AppCapture 3.4 Administration

VMware Horizon Cloud Service

Horizon Cloud with Hosted Infrastructure 16.11

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

https://docs.vmware.com/

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

Managing Applications for Deployment with AppCapture

1

You use AppCapture to create AppStacks for provisioning applications to user groups.

Before you can assign applications to users, you must capture and package applications into AppStacks by using the AppCapture utility. You then manually copy the AppStacks to a file share.

This section includes the following topics:

AppCapture System Requirements

Review these AppCapture minimum requirements for Windows platforms.

AppCapture System Requirements

To install and run AppCapture, you must verify that your system meets the following minimum requirements.

- OS. AppCapture works on all Windows platforms from Windows 7 onward, including Windows 8.1 and Windows 10, for both x86 (32-bit) and 64-bit machines: physical, Workstation, or ESX VMs.
- OS: AppCapture works on Windows 7 and Windows 10 platforms, for both x86 (32-bit) and 64-bit machines: physical, Workstation, or ESX VMs.
- OS: AppCapture works on Windows 7 and Windows 10 platforms, for both x86 (32-bit) and 64-bit machines: physical, Workstation, or ESX VMs.
- Disk space: The amount of disk space required depends on the number and size of the applications that you are provisioning. Verify that your system has enough disk space for all the AppStacks that you are creating.

Install AppCapture

Use the AppCapture utility to package applications to copy to a file share.

Prerequisites

Ensure that you do not have the App Volumes agent installed on the virtual machine where you plan to install AppCapture. If you have the App Volumes agent installed on the machine, take a snapshot of the machine, clone it, and uninstall the App Volumes agent.

Procedure

1. Log in as administrator to the machine where you want to install AppCapture.
2. Download the AppCapture installer, `VMware–appvolumes–appcapture–<buildnumber>.exe` from the VMware downloads page.
3. Double-click the installer and follow the on-screen instructions to install AppCapture.
4. (Optional) Verify that `AppCapture.exe` is installed in `C:\Program Files\VMware\AppCapture` (64-bit machines) or `C:\Program Files\VMware\AppCapture` (32-bit machines).

What to do next

The UEM Application profiler is also installed with the AppCapture utility. You can personalize AppStacks using the UEM Application profiler.

Using AppCapture

Before you can assign applications to users, you must package the applications into AppStacks. An AppStack is a collection of files, folders, registries, and metadata stored in `.vhd` or `.vmdk` files. The AppStack is accompanied by a `.json` file.

You use AppCapture to create and manage AppStacks. AppCapture is a standalone utility which you run outside of App Volumes. You can run AppCapture either from a command line or using Microsoft PowerShell.

You create AppStacks on a virtual machine with the AppCapture utility.

App Volumes uses only .vmdk files. You might use .vhd files to install applications on a physical machine with other VMware products.

AppCapture and UEM Application Profiler

You might want to personalize an AppStack after capturing the applications in it, without performing an actual assignment.

You can use the UEM application profiler that is packaged with the AppCapture installer for personalization. When you use the `AppCapture.exe` command with the `/personalize` option, the UEM application profiler window is displayed. You can choose the applications you want to personalize and store the settings.

See AppCapture Command-Line Options for details about using the `/personalize` option.

Using AppCapture from a Command Line

You can create an AppStack by running AppCapture from a command line or with Microsoft PowerShell.

To run AppCapture with Microsoft PowerShell, see Using AppCapture with Microsoft PowerShell.

Run AppCapture from the Command Line

You can run AppCapture from a command line.

Note You must capture applications from the same operating system into which you mount them. For example, if users are operating a Win7x64 operating system, you must capture the applications by using a similar or an identical base operating system Win7x64 image.

Prerequisites

1 You must run AppCapture as administrator.

2 Verify that User Account Control (UAC) in Windows is disabled. To turn off UAC, see http://windows.microsoft.com/en-us/windows/turn-user-account-control-on-off#1TC=windows-7.

3 Verify that the CLI command `AppCapture.exe` is installed in `C:\Program Files (x86)\VMware\AppCapture` (64-bit machines) or `C:\Program Files\VMware\AppCapture` (32-bit machines).

4 To view options of the `AppCapture.exe` command, see AppCapture Command-Line Options.

Procedure

1 Take a snapshot of the system.

You can revert to the snapshot after the capture session.

2 Open a console window.

3 Run the `AppCapture.exe` command: **`AppCapture.exe /n`** *your_appstack_name*.

Do not press Enter at this point. Run all installers first, and return to this window.

The AppStack virtual machine disk is ready in generally less than a minute.

4 Close the console window and run normal Windows installation for each of the applications that you want to capture.

   a Accept the default installation of all applications on the C: drive. The installation activity redirects to the virtual output disk.

   b If an installer requires a reboot, wait for the reboot to finish.

   c If this feature is available, you can also run ThinApp MSI packages. You can install these packages in the same way that you install other application MSI packages. See the latest ThinApp documentation for information about how to create ThinApp MSI packages.

5 Finish virtual disk creation.

   a After all installers that are required to be captured in this AppStack have run, return to the console window.

   b Press **Enter** to initiate a reboot and finish virtual disk creation.

   After the reboot you see new AppStacks containing applications.

   c Verify that you have new VHD and VMDK files in `C:\ProgramData\VMware\AppCapture\appvhds`.

6 Run the `AppCapture.exe` command to view applications in the VHD file and VMDK files. For VHD files: **`AppCapture.exe /list`** *my_AppStack_Name.vhd* and for VMDK files: **`AppCapture.exe /list`** *my_AppStack_Name.vmdk*

7 Copy the AppStacks that you have created to a staging file share of your choice.

8 Revert to the system snapshot that you captured before you started the first capture session.

9 Copy the AppStacks from the staging file share to your system.

AppCapture Command-Line Options

Use the AppCapture command-line options to create and manage AppStacks.

`AppCapture.exe` Command Options

The `/meta`, `/vhd`, and `/vmdk` options are useful if you accidentally delete a JSON, VHD, or VMDK file. If a JSON file is deleted, App Volumes cannot read the AppStack.

You can personalize an AppStack using the `/personalize` command.

The `AppCapture.exe` command accepts the following options:

Table 1-1. `AppCapture.exe` Command-line Options

Task	Option
Display help for the `AppCapture.exe` command.	`/?`
Specify an author's name for the AppStack. If the name contains at least one space, put the name in parentheses. Example: **`AppCapture.exe /n /a (IT Admin)`**	`/a`

Table 1-1. `AppCapture.exe` Command-line Options (Continued)

Task	Option
Specify a description for an AppStack. Example: **`This disk contains XYZ suite of applications.`**	`/d`
List the contents of the AppStack JSON, VHD, and VMDK files. If you are not using the default directory, specify the directory where the files are located. Example: **`AppCapture.exe /list`** ***`filePath`***	`/list`
Generate a `.json` file by using a VMDK file as input. If you are not using the default path, specify the path containing the VMDK file. Example: **`AppCapture.exe /meta`** ***`appStackPath`***.	`/meta`
Create an AppStack. Example: **`AppCapture.exe /n`**	`/n`
Specify an output directory for the AppStack files. The default directory is `C:\ProgramData\VMware\AppCapture\appvhds`. You can use this option with the `/s` option to create an AppStack from an existing AppStack. See Update an AppStack from the Command Line. Example: **`AppCapture.exe /s`** ***`oldAppStackDir`*** **`/o`** ***`newAppStackDir`***	`/o`
Specify a source directory for the AppStack files. The default directory is `C:\ProgramData\VMware\AppCapture\appvhds`. Do not use this option if you are installing a new application. You can use this option with the `/o` option to create an AppStack from an existing AppStack. See Update an AppStack from the Command Line. Example: **`AppCapture.exe /s`** ***`oldAppStackDir`*** **`/o`** ***`newAppStackDir`*** You can also use the `/s` option with `/n` to update an old AppStack with a new one. In this example, the existing *oldAppStack.vhd* AppStack is copied as a base AppStack and can be updated as *newAppstackName*: **`AppCapture.exe /n`** ***`newAppstackName`*** **`/s`** ***`oldAppStack.vhd`*** **`/o`** ***`newAppStackDir`***	`/s`
Create a `.vhd` file from a `.vmdk` file. If you are not using the default path, specify the path containing the `.vhd` file. Example: **`AppCapture.exe /vhd`** ***`appStackPath`*** **`.vmdk`**	`/vhd`
Generate a VMDK file by using a VHD file as input. If you are not using the default path, specify the path containing the `.vhd` file. Example: **`AppCapture.exe /vmdk`** ***`appStackPath`*** **`.vhd`**.	`/vmdk`

Table 1-1. `AppCapture.exe` Command-line Options (Continued)

Task	Option
Virtualize the application after provisioning it for pre-verification. When using the `/test` option with no other parameters, the AppStack should contain only one application bundle. Example: **AppCapture.exe /test *Provisioned appStackPath*.vhd** Virtualize all application bundles in the AppStack. Example: **AppCapture.exe /test *Provisioned appStackPath*.vhd** * Virtualize application bundles that are identified by their corresponding GUIDs in the AppStack. Example: **AppCapture.exe /test *Provisioned appStackPath*.vhd *GUID1*, *GUID2*.. *GUIDn***	`/test <Provisioned AppStackPath>.vhd [* | GUID]`

Enable the user to personalize the application bundle using the UEM application profiler. Configuration files that contain the personalization settings are generated. By default, the files are saved in the same location as the VHD, under the `UEMConfigFiles\`*`AppStack`* folder.
Example:
AppCapture.exe /personalize *C:\FinanceApps.vhd* - Personalization settings are saved under `C:\ProgramData\VMware\AppCapture\appvhds\UEMConfigFiles\FinanceApps`.
The `/predef` sub-option is an optional boolean switch that can be used with the `/personalize` option to capture the predefined settings of the specified application bundle into a configuration file. The predefined settings are captured in an additional configuration file.
Example:
AppCapture.exe /personalize *C:\FinanceApps.vhd* /predef - Personalization settings along with predefined settings are saved under `C:\ProgramData\VMware\AppCapture\appvhds\UEMConfigFiles\FinanceApps`
The `/flexconfigname` sub-option can be used with the `/personalize` command to store the personalization settings into a user-friendly configuration file name.
Example:
AppCapture.exe /personalize *C:\FinanceApps.vhd* /flexconfigname *MSOffice2016* - Personalization settings are saved under `C:\ProgramData\VMware\AppCapture\appvhds\UEMConfigFiles\MSOffice2016`
	`/personalize <ProvisionedAppStackPath>.vhd`
	`[/predef | flexconfigname <flexconfigfilename> ]`

Merging AppStacks

You can merge two or more AppStacks from the command line by using AppMerge.

Use AppMerge to merge two or more existing AppStacks into one file. AppMerge takes as its input VHD files associated with an AppStack.

Note The input AppStack files must all be of type VHD. You can create a merged output AppStack of a different type with the `/vhd` and `/vmdk` options.

AppMerge has this syntax:

```
AppMerge.exe /o outputAppStack /s "inputAppStack1file","inputAppStack2file",
"inputAppStack3file",...
```

Example: Creating a Merged AppStack

In this example, you create an AppStack file called `MergedAppstack.vhd` from three existing AppStack files, `Office.vhd`, `Notepad++.vhd`, and `Firefox.vhd`:

```
AppMerge.exe /o C:\MergedAppstack.vhd /s "Office.vhd","Notepad++.vhd","Firefox.vhd"
```

You can specify input file paths, output file paths, and file names. In this case, the three input AppStacks are presumed to be in the default AppStack location. The output AppStack goes in the C: drive.

Besides the `/o` and `/s` parameters, AppMerge accepts the following options:

- `/df`. Deletes a specific application bundle. Takes a full path of a file that contains a single GUID in each line as its arguments.
- `/dl`. Deletes a specific application bundle. Takes comma-separated GUIDs as arguments.
- `/list`. Lists the content of the newly created AppStack file.
- `/meta`. Creates a JSON file from the output AppStack file.
- `/vhd` Creates a VHD output AppStack file from VMDK AppStack input files.
- `/vmdk`. Creates a VMDK output AppStack file from VHD AppStack input files.

See also AppCapture Command-Line Options.

Update an AppStack from the Command Line

You update an AppStack to add applications, update existing applications, or remove applications from the AppStack.

Prerequisites

Verify that you have the correct credentials and you are taking the appropriate precautions:

- Run AppCapture as administrator.
- Create at least one AppStack.

- Disable User Account Control (UAC) notifications on the provisioning machine. See http://windows.microsoft.com/en-us/windows/turn-user-account-control-on-off#1TC=windows-7.
- Become familiar with the command options that apply to updating an AppStack. See AppCapture Command-Line Options).

Procedure

1 Open the command prompt and navigate to the AppCapture folder with either `cd "\Program Files\VMware\AppCapture"` (64-bit) or `cd "\Program Files (x86)\VMware\AppCapture"` (32-bit).

2 Update an AppStack:

   a Run `AppCapture.exe /n` *`appStackName`* `/s` *`sourceAppStackDir`*.

   sourceAppStackDir is the path of the AppStack that you want to update.

   This example takes an existing AppStack and updates it into a new update AppStack:

   ```
   AppCapture.exe /n AdminUser2.0 /s
   "C:ProgramData\VMware\AppCapture\appvhds\AdminUser1.0" /o C:\NewFolder
   ```

   You can include other command options that apply to updating an AppStack.

   The AppStack is created and stored in the location that you specify, or by default in the `appvhds` folder.

   b Add applications, update existing applications, or remove applications from the AppStack.

Task	Action
Add applications or update existing applications	Run the installers for the applications that you want to install or update on the AppStack.
(Optional) Remove applications	1 Navigate to **Control Panel > Programs and Features**. 2 Select the applications that you want to remove from the AppStack and complete the uninstall procedure.

3 After you add or remove the applications, navigate to the command prompt and press **Enter**.

4 Press **Enter** to restart the machine and finalize the AppStack update procedure.

After the machine restarts, the JSON, VHD, and VMDK files are created. When the application capture process finishes, the applications are removed from the machine.

Using AppCapture with Microsoft PowerShell

You can use Microsoft PowerShell cmdlets to capture applications, create and update AppStacks, and recreate deleted AppStacks with AppCapture. You can use the 32-bit or 64-bit PowerShell console to run the AppCapture module.

You can also run AppCapture from the command line, as described in Run AppCapture from the Command Line.

Note You must capture applications from the same OS into which you mount them. For example, if users are operating a Win7x64 OS, you must capture the applications by using a similar or an identical base OS Win7x64 image.

Run AppCapture Using PowerShell

You can run AppCapture using Microsoft PowerShell.

Prerequisites

Verify that you are logged in as administrator and you are taking the appropriate precautions:

- Run AppCapture as administrator.
- Disable User Account Control (UAC). See http://windows.microsoft.com/en-us/windows/turn-user-account-control-on-off#1TC=windows-7
- Become familiar with the AppCapture cmdlets. See PowerShell Options and Parameters

Procedure

1. Take a snapshot of the system.

   You can revert to the snapshot after the capture session.

2. Open a 32-bit or 64-bit PowerShell console,
3. Import the PowerCLI module using the **`import-module vmware.appcapture`** command.

   This imports the AppCapture module.

4. (optional) To see a list of all modules, run the **`get-module`** command.
5. Run the command `Start-AVAppCapture -Name` *appStackFile*, where *appStackFile* is the name of the AppStack `.vhd` file to create.

   Do not press **Enter** yet.

   appStackFile`.vhd` is created.

6. Leave the PowerShell console and install, on this machine, any applications to be provisioned.
7. After all of the applications have been installed, open the PowerShell console again.

8 Press **Enter**.

9 Reboot your machine if necessary.

In the AppCapture console window you see the locations of the AppStack files `.json`, `.vhd` and `.vmdk`. By default, these files are stored in `C:\ProgramData\VMware\AppCapture\appvhds`.

10 (Optional) Examine the `.json`, `.vhd`, and `.vmdk` files in that directory to ensure that the applications have been bundled.

11 Copy the AppStacks that you have created to a staging file share.

12 Revert to the system snapshot that you captured before you started the first capture session.

13 Copy the AppStacks from the staging file share to your system.

PowerShell Options and Parameters

You can use several options when you run AppCapture with Microsoft PowerShell.

AppCapture Options and Parameters with PowerShell

Use the `Start-AVAppCapture` to create an AppStack and add applications to it. The UEM Application Profiler is installed with the AppCapture utility and you can personalize the AppStacks using the profiler.

Table 1-2. `Start-AVAppCapture` Options

`Start-AVAppCapture` Parameter	**Description**
`-Author` *Author-name*	Specify an author who is associated with this AppStack.
CommonParameters	Use one or more common parameters. The common parameters are a set of cmdlet parameters implemented by Windows PowerShell. `Start-AVAppCapture` supports these common parameters: ■ Debug ■ ErrorAction ■ ErrorVariable ■ OutBuffer ■ OutVariable ■ PipelineVariable ■ Verbose ■ WarningAction ■ WarningVariable For more information about common parameters, see about_CommonParameters.
`-Description` *text*	Specify a description for an AppStack. If the description includes a space, enter the description inside parentheses, for example, **`-Description (HR Apps)`**.
`-Destination` *output-directory*	Specify an output directory for an AppStack. By default, AppStacks are placed in `C:\ProgramData\VMware\AppCapture\appvhds`.
`-Force`	Create an output directory if it does not exist. You specify the output directory with the `-Destination` parameter.

Table 1-2. `Start–AVAppCapture` Options (Continued)

`Start–AVAppCapture` **Parameter**	**Description**
`–Name` *vhd-name*	Specify a name for the applications being captured. The output `.vhd` file is named by using the specified application name.
`–Novmdk`	Specify this option to prevent post-capture VMDK disk creation.
`–Path` *directory-path*	Specify a path to an AppStack. The AppStack is used as a template for the current capture. Do not use this option if you are installing a new application.

You can perform several workflows with the AppCapture command.

Table 1-3. AppCapture PowerShell Workflows

Workflow	**Description**
`ConvertTo–AVVhdDisk`	Generate a `.vhd` file by using the `.vmdk` file as input.
`ConvertTo–AVVmdkDisk`	Generate a `.vmdk` file by using the `.vhd` file as input.
`Export–AVMetadata`	Generate a `.json` file by using a `.vhd` or `.vmdk` file as input.
`Merge–AVAppDisks`	Merge AppStack `.vhd` files into a new AppStack `.vhd`. Merging AppStacks describes the command-line version, which is similar.
`Remove–AVApp`	Delete an AppStack from a disk or remove specific applications from an AppStack. If you remove any applications from the AppStack, the AppStack must be imported again into the App Volumes Manager.
`Reset–AVConfig`	Clear AppCapture configuration information from the machine
`Show–AVDiskDetails`	List the contents of the `.vhd` file, `.json` file, or `.vmdk` file.
`Start–AVAppCapture`	Start the procedure to capture applications.
`Start–AVAppUpdate`	Update an AppStack.
`Start–AVAppPersonalization`	Attach the AppStack (`.vhd`) and personalize the specified application bundle using the UEM Application Profiler.

The examples below include the workflow file paths and the commands to reach the workflows.

- Begin a new capture session. The output is generated in the form of a `.vhd` file and is named *`AdobeSuite.vhd`*. The author is *John* and a description is added.

  ```
  Start–AVAppCapture –Name AdobeSuite –Author John –Description "This disk contains
  the AdobeSuite application"
  ```

- `ConverTo–AVVhdDisk`. This example generates an output `.vhd` format file, `Adobe.vhd`, from a source file, `Adobe.vmdk`. The output file is placed in a different directory from the source file:

  ```
  ConvertTo–AVVhdDisk –Path "C:\Program Files
  (x86)\VMware\AppCapture\appvhds\Adobe.vmdk" –Destination "C:\AppCaptures"
  ```

- `Export-AVMetadata`. This example generates the output metadata file `Adobe.json`. The file is generated in the same place as `Adobe.vhd`:

  ```
  Export-AVMetadata -Path "C:\Program Files
  (x86)\VMware\AppCapture\appvhds\Adobe.vhd"
  ```

- `Merge-AVAppDisks`. This example merges all the `.vhd` files under the `.\temp` and `.\appstacks` directories and generates a `Notepad+Adobe.vhd` file in `C\temp`.

  ```
  Merge-AVAppDisks -Path .\temp\*.vhd .\appstacks\*.vhd -Destination
  c:\temp\Notepad+Adobe.vhd
  ```

- `Remove-AVApp`. This example deletes the Adobe and Notepad applications from the input disk `Adobe+Notepad.vhd`. Each application is identified by its unique GUID:

  ```
  Remove-AVApp -Path C:\Temp\Adobe+Notepad.vhd -Destination c:\Temp\empty.vhd -
  Guids GUID1, GUID2
  ```

- `Show-AVDiskDetails`. This example displays the details from a `.json` file. The syntax is the same for `.vhd` and `.vmdk` files:

  ```
  Show-AVDiskDetails -Path "C:\Program Files (x86)\VMware\WEM
  Capture\appvhds\Adobe.json"
  ```

- `Start-AVAppUdate`. This example updates the `AdobeSuite.vhd` with a hot fix. A copy of `AdobeSuite.vhd` is created and is named `AdobeHotfixUpdate.vhd`. All the hot fix installations are captured in `AdobeHotfixUpdate.vhd`:

  ```
  Start-AVAppUpdate -Name AdobeHotfixUpdate -Path "C:\Program Files
  (x86)\VMware\AppCapture\appvhds\AdobeSuite.vhd"
  ```

- `Start-AVAppPersonalization -Path`. This cmdlet attaches the VHD and enables the user to personalize the application bundle using the UEM application profiler. Personalization settings are saved in `C:\ProgramData\VMware\AppCapture\appvhds\UEMConfigFiles\Chrome`.

  ```
  Start-AVAppPersonalization -Path
  "C:\ProgramData\VMware\AppCapture\appvhds\Chrome.vhd"
  ```

  - **`Start-AVAppPersonalization -Path "C:\appvhds\Chrome.vhd" -Predef`**. This cmdlet attaches the VHD and enables the user to personalize the application bundle using the UEM application profiler. Predefined settings and personalization settings are saved in `C:\ProgramData\VMware\AppCapture\appvhds\UEMConfigFiles\Chrome`.
  - **`Start-AVAppPersonalization -Path "C:\appvhds\Chrome.vhd" -Name Browser1`**. This cmdlet attaches the VHD and enables the user to personalize the application bundle using the UEM application profiler. Personalization settings files are saved in `C:\ProgramData\VMware\AppCapture\appvhds\UEMConfigFiles\Browser1`.

To get help about the workflows, run the **`get-help`** command.

Table 1-4. AppCapture PowerShell Workflow Information and Examples

Command	Description
`get-help` *WorkFlowName*	View general information for a workflow.
`get-help` *WorkFlowName* `-detailed`	View detailed information for a workflow.
`get-help` *WorkFlowName* `-examples`	View an example of a workflow.
`get-help` *WorkFlowName* `-full`	View technical information for a workflow.

AppCapture Folders and Files

AppCapture creates several files and folders.

AppCapture creates various folders in `C:\ProgramData\VMware\AppCapture\appvhds`.

Table 1-5. AppCapture Folders

Folder	Description
`appvhds`	`.vhd`, `.json`, and `.vdmk` files that are generated when you create an AppStack by using AppCapture.
`logs`	Log file generated by AppCapture. The log file is named `AppCapture.log` and is located in `C:\ProgramData\VMware\AppCapture\logs`.
`modules`	PowerCLI `.dll` files that are required to perform PowerCLI operations.
`plugins`	VMware Horizon Air Hybrid-Mode plug-ins. Plug-ins convert the AppStack to the correct format for deployment to end users.
`templates`	`.vhd` file templates that act as boilerplate `.vhd` files on which AppStacks are created.

AppCapture creates these files in the `appvhds` directory unless you specify a different directory. See AppCapture Command-Line Options.

Table 1-6. AppCapture Files

File	Description
application`.vhd`	`.vhd` file that holds the application files that are part of the AppStack.
application`.vmdk`	VMDK-format Virtual Hard Disk file that VMware Horizon Air Hybrid-Mode natively uses.
application`.json`	The `.json` file with information about the applications that are captured in the AppStack.