletebuild`	Deletes a specified build instance or all instances for a specified build profile including its virtual appliance, log and update files. For example, to delete all files associated with the build instance myVA, run:
		`studiocli --deletebuild --verbose --instance myVA`
		To delete all files associated with the build profile myBuildProfile.xml, run:
		`studiocli --deletebuild --verbose --profile myBuildProfile`

`-f`	`--flushisocache`	Deletes the ISO image extraction file cache `/opt/vmware/cache/build/vol/*`. The following command clears the ISO cache after building an virtual appliance. Use it to reclaim disk space. Note that the next build using the associated ISO will be slower because the cache will be rebuilt. `studiocli --flushisocache`
	`--mkpkg`	Helps in the creation of application packages. Using this tool you can create either DEB or RPM application packages. For more information, see Appendix B, "Creating Application Packages," on page 39

Managing Builds

For every build of a virtual appliance, VMware Studio maintains a build instance number and the logs for that build. For example, build instance 1 stores logs in the following directory:

```
/opt/vmware/www/build/1/status/
```

This directory contains three important log files for each build:

- **`verbose.log`** – High level information of the build process.
- **`postinstall.log`** – Information about the build of the virtual appliance including package installation.
- **`debug.log`** – Detailed information about the build including ISO management, network communications, and provisioning information.

Build logs are accessible using the Studio Web console to troubleshoot build failures.

Old Builds

VMware Studio stores the virtual appliances it builds, and logs for all these builds. If you no longer need the older builds, you should regularly delete them or copy them to another host, so as to recover free disk space for VMware Studio.

You can obtain a list of all builds by running this command:

```
studiocli --buildstatus
```

You can obtain information about a specific build by specifying its profile and instance number:

```
studiocli --buildstatus --profile myVirtualAppliance --instance 7
```

Old builds and cached ISO image extractions can rapidly consume disk space. To check the free disk space available to VMware Studio, run the following command:

```
df
```

To delete just one instance of a build, run the following command:

```
studiocli --deletebuild --instance <number> --verbose
```

To delete all builds created from a common build profile, run the following command:

```
studiocli --deletebuild --profile <build-profile-name> --verbose
```

Even if you specify an instance number with the above command, it deletes all instances.

The ISO image extraction cache can be found at:

```
/opt/vmware/cache/build/vol
```

To flush the ISO image extraction cache, run the following command:

```
studiocli --flushisocache
```

The ISO image download cache can be found at:

```
/opt/vmware/cache/build/ISO
```

To flush the ISO image download cache, run the following command:

```
rm -v /opt/vmware/cache/build/ISO/*
```

Managing the VMware Studio Appliance 6

This chapter outlines procedures to change network settings, shut down, and reboot the VMware Studio virtual appliance, including the following topics:

- "Shutting Down or Rebooting the VMware Studio Appliance" on page 35
- "Network Configuration" on page 35

Shutting Down or Rebooting the VMware Studio Appliance

Click the **System** tab in the virtual appliance Web console.

On the **System Information** tab you see the virtual appliance vendor, appliance name, and appliance version. The computer host name, operating system, and operating system version are also shown.

Shutdown and **Reboot** buttons are under the Actions heading on the right side of the window.

- To shut down the virtual appliance, click the **Shutdown** button.
- To restart the virtual appliance, click the **Reboot** button.

Network Configuration

The **Network** tab contains sub-tabs for the following operations:

- **Status** – Displays network status information, such as the interface name, whether DHCP was used, the IP address, netmask, gateway IP address, and DNS servers. You can click the **Refresh** button under the Actions heading (on the right side) to obtain the most current information.
- **Address** – Allows you to specify static IP information or to retrieve the IP settings from a DHCP server. Click **Save Settings** to accept any changes that you make to the IP settings. Click **Cancel Changes** to discard any changes. If you set a static IP address you must enter values in all the following fields:
  - **IP Address** – IP address of virtual appliance.
  - **Netmask** – Network mask for the virtual appliance.
  - **Gateway** – IP address of the gateway (network router).
  - **Preferred DNS Server** – IP address of the primary DNS server.
  - **Alternate DNS Server** – IP address of the secondary DNS server.
- **Proxy** – Allows you to specify a proxy server and port for accessing external networks (for example, the wide-area Internet). Click **Save Settings** to accept any changes that you make to the proxy settings. Click **Cancel Changes** to discard any changes. If you check the checkbox next to **Use a Proxy Server**, you must enter values in the following fields:
  - **Proxy Server** – Host name or IP address for the proxy server.
  - **Proxy Port** – Proxy server communications port.

Installing SSH on Windows A

VMware Studio requires secure shell (SSH) to communicate with VMware Server. Because Windows systems do not ordinarily include support for the SSH protocol, when VMware Server is hosted on a Windows system, you must install it. To install the SSH daemon on a Windows system, perform the following procedure:

1 Download the **OpenSSH 4.2p1-1 for Windows** Binary Installer from this Web site:

   http://sshwindows.webheat.co.uk/

2 Run the installer, `setupssh.exe`.

3 After installation, create a working directory for provisioning virtual appliances, for example:

   ```
   mkdir C:\vmware_studio-tmp
   ```

   Note that `c:\vmware_studio-tmp` translates to `/cygdrive/c/vmware_studio-tmp` when you set the VMware Server File Transfer Path.

4 Select a username and password to use in the script below.

5 The following script uses the `net user` command to create a user named `vmware_studiouser` with password `VMware`. Later it uses the `cacls` command to control ACLs for the `C:\vmware_studio-tmp` directory you just created. Either create a batch script containing the following commands, or run these commands sequentially in a command prompt window:

   ```
   net user vmware_studiouser VMware /add
   C:
   cd \"Documents and Settings"
   mkdir vmware_studiouser\.ssh
   cacls vmware_studiouser /T /P everyone:F
   cd \
   cacls vmware_studio-tmp /G everyone:F

   cd \Program Files\OpenSSH\bin
   mkgroup -l > ..\etc\group
   mkpasswd -l > ..\etc\passwd
   ```

6 Launch the OpenSSH daemon by running the `net start` command:

   ```
   net start opensshd
   ```

7 Disable any firewall, or preferably, create an exception in the Windows system firewall to avoid blocking the SSH connection with VMware Studio (both TCP and UDP protocols on port 22).

Creating Application Packages

B

This appendix contains the following sections:

- "VMware Studio Make Package (mkpkg) Tool" on page 39.
- "Operating System Native Tools" on page 42.

VMware Studio Make Package (mkpkg) Tool

Vendors can make Linux applications available in several formats: in the native packaging format of the operating system (DEB or RPM), or in some compressed archive format (ZIP, `tgz`, and so forth). To deliver an application as a virtual appliance, VMware Studio requires that the application be packaged in the native packaging format of the operating system. This requirement guarantees that the software vendor can update the application in a consistent fashion after an end user deploys the virtual appliance. Automatic updates, version tracking, dependency resolution, rollback, and complete virtual appliance manifests are a few of the benefits of requiring that applications be delivered in the native operating system packaging format.

Options of the

VMware Studio provides a command line tool, `mkpkg`, to help in the creation of application packages. Using the tool you can create either DEB or RPM application packages. The following list shows the operating systems supported by VMware Studio and the package format for the operating system:

- CentOS – RPM
- Redhat – RPM
- SLES – RPM
- Ubuntu – DEB. For DEB packages, creation of the pre-install and post-install scripts is specified and used at the time of building the package.

For RPM packages, creation of the pre-install and post-install scripts is done at the time of creating the `control.spec` file. The scripts are inserted into the generated template `control.spec` so they need to be specified when you run `mkpkg –t rpm –T`.

The `mkpkg` command syntax, with options described in Table B-1, is as follows:

```
mkpkg –t {rpm|deb} [–T] [–c ctrl] [–P preinst] [–p postinst] [–R prerem] [–r postrem] dir
```

This is the same as running `mkpkg` from the `studiocli` front-end:

```
studiocli ––mkpkg –t {rpm|deb} [–T] [–c ctrl] [–P preinst] [–p postinst] [–R prerem] [–r postrem] dir
```

Table B-1. mkpkg Options

Option	Description
–t	Type of package to create (`rpm` or `deb`). Mandatory argument.
–T	Creates a sample control file to be used as a template for the package type.

Table B-1. mkpkg Options

-c	Package control file. You can create a sample control file to use as a template with the **-T** option.
-P	Pre-installation script to be run before installing the package.
-p	Post-installation script to be run after installation of the package.
-R	Pre-removal script to be run before removing the package.
-r	Post-removal script to be run after removing the package.
dir	The directory to be packaged. Programs and files in this directory should be in their proper place when installed, relative to this directory. For example, a file that should be installed into the `/etc` directory should be put into a subdirectory called `etc` in this directory.

Creating a DEB Package

Before you create a DEB package, you must organize your application packages into the proper directory structure. For example, executable files in `/bin` must be under a subdirectory named `bin` and files that belong in `/etc` should be under a subdirectory named `etc`.

Hadoop, an open source distributed file system written in Java, is our example here. To create a DEB package for Hadoop, perform the following procedure:

1 Extract Hadoop to `/<your_home>/hadoop-0.18.08` directory.

2 On the VMware Studio appliance, change directory to `/<your_home>.`

```
cd /<your_home>
```

3 Create a Control file by running this command:

```
mkpkg -t deb -T
```

This produces the following message:

```
deb control file template has been created as “control”
```

4 You can now modify the control file as needed.

```
Package: hadoop-dfs
Version: 1.00
Essential: yes
Priority: extra
Section: utils
Maintainer: yourname@example.com
Architecture: i386
Description:  Hadoop is a distributed file system based on map-reduce.
```

NOTE The control file also contains a Depends section. If you do not have any package dependencies, make sure you remove the Depends section.

5 You can create pre-install, post-install, pre-removal, and post-removal scripts for your application. These scripts can contain commands that you would run while installing your application. In this example we name our pre-install and post-install scripts `preinst` and `postinst`. Similar naming could be followed for the pre-removal and post-removal scripts. Run the following command:

```
mkpkg -t deb -c control -P preinst.sh -p postinst.sh hadoop-0.10.0
```

If your command syntax is correct, you get a message similar to the following:

```
109737 blocks
dpkg-deb: building package ‘hadoop-dfs’ in /<your_home>/hadoop-dfs-1.00_i386.deb’
```

6 Copy the generated DEB file to `/opt/vmware/www/ISV/appliancePackages` with the `cp` command:

```
cp hadoop-dfs_1.00_i386.deb /opt/vmware/www/ISV/appliancePackages
```

Creating an RPM Package

Before you create an RPM package, you must organize your application packages into the proper directory structure. For example, executable files in `/bin` must be under a subdirectory named `bin` and files that belong in `/etc` should be under a subdirectory named `etc`.

SugarCRM, an open-source software solution for customer relationship management, is our example here. To create an RPM package for SugarCRM, perform the following procedure:

1 Extract SugarCRM to `/<your_home>/SugarCE-Full-5.0.0g` by running the following command:

```
cd /<your_home>
```

2 Create a Control file by running:

```
mkpkg -t rpm -T
```

This produces the following message:

```
rpm control file template has been created as “control.spec”
```

Unlike for DEB, `mkpkg` for RPM places the pre-install and post-install scripts within the control file.

3 You can now modify the `control.spec` file as needed.

```
Summary: Sample SugarCRM package
Name: SugarCRM
Version: 5.0.0g
Release: 11
License: Commercial
Vendor: SugarCRM
Group: System Environment/Daemons
URL: http://www.sugarcrm.com
BuildArchitectures: noarch
#Requires:
%description
```

The following is an example of a SugarCRM package:

```
%post
#!/bin/bash
chown -R wwwrun /opt/SugarCE-Full-5.0.0g/

#
# Do not put anything below the %defattr line; the list of files
# in this package is automatically written there.
#
%files
%config /opt/SugarCE-Full-5.0.0g/config.php
%defattr(-,root,root)
```

The `%config` section lists configuration files that you do not want overwritten during an update.

4 Run `mkpkg` with the control.spec file you just edited.

```
mkpkg -t rpm -c control.spec SugarCE-Full-5.0.0g
```

If your command syntax and dependencies are correct, you will see messages similar to the following before your RPM is generated:

```
109737 blocks
Processing files: SugarCE-Full-5.0.0g
Checking for unpackaged file(s): /usr/lib/rpm/check-files
Wrote: /<your_home>/ sugarcrm_5.0.0g.0-11_i386.rpm
```

5 Copy the generated RPM file to `/opt/vmware/www/ISV/appliancePackages` with the `cp` command:

```
cp sugarcrm_5.0.0g.0-11_i386.rpm /opt/vmware/www/ISV/appliancePackages
```

Operating System Native Tools

If you choose not to use mkpkg, the following sections provide a brief overview of how to package an application in DEB or RPM formats.

DEB

DEB packages are constructed by writing a small control file and then pointing the packaging tool at a directory hierarchy that contains only the application files installed in their proper order.

A control file can be very simple, for example:

```
Package: change-this-package-name
Version: 99.99
Essential: no
Priority: extra
Section: utils
Maintainer: yourname@example.com
Architecture: i386
Description: This is a description of the package, and should contain summary informaton about
               what the package provides.
```

The command to generate the package is: dpkg-deb. Because of file permission issues, you usually call it from a command called fakeroot, as in this example:

```
fakeroot dpkg-deb -b <directory-hierarchy>
```

RPM

RPM packages are constructed similarly to DEB-based packages, but in addition to pointing the packaging tool at a directory containing the application's files, each file in the application must be referenced in the package control file, as in this example:

```
Summary: This is the short description of the package.
Name: change-this-package-name
Version: 99.99
Release: 99.99
License: Commercial
Vendor: Your Company Name
Group: System Environment/Daemons
URL: http://YourCompanyURL
BuildArchitectures: noarch
%description
This is the longer description of the package,
and should contain more detailed informaton about
what the package provides.

%files
%defattr(-,root,root)
/bin/ls
```

The command to generate RPM packages is rpmbuild, which you run as in this example:

```
rpmbuild -bb <controlfile> --buildroot <directory-hierarchy>
```

Manually Editing Build Profiles

C

This chapter can help you edit build profiles. It contains the following sections:

- "Structure of XML Build Profiles" on page 43
- "Creating Virtual Disks, Partitions, and Virtual Devices" on page 50

Structure of XML Build Profiles

VMware Studio creates a build profile that defines a virtual appliance. Currently these build files conform to the OVF 0.9 draft specification. For more about OVF, see http://www.vmware.com/appliances/learn/ovf.html. VMware recommends that you use the VMware Studio Web console to create build profiles. Once created, you can edit the build profile and build or rebuild the virtual appliance. The VMware Studio Web console saves build profiles in the following location:

```
/opt/vmware/var/lib/build/profiles/<profileName>.xml
```

VMware Studio includes pre-configured build profile template, which are located in the following directories:

```
/opt/vmware/etc/build/templates/<OS>/<release>/<version>/build_profile.xml
```

The sections of the build profile correspond to tabs in the VMware Studio build interface. Table C-1 describes the configuration properties in the build profile XML file.

Table C-1. Build Profile XML Configuration Properties

Property	Description
Application Information	
Appliance Name	This name is visible to users when they download the virtual appliance ZIP or OVF file. `//ovf:Envelope/Section[@xsi:type='ovf:ProductSection_Type']/Product`
Version	VMware Studio uses the version of the virtual appliance to track updates. It must be in X.X.X.X format, where X must be a number between 0 and 65535. `//ovf:Envelope/Section[@xsi:type='ovf:ProductSection_Type']/Version`
Full Version	Free text string used to display the version information of the virtual appliance. `//ovf:Envelope/Section[@xsi:type='ovf:ProductSection_Type']/FullVersion`
Vendor Name	Name of the virtual appliance vendor. `//ovf:Envelope/Section[@xsi:type='ovf:ProductSection_Type']/Vendor`
Description	Description of the functionality the virtual appliance provides. It is displayed when the virtual appliance is imported into ESX as an OVF. `//ovf:Envelope/Section[@xsi:type='ovf:AnnotationSection_Type']/Annotation`

Table C-1. Build Profile XML Configuration Properties (Continued)

Property	Description

EULA Text	End user licensing agreement for the virtual appliance. The user must accept this agreement before using the virtual appliance.
	`//ovf:Envelope/Section[@xsi:type='ovf:EulaSection_Type']/`
	`License`
	You can add more than EULA in the build profile by copying the <Section> element containing the EulaSection_Type and placing the additional EULA text in the second <Section> element. For example:
	`<Section xsi:type="ovf:EulaSection_Type" xml:lang="en">`
	`   <Info>End User License Agreement</Info>`
	`   <License>add first EULA text here</License>`
	`</Section>`
	`<Section xsi:type="ovf:EulaSection_Type" xml:lang="en">`
	`   <Info>End User License Agreement</Info>`
	`   <License>add second EULA text here</License>`
	`</Section>`

Vendor UUID	Universally Unique Identifier for the ISV. For more information, see "Supporting Central Update Management with VMware Update Manager" on page 30.
	`//ovf:Envelope/Section[@xsi:type='vadk:`
	`ExtendedProductSection_Type']/vadk:VendorUUID`

Product RID	Repository Identifier for the virtual appliance. For more information, see "Supporting Central Update Management with VMware Update Manager" on page 30.
	`//ovf:Envelope/Section[@xsi:type='vadk:`
	`ExtendedProductSection_Type']/vadk:ProductRID`

Welcome Text	Welcome text for the virtual appliance shown in the Welcome screen. The welcome text must be 16 lines or less, with a maximum of 79 characters in each line. Any lines or characters greater than these limits will not display in the virtual appliance console.
	`//ovf:Envelope/Section[@xsi:type='vadk:`
	`ExtendedProductSection_Type']/vadk:WelcomeText`

Visual Customization	

Logo Path	Icon displayed in the Web console for the virtual appliance. Specify the location of an image file that you have copied into a directory in VMware Studio.
	`//ovf:Envelope/Section[@xsi:type='vadk:`
	`ExtendedProductSection_Type']/vadk:Logo`

Logo Link URL	URL to your company's Web site.
	`//ovf:Envelope/Section[@xsi:type='vadk:`
	`ExtendedProductSection_Type']/vadk:LogoUrl`

Application Link Label	Label provided in this field is shown in the Web console for this virtual appliance. This label points to the URL provided in Application Link URL field.
	`//ovf:Envelope/Section[@xsi:type='vadk:`
	`ExtendedProductSection_Type']/vadk:AppUrlName`

Application Link URL	URL on the Web console to the application provided in this virtual appliance. Replace hostname with "`${app.ip}`" to resolve address at runtime.
	`//ovf:Envelope/Section[@xsi:type='ovf:ProductSection_Type']/`
	`AppUrl`

Boot Customization	

First Boot Script	Shell script that runs when the virtual appliance boots up for the first time.
	`//ovf:Envelope/Section[@xsi:type='vadk:ApplicationSection_Type']/`
	`vadk:BootActions/vadk:FirstBoot`

Subsequent Boot Script	Shell script that runs when the virtual appliance boots up each time except for the first time.
	`//ovf:Envelope/Section[@xsi:type='vadk:ApplicationSection_Type']/`
	`vadk:BootActions/vadk:SubsequentBoot`

Table C-1. Build Profile XML Configuration Properties (Continued)

Property	Description
SSL Certificate Information	
Common Name	Server certificates are issued to a common name. In most cases, this is the full DNS name used when navigating a website. `//ovf:Envelope/Section[@xsi:type='vadk:SSLCertificateSection_Type']/vadk:CommonName`
Organization	Legal name of your organization. It is used to sign the certificate. `//ovf:Envelope/Section[@xsi:type='vadk:SSLCertificateSection_Type']/vadk:OrganizationName`
Unit	Department or section of your organization. This information is used in the creation of the certificate. `//ovf:Envelope/Section[@xsi:type='vadk:SSLCertificateSection_Type']/vadk:OrganizationalUnitName`
Country Code	Two letter country code representing the country in which the organization or unit is based. This information is used in the creation of the certificate. `//ovf:Envelope/Section[@xsi:type='vadk:SSLCertificateSection_Type']/vadk:CountryName`
Hardware	
	WARNING It is highly recommended that you use the VMware Studio Web console to edit this section. Invalid configuration will result in build failure. See "Creating Virtual Disks, Partitions, and Virtual Devices" on page 50
Virtual CPU and Memory	
	Item `<rasd:InstanceId>` in the following XML should be sequential beginning with number 1 and unique for every new resource.

CPUs	The number of virtual processors to be configured for this virtual appliance.
	WARNING This number must be less than or equal to the number of CPUs in the VMware Server used to create the virtual appliance. If the machine, on which VMware Server runs, is incapable of booting a virtual machine with more than one CPU, then only one CPU can be specified in the build profile.
	`//ovf:Envelope/Section[@xsi:type='ovf:VirtualHardwareSection_Type']`
	`Item <rasd:ResourceType> = 3 defines the CPUs`
	`Item <rasd:VirtualQuantity> defines the number of CPUs`

Memory (MB)	Amount of memory (RAM) in megabytes (MB) allocated to this virtual appliance in 4MB increments.
	`//ovf:Envelope/Section[@xsi:type='ovf:VirtualHardwareSection_Type']`
	`Item <rasd:ResourceType> = 4 defines the memory (RAM)`
	`Item <rasd:VirtualQuantity> defines the amount of memory`

Virtual Network	

Network Adapters	The number of network cards to be configured for this virtual appliance.
	`//ovf:Envelope/Section[@xsi:type='ovf:VirtualHardwareSection_Type']`
	`Item <rasd:ResourceType> = 10 defines a network adapter`

Virtual Hard Disks	
	There can be a maximum of 4 partitions on each non-root virtual disk. However, in most cases, one partition using up the size of the disk should be sufficient, since there is a maximum of 60 disks supported for CentOS 5.0 and RHEL 5.1. For Ubuntu 7.04, Ubuntu 7.10, and Ubuntu 8.04, 15 disks are supported. It is recommended that you use the VMware Studio Web console to add disks and partitions. This section is complicated and has many interdependencies within the build profile XML. For more information, see "Creating Virtual Disks, Partitions, and Virtual Devices" on page 50.

Table C-1. Build Profile XML Configuration Properties (Continued)

Property	Description
OS	
OS Distribution	Indicates the name of the OS for your virtual appliance. `//ovf:Envelope/Section[@xsi:type='vadk:JEOSSection_Type']/ vadk:Distribution`
ISO URL	URL from where the ISO is downloaded or accessed. Both file:// and http:// protocols are supported, but if the http:// protocol is used, it may take a very long time for the first build to download the ISO image. For that reason, the default build profile templates specify a file:// location on VMware Studio in which to put a pre-downloaded ISO image. Please ensure that the ISO URL is valid and reachable from VMware Studio. `//ovf:Envelope/Section[@xsi:type='vadk:JEOSSection_Type']/ vadk:ISO/@vadk:path`
ISO MD5	Any ISO downloaded from the ISO URL must have the same MD5 checksum shown here. `//ovf:Envelope/Section[@xsi:type='vadk:JEOSSection_Type']/ vadk:ISO/@vadk:md5sum`
Root Password	Password for the administrative user account for the virtual appliance. `//ovf:Envelope/Section[@xsi:type='vadk:ApplicationSection_Type']/ vadk:Users/vadk:User[@vadk:username='root']/@vadk:password`
Pre-Install Script	Shell script run before the installation of application packages during the virtual appliance provisioning phase. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/ vadk:PreInstallShellScript`
Post-Install Script	Shell script run after the installation of application packages during the virtual appliance provisioning phase. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/ vadk:PostInstallShellScript`
Application	

Application Package Repositories	List of URLs to repositories where the application packages are available. You can copy application packages into VMware Studio at:
	`/opt/vmware/www/ISV/appliancePackages`
	`http://[VADK.localIP]/ISV/appliancePackages`
	`URL points to this directory.`
	`//ovf:Envelope/Section[@xsi:type='vadk:ApplicationSection_Type']/ vadk:ApplicationPackages`

OS Package List	List of packages from the OS install media to be included in the virtual appliance. No packages can be removed from the required list of packages. `//ovf:Envelope/Section[@xsi:type='vadk:ApplicationSection_Type']/ vadk:OSPackages`
Output	
File Name	Name of the virtual appliance file. This name is visible to users when they download the virtual appliance ZIP or OVF package. `//ovf:Envelope/Section[@xsi:type='vadk:ApplicationSection_Type']/ vadk:VMXFilename`

Output Format	Use OVF format if your virtual appliance is going to be deployed on any VMware ESX Platform. ZIP format will work on any one of VMware's hosted products.
	`//ovf:Envelope/Section[@xsi:type='vadk:ApplicationSection_Type']/ vadk:DistributionFormat/`
	`@vadk:format`

Table C-1. Build Profile XML Configuration Properties (Continued)

Property	Description

Archive Directory Name	Directory under which all files related to virtual appliance should appear within the generated archive.
	`//ovf:Envelope/Section[@xsi:type='vadk:ApplicationSection_Type']/ vadk:DistributionFormat/`
	`@vadk:directory`

Update Repository	
Enable Repository	If enabled, an update repository is created for this virtual appliance. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type'] must be present.`
URL	URL to the update repository. This URL is embedded in the generated virtual appliance and is used by the virtual appliance to check for and install available updates. If the URL is changed in a subsequent version, you should inform end users, in the update description, to reboot the virtual appliance so that Small Footprint CIM Broker (SFCB) processes the new URL. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:Repository/@vadk:url`
Username (optional)	Username used to authenticate to the Web server at the URL for the update repository. To comply with licensing restrictions, this field is required for updating any virtual appliance based on the Red Hat Enterprise Linux (RHEL) operating system. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:Repository/@vadk:username`
Password (optional)	Password to be used to authenticate to the Web server at the URL for the update repository. In order to comply with licensing restrictions, this field is required for updating any virtual appliance based on the Red Hat Enterprise Linux (RHEL) operating system. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:Repository/@vadk:password`
Update Description	Update information presented here is displayed in the Web console of the virtual appliance when the update becomes available. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:Info`
File Transfer Protocol	Protocol used to publish software updates to the Web server which serves as an update repository. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:RepositoryStagingServer/@vadk:fileProto`
File Transfer Hostname	Hostname of the host running the Web server which serves as an update repository. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:RepositoryStagingServer/@vadk:hostname`
File Transfer Username	This is the username used by VMware Studio to copy files using scp protocol to the repository staging server. It can be different from what the virtual appliances use to download updates via http. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:RepositoryStagingServer/@vadk:fileLogin`
File Transfer Password	This is the password used by VMware Studio to copy files using scp protocol to the repository staging server. It can be different from what the virtual appliances use to download updates via http. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:RepositoryStagingServer/@vadk:filePassword`
File Transfer Path	Path used for publishing virtual appliance updates to the update repository. If the virtual appliance has to find any updates, this path should translate to the update repository URL. `//ovf:Envelope/Section[@xsi:type='vadk:UpdateSection_Type']/ vadk:RepositoryStagingServer/@vadk:filePath`

Table C-1. Build Profile XML Configuration Properties (Continued)

Property	Description
Appliance Provisioning Settings	
Hostname	Specify the host where VMware Server is running. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:hostname`
Port	Port on which VMware Server is running. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:port`
Username	Username on the host used to authenticate to the VMware Server for remotely controlling the provisioning virtual appliance using the VIX protocol. //ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:login
Password	Password for Username used to authenticate to the VMware Server for remotely controlling the provisioning virtual appliance using the VIX protocol. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:password`
Upload Path	A local directory path on the physical box where the VMware Server runs. This location is used to store the provisioning virtual machine for building the virtual appliance. The Upload Path and the Registration Path must refer to same physical directory. For example, a Linux box is the host of the VMware Server and /tmp is the provisioning directory. In this case, both Upload Path and Registration Path would have /tmp as their entries. A Windows box is the host of the VMware Server and c:\tmp is the provisioning directory. In this case Upload Path would have c:\tmp as its entry while Registration Path would have /cygdrive/c/tmp as its entry. This presumes OpenSSH for Windows is installed for the SSH service. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:path`
File Transfer	
File Transfer Protocol	Protocol used to communicate between the VMware Server host and the VMware Studio appliance. Ensure that communication using the specified protocol works between the VMware Studio appliance and the host where VMware Server runs. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:fileProto`
File Transfer Username	Username used to transfer provisioning virtual appliance files to and from the VMware Server host. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:fileLogin`
File Transfer Password	Password used to authenticate the file transfers to and from the VMware Server host. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:filePassword`
File Transfer Path	A path used by VMware Studio for uploading the provisioning virtual machine to the Host Path. The Registration Path and the Upload Path must refer to same physical directory. For example, a Linux box is the host of the VMware Server and /tmp is the provisioning directory. In this case, both Upload Path and Registration Path would have /tmp as their entries. A Windows box is the host of the VMware Server and c:\tmp is the provisioning directory. In this case Upload Path would have c:\tmp as its entry while Registration Path would have /cygdrive/c/tmp as its entry. This presumes OpenSSH for Windows is installed for the SSH service. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningEngine/vadk:VmhostLan/@vadk:filePath`

Table C-1. Build Profile XML Configuration Properties (Continued)

Property	Description
VM Network Mode	Specify one of the following options: ■ Bridged ■ Host only Virtual NIC mode used during virtual appliance software provisioning. The network mode is always reset to bridged prior to ZIP and OVF packaging. `//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningSettings/vadk:NetworkMode`

VM Network Type	VMware Studio provides two options for obtaining IP addresses for the virtual appliance being built during provisioning, using DHCP or by specifying a static IP address. Specify one of the following:
	■ `DHCP`
	■ Static
	`//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:P vadkProvisioningSettings/vadk:NetworkType`
	IP Address – the static IP address to be used for the virtual appliance during provisioning.
	//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningSttings/vadk:IP
	Netmask – The netmask to be used for the virtual appliance during provisioning. A netmask enables the recipient of IP packets to distinguish the network ID and host ID portions of the IP address. Netmasks normally are of the format 255.x.x.x.
	//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningSettings/vadk:Netmask
	//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/
	Gateway – The IP Address of the designated gateway for this network.
	//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningSettings/vadk:Gateway
	Preferred DNS Server (Optional) – The primary DNS server to be used for the virtual appliance during provisioning. A name server is responsible for mapping IP addresses to names.
	//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningSettings/vadk:DNS1
	Alternate DNS Server (Optional) – The alternate DNS server to be used for the virtual appliance during provisioning. A name server is responsible for mapping IP addresses to names.
	//ovf:Envelope/Section[@xsi:type='vadk:BuildSection_Type']/vadk:ProvisioningSettings/vadk:DNS2

Creating Virtual Disks, Partitions, and Virtual Devices

This section provides information about the virtual hardware that is defined in the build profile for a virtual appliance. In the VMware Studio GUI, the Hardware tab is where this information is defined.

Hardware

This section defines the virtual hard drives and SCSI controllers that appear in the virtual appliance.

An example controller specification looks like this:

```
/ovf:Envelope/Section[@xsi:type='ovf:VirtualHardwareSection_Type']
...
    <Item>
        <rasd:BusNumber>0</rasd:BusNumber>
        <rasd:Caption>SCSI Controller 0 – LSI Logic</rasd:Caption>
        <rasd:InstanceId>4</rasd:InstanceId>
        <rasd:ResourceSubType>lsilogic</rasd:ResourceSubType>
        <rasd:ResourceType>6</rasd:ResourceType>
    </Item>
```

- **rasd:BusNumber** – The controller number. Controller 0, hard drive 0 is defined to be the boot device. Four controllers are supported (0-3).
- **rasd:Caption** – A text description of the controller.
- **rasd:InstanceId** – The order of this Item in the hardware section.
- **rasd:ResourceSubType** – The vendor of the controller. The keyword lsilogic is the only supported type.
- **rasd:ResourceType** – The type of controller. ResourceType 6 is the only supported type, indicating a SCSI controller.

An example of a hard drive looks like this:

```
<Item>
    <rasd:AddressOnParent>1</rasd:AddressOnParent>
    <rasd:Caption>Harddisk 1</rasd:Caption>
    <rasd:HostResource>/disk/newdisk</rasd:HostResource>
    <rasd:InstanceId>5</rasd:InstanceId>
    <rasd:Parent>4</rasd:Parent>
    <rasd:ResourceType>17</rasd:ResourceType>
</Item>
```

- **rasd:AddressOnParent** – The hard drive address on the controller. Fifteen hard drives per controller are supported, numbered 0-15, with address 7 being reserved.
- **rasd:Caption** – A text description of the hard drive.
- **rasd:HostResource** – A string that associated this hard drive with an associated entry in the DiskSection. This entry always begins with /disk/ followed by the id from the entry in the DiskSection.
- **rasd:ResourceSubType** – The vendor of the controller. The keyword lsilogic is the only supported type.
- **rasd:InstanceId** – The order of this Item in the hardware section.
- **rasd:Parent** – The controller on which the hard drive is to reside. This number is the InstanceId of the controller, not the controller number.
- **rasd:ResourceType** – The type of hardware device. 17 is a hard drive.

Disk

This section defines the name, size, and format of the virtual hard drives that appear in the virtual appliance.

The following is an example of a disk specification:

```
/ovf:Envelope/Section[@xsi:type='ovf:DiskSection_Type']
...
    <Disk ovf:diskId="newdisk" ovf:capacity="2048"
        ovf:capacityAllocationUnits="MegaBytes"
        ovf:fileRef="newdisk.vmdk"
        ovf:format="http://www.vmware.com/specifications/vmdk.html#sparse"
    />
```

- **ovf:diskId** – The id of this disk. The id matches this disk with a hard drive defined in the Hardware section.
- **ovf:capacity** – The size of the disk, in ovf:capacityAllocationUnits
- **ovf:capacityAllocationUnits** – The size units. MegaBytes are the supported value.
- **ovf:fileRef** – The name of the generated file for this disk.
- **ovf:format** – The format of the vmdk file. This value should always be: `http://www.vmware.com/specifications/vmdk.html#sparse`

Partition

This section defines the partitions on the disks specified in the Disk section.

The following is an example of a partition specification:

```
/ovf:Envelope/Section[@xsi:type='vadk:DiskPartitionSection_Type']
...
    <vadk:Partition vadk:diskId="newdisk" vadk:bootable="0" vadk:fs="ext3" vadk:mount="/newdisk"
                   vadk:size="grow"/>
```

- **vadk:diskId** – The ovf:diskId in the Disk section to which this partition refers.
- **vadk:bootable** – Whether this partition is bootable or not. This flag is only set in the root partition (controller 0, disk 0) and only on the boot partition.
- **vadk:fs** – The type of partition to make. A standard file system will have the value ext3; the other valid value is swap to define a swap partition to extend the virtual memory of the virtual appliance.
- **vadk:mount** – The place where the disk appears on the virtual appliance's file system. This field is ignored for partitions marked swap.
- **vadk:size** – The size of the partition. Normally, each new disk will have only one partition, specifying the entire disk. This is specified with the word grow in this attribute. Otherwise, a value may be entered here that specifies the size of the partition in megabytes. A grow partition should be the last partition specified in the disk.

Virtual USB Controller

To add a virtual USB controller to the virtual appliance, in the build profile section, `<Section xsi:type="ovf:VirtualHardwareSection_Type">`, add the following:

```
<Item>
    <rasd:Caption>USB controller</rasd:Caption>
    <rasd:InstanceId>8</rasd:InstanceId>
    <rasd:ResourceType>23</rasd:ResourceType>
</Item>
```

Virtual Sound Card

To add a virtual sound card to the virtual appliance, in the build profile section, `<Section xsi:type="ovf:VirtualHardwareSection_Type">` add the following:

```
<Item>
    <rasd:Caption>es1371 sound controller</rasd:Caption>
    <rasd:InstanceId>7</rasd:InstanceId>
    <rasd:ResourceSubType>es1371</rasd:ResourceSubType>
    <rasd:ResourceType>35</rasd:ResourceType>
</Item>
```

Troubleshooting D

Table D-1 lists problems that you might encounter when using VMware Studio, and provides possible resolutions for these problems.

Table D-1. Troubleshooting information for VMware Studio

Problem	Resolution
Troubleshooting Builds	
During the build of your virtual appliance, the build does not progress.	This can happen because you specified an OS package in the build profile that does not exist as part of the OS distribution media in the expanded ISO located in the `/opt/vmware/cache/build/vol` directory. This can be verified while monitoring the virtual appliance provisioning on the VMware Server console. To prevent this situation, ensure that the OS package specified in the build profile exists in the expanded ISO directory for the selected OS.
During build, you get error: The virtual appliance guest IP Address <IP Address> could not be contacted.	You might have problems getting an IP Address from your DHCP server. The alternative is to use a static IP address for the provisioning process. Obtain an IP address from your network, and update the virtual machine Network Type on the Setup tab in the build profile.
If the same VMware Studio virtual machine is used to build the same virtual appliance on different operating systems, the VMware Studio virtual machine runs out of space.	Use the `studiocli --flushISOCache` command to empty the ISO cache and restart the build. This command clears out ISO images for operating systems that you no longer need. See "Managing Builds" on page 32 for more information.
During operating system installation in a virtual appliance build, the /boot partition size causes the build to fail.	Check the `/var/log/syslog` and see if there are disk full errors. If so, you can increase the size of your `/boot` partition in the Hardware tab of the build profile.
Unable to build Ubuntu virtual appliances that contain more than 15 virtual disks.	The Ubuntu installer has a limitation of creating a maximum of 15 disks during the OS installation process. Given the restriction, VMware Studio is also bound to create no more than 15 virtual disks for any Ubuntu virtual appliances. Manually add the desired virtual disks to the virtual appliance after it is built by VMware Studio.
With http_proxy set, build of virtual appliance fails with the following error: "Bad url: http://localhost/ISV/appliance"	Change the [VADK.LocalIP] in the application package URL to the IP address of the VMware Studio appliance.
The virtual appliance build process fails due to a SSH/SCP failure, but a manual SCP test from VMware Studio to the host of VMware Server (the provisioning engine) does not have a problem.	The problem could be a slow network with the SCP process timing out after hitting the 5 second limit. Try running on a faster network.

Table D-1. Troubleshooting information for VMware Studio (Continued)

Problem	Resolution

Application packages specified in the build profile are not included in the completed virtual appliance.	This can occur if the proper file system permissions are not set on the application packages. VMware Studio must be able to read the application package files. You may see errors in the build log, such as the following:
	`http://10.20.96.22/ISV/appliancePackages/firefox-3`
	`.0-2.el5.centos.i386.rpm:`
	`16:49:43 ERROR 403: Forbidden.`
	`http://10.20.96.22/ISV/appliancePackages/xulrunner`
	`-1.9-1.el5.i386.rpm:`
	`16:49:43 ERROR 403: Forbidden.`
	Ensure that users can read application package files by setting the proper permissions on the file.

The build stops while failing to power on the transient virtual machine on the VMware Server. Review the last entries of the build debug log for information on VIX power on.	This can be caused by invalid characters in the build profile. Make certain any disk names specified in the profile do not contain invalid characters like < or >.
VMware Studio does not restrict build instance deletion while the build process is running.	Users may call "`studiocli --deletebuild`" to delete an active build instance, which would then leave a stopped build process in memory. Run "`studiocli --cancelbuild`" or cancel the build process from the Web interface before deleting an active build instance. If the command "`studiocli --deletebuild`" was already executed by mistake, simply terminate the stopped vabs.pl process in the memory.

Table D-1. Troubleshooting information for VMware Studio (Continued)

Problem	Resolution
The provisioning phase stops while waiting for DHCP configuration or the build fails after a long time with an error message stating that the virtual appliance was unable to get an IP address.	There are a couple of possible causes. One cause of this problem is if the virtual appliance has 4 ethernet adapters, 60 disks and 4 scsi controllers—VMware Studio does not support this configuration. This problem is also seen when running VMware Studio and VMware Server on a host that is on a public network. During the virtual appliance build process, VMware Studio creates a transient virtual machine that installs the operating system and applications. This transient virtual machine is controlled by VMware Studio through a network connection. The default network configuration of the VMware Studio appliance bridges the VMware Studio appliance to the physical network on the host machine. There are special cases where a bridged network may not work well for VMware Studio during virtual appliance creation. A typical network with a DHCP server is sufficient; however some DHCP servers are incompatible with Etherboot, which is used in the virtual appliance build process. If your network is using an incompatible DHCP server, virtual appliance builds will fail. To identify this situation, view the VMware server console during the virtual appliance build. If the appliance virtual machine fails to get an IP address during the boot process, then either there is no DHCP server available, or the DHCP server is not compatible with Etherboot. The way to work around this issue is to use the virtual DHCP server provided in the VMnet1(host-only) network rather than depending on the external DHCP server. Perform the following procedure to correct this problem: 1 In the VMware Server Console that builds the virtual appliance, select **Host>Virtual Network Settings...** 2 In the Virtual Network Editor dialog box, select the **NAT tab**. 3 The default option for VMnet host is VMnet8. Change this to **VMnet1** and click **OK**. Once this change is done, this step does not need to be repeated for future builds. In the VMware Server Console, edit the settings for the virtual machine. 1 Ensure that the VMware Studio appliance is shut down. 2 In the Virtual Machine Settings dialog box, open the **Hardware tab**. 3 Under the device list, click Ethernet and change the Network Connection to **Host-only**. 4 Click **OK**. 5 Start the VMware Studio appliance. Notice that the IP address might be different from any previous value. Once this change is done, this step does not need to be repeated for future builds. Before building a virtual appliance, the build profile should be set to use the correct networks settings. 1 In VMware Studio console edit the build profile. 2 Click the **Setup** tab. 3 In the VMware Server section, set the VM Network Mode to **Host only**. 4 Set the property **Hostname** to the virtual IP address given to the host OS for VMnet1. 5 On Windows, this is found on a command line by running `ipconfig` and finding the IP Address for VMware Network Adapter VMnet1. On LinuxR this is found on a command line by running `ifconfig vmnet1` and finding the IPv4 address in the inet addr property. This step must be repeated for every virtual appliance profile.

Table D-1. Troubleshooting information for VMware Studio (Continued)

Problem	Resolution
	6 If you use the VMware Studio appliance as the repository staging server for updates, this workaround will limit the mobility of a generated virtual appliance if updates are used. Using a separate server as the repository staging server allows updates to work correctly without modification. VMware Studio generates virtual appliances with a default network mode of bridged. Because this work-around changes the VMware Studio appliance network interface to be host-only, the generated virtual appliance will not be able to connect to the VMware Studio appliance. To use the update process while also using this work-around, manually change the generated virtual appliance to host]only before checking for updates as described above. Also, ensure that in the virtual appliance profile, the Update Repository URL is pointing to the IP address of the VMware Studio appliance after switching to host-only. This only works if the generated virtual appliance is on the same host.
When connecting to the VMware Server, an error displays: `Failure: An error occurred when connecting to the VMware Server at 10.20.30.123: Insufficient permissions in host operating system`	Ensure that you have the correct password for VMware Server.
VMware Server running on Windows reports: “`VMware Server unrecoverable error: (vmx)`” during the virtual appliance build process. If a user, without the local administrative privileges to access a VMware Server on Windows, runs the build process it may cause this error.	Select a user with Windows administrative privileges for the VMware Server section in the build profile.
VMware Server gives this unrecoverable error: “Cannot get temporary directory for log file.” A core file is available. Please request support.	Ensure you have given Administrative privileges to the user on the host running VMware Server. This is the same user that you specified in the Setup Tab in VMware Server Section during build profile creation, for authenticating to the VMware Server for remotely controlling the provisioning virtual appliance using the VIX protocol.

When provisioning a virtual appliance with Redhat 5.1, wget segment fault occurs. This fault happens sporadically. You can identify this fault in the error log by the following:
`Segmentation fault wget $WGET_TIMEOUT –nv –nd ––no–parent –r –A .deb,.rpm,.sh –l1 –P $installdir`
`VADK ABORT: Downloading of VADK runtime packages from http://10.20.29.216/build/Rhel532534/repo/vadkPackages/ failed.`
	Run the virtual appliance build again as the fault is sporadic.

During the network provisioning of your virtual appliance, you receive the following error message in the log: `Failed to get DHCP or failed to communicate with port 80 in the VADK host.`	Try increasing the ProvisioningTimouts vadk:pxe value from the default value of “600” to “800” in the build profile for your virtual appliance: `<vadk:ProvisioningTimeouts vadk:pxe="800" vadk:poweroff="25" vadk:hang="7200"/>` If increasing the timeout does not resolve this issue then use Static IP provisioning instead of DHCP. Contact your network administrator to obtain valid IP network settings for your environment.

Table D-1. Troubleshooting information for VMware Studio (Continued)

Problem	Resolution
Build finished with a successful status but the file systems for the disks are dirty or corrupt.	Try increasing the ProvisioningTimouts vadk:powerOff timeout value from the default value of "25" to "60" in the build profile for your virtual appliance. Some operating systems require longer to finish installation after the post installation script has been executed. Since we do not have control over the provisioning virtual machine after that point and before actual shutdown this timeout needs to be increased to give the OS the time it needs to finish up. `<vadk:ProvisioningTimeouts vadk:pxe="600" vadk:poweroff="25" vadk:hang="7200"/>`
A virtual appliance, running the SLES operating system, might not get discovered as virtual appliance in VUM 1.0u2. The virtual appliance might still show as a virtual machine after the virtual appliance is fully powered on and finishes the first-time power on configuration.	Power off the virtual appliance running the SLES operating system. Remove it from the VMware vCenter Server inventory. Then browse the host datastore and locate the virtual appliance, and add it back into the vCenter Server inventory. Power on the virtual appliance again, wait for the virtual appliance to finish booting, virtual appliance discovery will be run again and this time it should recognize the virtual appliance correctly.
Disk partitioning for your virtual appliance has partitions assigned to the wrong mounts.	Ensure that if you assign a partition with size=grow' that it is the last partition on the disk.
Update package build fails, but a ZIP or OVF file is created.	This problem can be caused by not changing the version number for the update package. The package has new content, but the version number is the same as a previous package. Ensure that when you create an update package that you change the version number. The problem can also occur if the application packages are of the same version, but different checksum
Using a static IP address for the provisioning virtual machine, VMware Studio reports that the virtual appliance could not be contacted. Watching the provisioning virtual machine boot on the VMware Server console, the boot process attempts to download pxelinux.0, but does not succeed.	A recently identified issue in the etherboot code causes the network mask to be read in reverse order. When specifying the netmask parameter for the virtual appliance in VMware Studio, enter the netmask in the reverse order. For example, if the netmask is supposed to be 255.255.252.0, enter it as 0.252.255.255.
Troubleshooting Problems Running VMware Studio	
VMware Studio appliance does not get an IP Address assigned.	Go to VMware Studio console, and use the "Configure Network" option on VMware Studio welcome screen.
During the first boot of a virtual appliance, the virtual appliance stops running before reaching the login prompt. Restarting the virtual appliance results in the First Boot Script being skipped.	Discard the virtual appliance and start over with the virtual appliance OVF or ZIP file.