

# Rapid Cloning Utility 1.0

## 1. Overview

The Rapid Cloning Utility allows users to quickly create and deploy VMware virtual machines using Flexclone technology in an automated fashion, without having to develop custom scripts. The primary tool, `na_vi_clone`, allows customers to quickly and efficiently clone datastores, create virtual machine clones in VirtualCenter, power up the new virtual machines, and apply guest customization specifications to the guest operating system in each. The two supporting tools are `rfwm` (an ESX service console tool) and `na_vi_util.exe`. Neither of these tools is required, they are provided because they offer additional efficiencies and convenience.

The zip file package delivered for the script (`RapidCloningUtilityPackage.zip`) contains the following:

- `RCU_1.0_README.pdf`: This document.
- `na_vi_clone.exe`: An executable that runs on Windows Operating Systems.
- `na_vi_util.exe`: An executable that runs on Windows Operating Systems.
- `rfwm`: An executable that runs on the ESX service console.

### 1.1 na\_vi\_clone

`na_vi_clone.exe` takes a repository of “golden” virtual disk files and creates virtual machines based on them. The hardware configuration of the resulting virtual machines is based on a VirtualCenter Virtual Machine Template that is specified at execution of the tool. These templates should not include a virtual hard disk at SCSI address 0, because this disk will be added to the VM from the repository at creation. The tool can be used to apply a VirtualCenter Guest Customization Specification to the new VMs as well. This option requires that the “Microsoft Sysprep Tools” have been properly added to the VirtualCenter Server. Please see the “Basic System Administration Guide” from VMware for more information on the how to do this.

This tool can automatically create and attach the newly created FlexClones to the ESX servers as datastores in NFS environments. In iSCSI or FCP environments, this must be done manually.

### 1.2 rfwm

`rfwm`, or read few write many, creates multiple copies of `vmdk` or `vhd` files for use as a golden directory referred to as the ‘`golddir`’ or simply the repository. `rfwm` runs on an ESX service console and can be used with VMFS or NFS based datastores. This tool can also be run from a linux server connected to an NFS datastore.

The goal of `rfwm` is to reduce the performance impact on the virtual infrastructure required to make a golden datastores, but do so as fast or faster than traditional methods. It accomplishes this by reducing the number of read operations and optionally reducing the number of write operations. Fewer reads are required because by default, `rfwm` writes each block read to 8 different files. It can also reduce the number of writes by not writing the blocks that makeup the unused portion of the `vmdk` file.

## 1.3 na\_vi\_util

The na\_vi\_util tool provides the ability to take an action on a group of virtual machines. The group of virtual machines effected is based on the prefix of the name of the VM. The actions that can be taken are listed below.

|                 |                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| answeryes       | Answer 'yes' to a question blocking the VM(s) from starting.                                                                                      |
| answerno        | Answer 'no' to a question blocking the VM(s) from starting.                                                                                       |
| count           | Display the number of VM(s) that match the --vmprefix option. The --list option can be used to show the VM name(s) that match.                    |
| countrunning    | Display the number of VM(s) that match the --vmprefix option and are powered on. The --list option can be used to show the VM name(s) that match. |
| poweron         | Power on the VM(s) that match the --vmprefix option and are powered off.                                                                          |
| poweron_perhost | Power on a maximum number of VM(s) on each ESX server that matches the --vmprefix option and is powered off. Requires --maxperhost option.        |
| shutdown        | Request that the 'VMware tools' shutdown the guest OS in the VM(s) that match the --vmprefix option and are powered on.                           |
| poweroff        | Power off the VM(s) that match the --vmprefix option and are powered on.                                                                          |
| cust            | Apply a 'Guest Customization Specification' to the VM(s) that match the --vmprefix option and are powered off. Requires --custspec option.        |
| refreshstorage  | Not a VM command, but still useful. This will refresh each ESX servers' storage system.                                                           |

## 2. Installation

☐ **Step 1:** Make sure that Microsoft .NET Framework 2.0 Service Pack 1 is installed on the host system that will be running the na\_vi\_clone tool.

☐ **Step 2:** Unzip the RapidCloningUtilityPackage.zip file.

☐ **Step 3 (optional):** Copy the rfwm binary to an ESX server. On the ESX service console, use the chmod command to make it executable:

```
chmod 555 rfwm
```

### 3. Prerequisites for using na\_vi\_clone

☐ Ensure that the golden VMDK is created for use with the **LSI Logic SCSI Adapter**. A vmdk created with the Bus Logic SCSI Adapter can be changed to a LSI type. You will need to make sure that the guest operating system has the drivers capable of supporting the LSI Logic SCSI adapter. The adapter can be changed by selecting the “SCSI Controller 0” item in the “Virtual Machine Properties” window and clicking in the “Change Type...” button. You will then need to start the VM and answer yes to the “Virtual Machine Question” when prompted. This should be done in the golden virtual machine (the one you are going to copy the vmdk files from).

☐ Ensure that Microsoft .NET Framework 2.0 Service Pack 1 is installed on the host system that will be running the Rapid Cloning Utility.

☐ In the case of NFS, the name of the datastore (in ESX) must match the name of the volume (on the NetApp Controller) and cannot contain any spaces.

☐ Create the golden repository (directory) within a new datastore using the VirtualCenter Client or the ESX service console. To create a new directory using the VirtualCenter Client, right-click the datastore and choose “Browse Datastore.” Once the browser window appears, create a new directory in the datastore.

Alternatively, you can create the directory using the following command from the ESX service console:

```
mkdir /vmfs/volumes/gold/hr_winxp
```

**Note:** The name of the repository cannot contain any spaces. Ensure that you are not using any mount points with the same name as the repository. Further, the name of the repository will be used as the prefix of all newly created datastores.

☐ Create clones of the golden vmdk file from the ESX service console in the golden repository (directory).

To create a single clone of the golden vmdk file, use the command below:

```
vmkfstools -i golden.vmdk ../../gold/hr_winxp/clone001.vmdk
```

To create 8 copies, use the loop example below:

```
for clone in `seq 0 8`;do
  vmkfstools -i golden.vmdk ../../gold/hr_winxp/clone00${clone}.vmdk
done
```

This example uses the rfwm tool provided in the Rapid Cloning Utility to create 64 copies:

```
rfwm --copies 64 golden.vmdk ../../gold/hr_winxp/
```

☐ **(optional):** Create a customization specification. If specified as a command-line argument, this customization specification will be applied to all newly created virtual machine clones. The use of this option requires that the appropriate sysprep tool(s) for the guest operating system(s) have been installed on the VirtualCenter server. For more information, please see the “Basic System Administration” guide from VMware.

☐ Create a virtual machine template to base the new VMs on. Ensure that during the virtual machine creation wizard, the custom configuration option is selected (See figure 3). When you reach the “Select a Disk” wizard page, make sure you select the “Do not create disk” option (See figure 4).

An existing template or VM can be used as well, however virtual device node SCSI (0:0) will need to be removed from the configuration.

Note: Additional disks may be added to the template as needed as long as the additional drives start at virtual device node SCSI (0:1). These drives will be duplicated as part of the VMware cloning process (not FlexClone), as such will increase the time it takes to deploy.

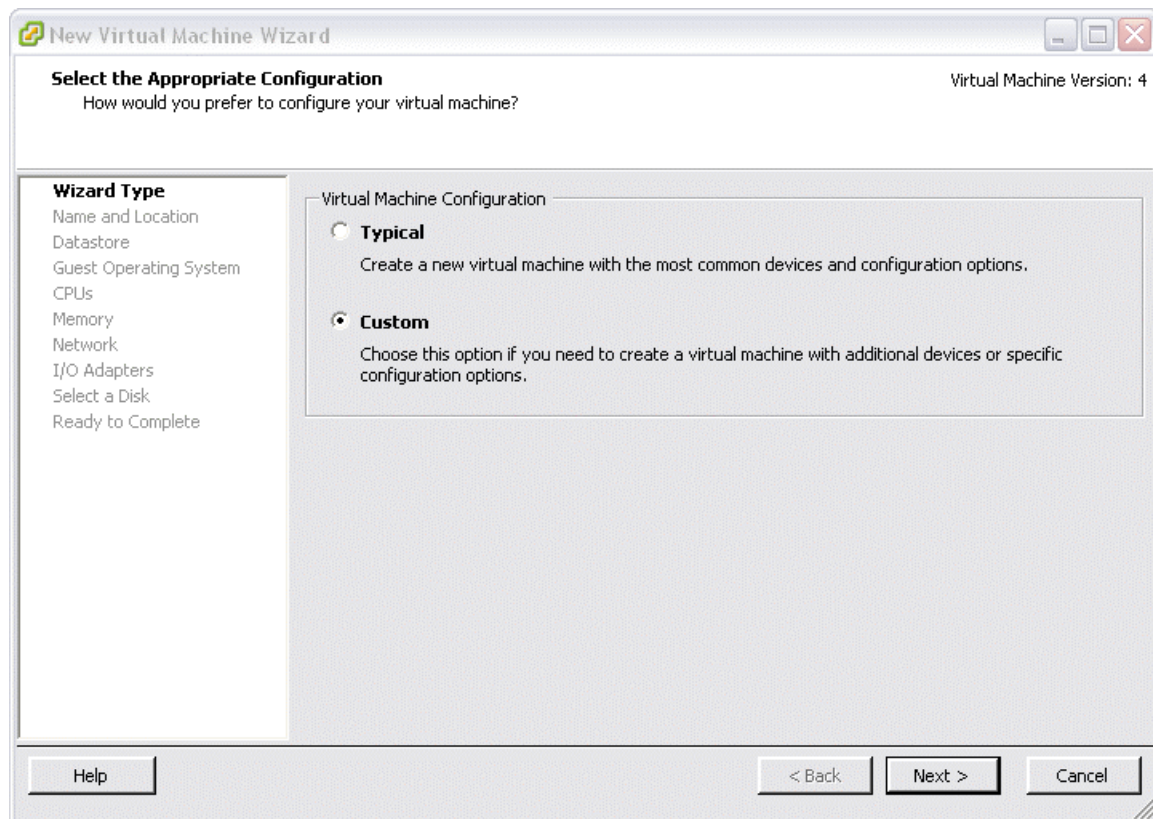


**Figure 3:** Select “Custom” during template configuration

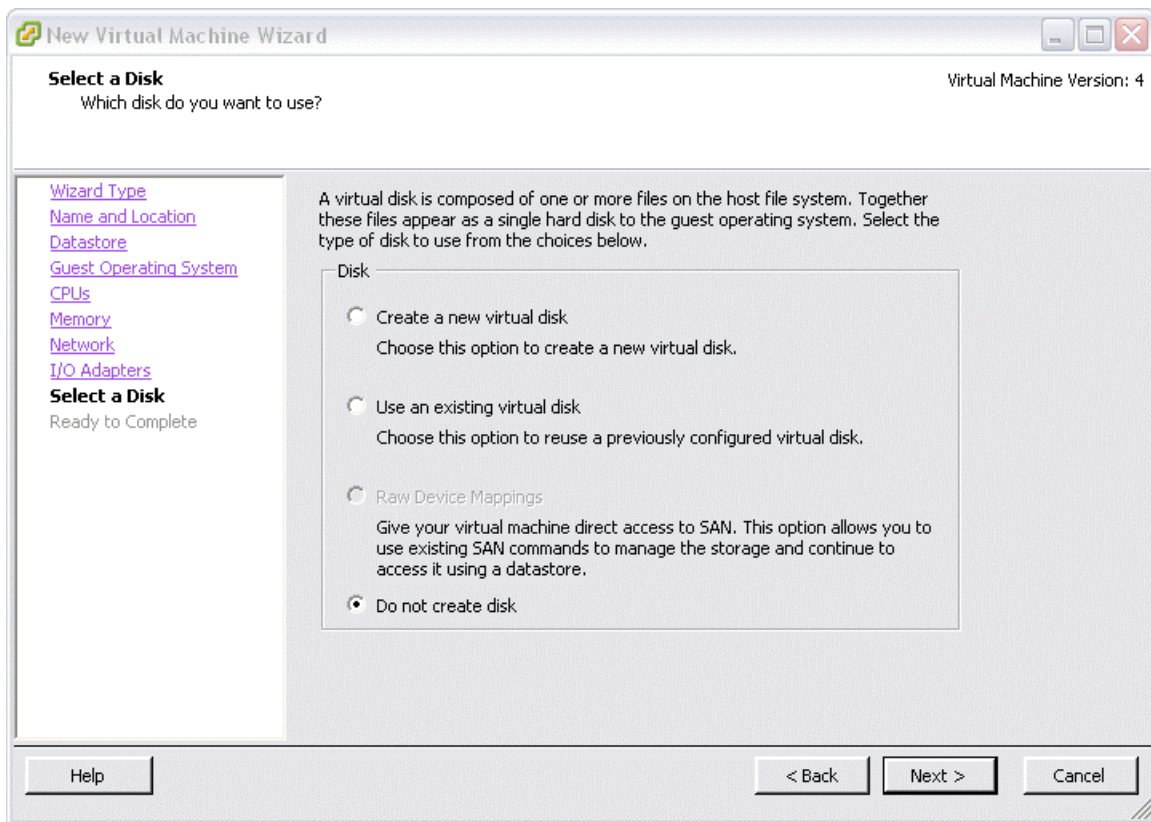


Figure 4: Select “Do not create disk” during template configuration

## 4. Execution of na\_vi\_clone

Most arguments specified will be taken from the list of command specific arguments below. Please note that most of the Common VI options can be safely ignored.

### Command specific arguments:

#### **--datacenter <datacenter>**

Required argument. This argument specifies the name of the datacenter where the virtual machines will be managed.

#### **--golddir <golden directory>**

Required argument. This argument specifies the name of the repository or golden directory where unused VMDK files should be taken from. This directory was created and populated in the Prerequisites.

#### **--goldvol <golden volume>**

Required argument. This argument specifies the name of the golden volume (flexvol) that all new datastores will be based on. This volume was created in step 1 of the Prerequisites (above). In the case of an iSCSI or FCP environment, this option should specify the name of the datastore.

#### **--template <template>**

Required argument. This argument specifies the name of the Virtual Machine template to use when cloning new Virtual Machines. This template (or virtual machine) was created in the Prerequisites (above).

#### **--fas <FAS hostname or IP address>**

Required argument. This argument specifies the hostname or IP address of the NetApp storage controller.

#### **--fasuser <FAS username>**

Required argument. This argument specifies the username used to connect to the NetApp storage controller.

#### **--fasspass <FAS password>**

Required argument. This argument specifies the password used to connect to the NetApp storage controller.

#### **--custspec <customization specification>**

Optional argument. This argument specifies the name of the "Guest Customization Specification" that will be applied to the newly cloned Virtual Machines. This specification must have been configured and exist within Virtual Center. This was discussed the Prerequisites (above).

#### **--datastores <number of datastores>**

Optional argument for NFS only. This argument specifies the number of datastores that will be created from the goldvol. These datastores will be added to Virtual Center.

#### **--vms <number of virtual machines>**

Optional argument. This argument is used to specify the number of virtual machine clones to create. If not specified, the default is to create a clone for each vmdk file found.

***--hostlimits <host limits>***

Optional argument. This argument specifies a semicolon-separated string of ESX hostname and the maximum number of guests to attach to that ESX server. For example, “--hostlimits sever1,24;server2,10;server3;0” would be limit the server 1 to a maximum of 24 new VMs, server2 to 10 new VMs and server3 would not be used at all. Not all ESX servers are used to create VMs unless the number of --threads option is equal to the number of ESX servers.

***--threads <threads>***

Optional argument. This argument is used to specify the number of task management threads the process should use.

***--logfile <path to log file>***

Optional argument. This argument specifies the path and name of the file used for logging. The default is na\_vi\_clone.log, which will be created in the current working directory.

***--poweron***

Optional argument. This argument specifies whether clones should be powered on once they are created.

***--vcfm***

Optional argument. This argument is used to specify whether the Virtual Center File Manager will be used to move the vmdk files to their new locations in the datastores. This is required in iSCSI and FCP based datastores.

***--fasssl***

Optional argument. This flag forces the tool to use HTTPS to communicate with the NetApp Controller. The default it to use HTTP. Please see the SecureAdmin feature discussed in the NetApp “Data ONTAP® System Administration Guide” for more information.

### Common VI options:

**--server <server>**

Required argument. This argument specifies the hostname or IP address of the VirtualCenter server to run the tool against.

**--username <username>**

Optional argument. This argument specifies the username used to connect to the VirtualCenter server. If not specified, the user will be prompted for the username.

**--password <password>**

Optional argument. This argument specifies the password used to connect to the VirtualCenter server. If not specified, the user will be prompted for the password.

**--config <path to VI Perl configuration file>**

Optional argument. This argument specifies the path to the VI Perl configuration file.

**--encoding <encoding mechanism>**

Optional argument. This argument specifies the encoding mechanism to use. Choices include UTF8, CP936, ISO-8859-1, and shiftjis. The default is "UTF8"

**--passthroughauth**

Optional argument. This argument is used if you would like to attempt to use pass-through authentication.

**--passthroughauthpackage <authentication negotiation package>**

Optional argument. This argument specifies the pass-through authentication negotiation package. The default is "Negotiate".

**--portnumber <port number>**

Optional argument. This argument specifies the port number to use when connecting to the VirtualCenter server.

**--protocol <protocol>**

Optional argument. This argument specifies the protocol to use when connecting to the VirtualCenter server. The default is "HTTPS".

**--savesessionfile <file path>**

Optional argument. This argument specifies the path to the file that should be saved which contains the session ID/cookie used when running the tool.

**--servicepath <service path>**

Optional argument. This argument specifies the service path to use when connecting to the VirtualCenter server. The default is "/sdk/webService".

**--sessionfile <file path>**

Optional argument. This argument specifies the path to the file containing the session ID/cookie to use when running the tool.

**--url <url>**

Optional argument. This argument specifies the VI SDK URL to connect to and is required if the server argument is not specified.

***--verbose***

Optional argument. This argument displays additional debugging information for the APIs sent via the VI SDK to the VirtualCenter server.

***--version***

Optional argument. This argument displays the version information for the tool.

***--help***

Optional argument. This argument displays usage information for the tool.

## 5. Sample Execution of na\_vi\_clone

The examples below are based on the following prerequisite steps being completed:

- 1) A volume named /vol/myGoldVolume was created on the NetApp storage controller.
- 2) The volume was attached to at least 1 ESX server as a datastore.
- 3) A directory named xpsp3 was created in this datastore.
- 4) A vmdk file from an existing virtual machine (or template) was copied 10 times into this directory.

### **Example 1:**

```
C:\>na_vi_clone.exe --datacenter "Charlotte, NC" --server 10.60.60.75 --username Admin --password netapp --template TEMPLATE --goldldir xpsp3 --goldvol myGoldVolume --threads 4 --fas fas3050-2.acme.com --fasuser root --faspass password --datastores 5 --custspec "my Windows XP spec" --poweron
```

This will use NetApp FlexClone technology to create 5 new datastores and attach them to each ESX server managed by the specified VC server. Each of the 5 new datastores is an exact copy of the volume/datastore named 'myGoldVolume' specified in the -goldvol option above. Because there were 10 vmdk files copied into this volume in the prerequisite steps, there are now a total of 50 distinct copies of the original vmdk file available for use in the vm cloning process. It will then create 50 new virtual machine clones, based on the vm template named "TEMPLATE", by associating one of the 50 copies of the vmdk with each of the virtual machine clones created. Once the virtual machines are created, "my Windows XP spec" will be applied to each of them and they will each be powered on.

### **Example 2:**

```
C:\>na_vi_clone.exe --datacenter "Durham, NC" --server 10.60.60.76 --username Admin --password netapp --template TEMPLATE --goldldir xpsp3 --goldvol myGoldVolume --threads 4 --fas fas3050-3.acme.com --fasuser root --faspass password --vms 50 --custspec "my Windows XP spec"
```

This run will create and register 50 additional virtual machines from the existing xpsp3 directory. If there are not enough VMDK files, the tool will create as many as it can up to 50. Once created, the "my Windows XP spec" will be applied to each new virtual machine.

## 6. Using rfwm

### 6.1 rfwm Options

The rfwm tool has several options not shown in section 3. These options are listed in the table below.

|              |                                                                                                                                                                                                                    |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --thick      | The default behavior is to create a sparse file. Specifying this flag forces a block for block copy even in the case that a block is all zeros.                                                                    |
| --threads    | This option specifies how many writes (one per file) are done per read. The default is 8, which means that a max of 8 files will be written to for each read.                                                      |
| --copies     | This option specifies how many copies of the vmdk file to make. The default is 1. If the number of copies is greater than the --threads value, rfwm will loop until it has created the requested number of copies. |
| --first      | This option is used to specify what number filename to start creating. The default is 1. This is useful when adding additional files to a goldir. NOTE: This has no impact on the number of files created.         |
| --bs         | The number following this option is used to specify the block size of each IO (specified in kB). This must be one of (8, 16, 32, 64, 128, 1024). The default is 8.                                                 |
| --prefix     | Specifies the prefix to be used in the name of each new vmdk.                                                                                                                                                      |
| --vmkfstools | Use the VMware vmkfstools to create skeleton vmfs file pairs. This is only valid on the ESX service console, and ignored in all other cases.                                                                       |

### 6.2 rfwm Example

The following example will create 8 new vmdk files in the /vmfs/volumes/gold/hr\_winxp directory. The file names will be dev\_001.vmdk (and dev\_001-flat.vmdk) through dev\_008.vmdk (and dev\_008-flat.vmdk).

The second and third lines in this example are the output during the execution of rfwm. rfwm reports throughput statistics for the previous few seconds as well as the percentage of the file that has been read and written. The throughput numbers are in parenthesis; the first being the total throughput, followed by the read throughput and finally the write throughput. The percent read reports how much of the source file has been processed. The percent written reports how much of that file has been written. When using non-sparse or thick format, these numbers will match. In this case, because we are running in the default 'sparse mode', the percent written is smaller.

```
[root xp]# rfwm --copies 8--prefix dev winxp.vmdk ../../gold/hr_winxp/
rfwm: stats (last 13.24 seconds): total (51.96 MB/s),
0.76% read (5.90 MB/s), 0.74% written (46.06 MB/s).
```

## 7. Using na\_vi\_util

### 7.1 na\_vi\_util Options

|              |                                                                                                                                                                                           |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --action     | What action to take on matching VM(s). See the table in section 1.3 of this document.                                                                                                     |
| --waitontask | Do not exit until all tasks have completed.                                                                                                                                               |
| --vmprefix   | A string to identify multiple virtual machines. This prefix is compared to the beginning of each VM name. The action specified in the --action is then taken on each of the matching VMs. |
| --custspec   | The name of the 'Guest Customization Specification' previously configured and saved within Virtual Center. This only applies to 'cust' action.                                            |
| --maxperhost | The maximum number of VMs to apply an action to per ESX server. This only applies to the 'poweron_perhost' action                                                                         |

### 7.2 na\_vi\_util Examples

**Power on all VMs starting with hrdev:**

```
na_vi_util.exe --datacenter "MyDC" --server 10.1.23.9 --vmprefix hrdev  
--action poweron
```

**Customize all VMs starting with hrdev using the "winxp\_join\_dev\_domain" specification:**

```
na_vi_util.exe --datacenter "MyDC" --server 10.1.23.9 --vmprefix hrdev  
--action cust --custspec winxp_join_dev_domain
```

**Request an orderly shutdown of all VMs starting with test:**

```
na_vi_util.exe --datacenter "MyDC" --server 10.1.23.9 --vmprefix test  
--action shutdown
```

**Count and list the VMs that will match the --vmprefix of test\_0:**

```
na_vi_util.exe --datacenter "MyDC" --server 10.1.23.9 --vmprefix test_0  
--action count --list
```

## 8. Additional Information

**“Network Appliance and VMware Virtual Desktop Infrastructure “**

<http://www.netapp.com/us/library/technical-reports/tr-3612.html>

**“NetApp and VMware VDI Best Practices”**

<http://www.netapp.com/us/library/technical-reports/tr-3705.html>

**“NetApp and VMware Virtual Infrastructure 3 Storage Best Practices”**

<http://www.netapp.com/us/library/technical-reports/tr-3428.html>