# Rapid Cloning Utility 3.0
# **Installation and Administration Guide**

NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089 U.S.A.
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: http://www.netapp.com

Part number 215-05044_A0
February 2010

# Copyright and trademark information

## Copyright information

Copyright © 1994-2010 NetApp, Inc. All rights reserved. Printed in the U.S.A.

The third-party copyright statements and licenses that apply to open source software components distributed with the product are contained in the section “Third-Party Copyrights and Licenses.”

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S.A. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark information

NetApp, the Network Appliance logo, the bolt design, NetApp—the Network Appliance Company, Cryptainer, Cryptoshred, DataFabric, DataFort, Data ONTAP, Decru, FAServer, FilerView, FlexClone, FlexVol, Manage ONTAP, MultiStore, NearStore, NetCache, NOW NetApp on the Web, SANscreen, SecureShare, SnapDrive, SnapLock, SnapManager, SnapMirror, SnapMover, SnapRestore, SnapValidator, SnapVault, Spinnaker Networks, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, StoreVault, SyncMirror, Topio, VFM, VFM (Virtual File Manager), and WAFL are registered trademarks of NetApp, Inc. in the U.S.A. and/or other countries. gFiler, Network Appliance, SnapCopy, Snapshot, and The evolution of storage are trademarks of NetApp, Inc. in the U.S.A. and/or other countries and registered trademarks in some other countries. The NetApp arch logo; the StoreVault logo; ApplianceWatch; BareMetal; Camera-to-Viewer; ComplianceClock; ComplianceJournal; ContentDirector; ContentFabric; EdgeFiler; FlexShare; FPolicy; Go Further, Faster; HyperSAN; InfoFabric; Lifetime Key Management, LockVault; NOW; ONTAPI; OpenKey, RAID-DP; ReplicatorX; RoboCache; RoboFiler; SecureAdmin; Serving Data by

Design; Shadow Tape; SharedStorage; Simplicore; Simulate ONTAP; Smart SAN; SnapCache; SnapDirector; SnapFilter; SnapMigrator; SnapSuite; SohoFiler; SpinMirror; SpinRestore; SpinShot; SpinStor; vFiler; Virtual File Manager; VPolicy; and Web Filer are trademarks of NetApp, Inc. in the U.S.A. and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of NetApp, Inc. in the U.S.A.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. A complete and current list of other IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml.

Apple is a registered trademark and QuickTime is a trademark of Apple, Inc. in the U.S.A. and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the U.S.A. and/or other countries. RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the U.S.A. and/or other countries.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

NetApp, Inc. is a licensee of the CompactFlash and CF Logo trademarks. NetApp, Inc. NetCache is certified RealSystem compatible.

# Table of Contents


**Preface** . . . . . vii

**Chapter 1** **Installing and Launching the Utility** . . . . . 1

Overview . . . . . 2

System Requirements . . . . . 3

Installing the Rapid Cloning Utility . . . . . 4

Launching the Rapid Cloning Utility . . . . . 5

**Chapter 2** **Cloning Virtual Machines** . . . . . 11

Cloning virtual machines . . . . . 12

Re-deploying clones . . . . . 23

Importing virtual machines into XenDesktop . . . . . 26

**Chapter 3** **Managing Storage Controllers and Datastores** . . . . . 27

Managing storage controllers . . . . . 28

Managing datastores . . . . . 34

**Chapter 4** **Troubleshooting** . . . . . 41

Troubleshooting . . . . . 42

**Appendix A** **Storage system permissions** . . . . . 45

Create clones role . . . . . 46

Create storage role . . . . . 48

Modify storage role . . . . . 49

Destroy storage role . . . . . 50

**Appendix B** **Preferences File, Logs, and Export Files** . . . . . 51

Preferences File . . . . . 52

Logs . . . . . 53
Export Files . . . . . 54

**Appendix C** **Programmable API** . . . . . 55
Overview . . . . . 56
Methods . . . . . 58
Specifications and messages . . . . . 77
Example . . . . . 92
Java . . . . . 94

# Preface

## About this guide

This document describes how to install and run the NetApp® Rapid Cloning Utility.

## Audience

This document is for VMware® system administrators who manage large numbers of desktop environments or servers.

## Terminology

To understand the concepts in this document, you might need to know the terms defined here.

- *Controller* or *storage controller* refers to the component of a storage system that runs the Data ONTAP® operating system and controls its disk subsystem. Controllers or storage controllers are also sometimes called *storage appliances*, *appliances*, *storage engines*, *heads*, *CPU modules*, or *controller modules*.

## Command, keyboard, and typographic conventions

This document uses command, keyboard, and typographic conventions that help you enter commands.

**Keyboard conventions:** The following list describes keyboard conventions used in this document:

- When describing key combinations, this document uses the hyphen (-) to separate individual keys. For example, “Ctrl-D” means pressing the “Control” and “D” keys simultaneously.
- This document uses the term “Enter” to refer to the key that generates the digital equivalent of a carriage return, although the key is named “Return” on some keyboards.

**Typographic conventions:** The following table describes typographic conventions used in this document.

| Convention | Type of information |
|---|---|
| *Italic* font | Words or characters that require special attention.<br>Placeholders for information you must supply. For example, if the guide says to enter the **`arp -d`** ***`hostname`*** command, you enter the characters "arp -d" followed by the actual name of the host.<br>Book titles in cross-references. |
| `Monospaced` font | Command names, option names, keywords, and daemon names.<br>Information displayed on the system console or other computer monitors.<br>The contents of files. |
| **`Bold monospaced`** font | Words or characters you type. What you type is always shown in lowercase letters, unless you must type it in uppercase letters. |

## Special messages

This document might contain the following types of messages to alert you to conditions you need to be aware of.

**Note**

A note contains important information that helps you install or operate the system efficiently.

**Attention**

An attention notice contains instructions that you must follow to avoid a system crash, loss of data, or damage to the equipment.

# Installing and Launching the Utility 1

**Topics in this chapter**

This chapter includes the following topics.


- "Overview" on page 2
- "System Requirements" on page 3
- "Installing the Rapid Cloning Utility" on page 4
- "Launching the Rapid Cloning Utility" on page 5

# Overview

## About the Rapid Cloning Utility

Using FlexClone® technology, the NetApp® Rapid Cloning Utility (RCU) allows users to quickly create, deploy and manage the lifecycle of VMware® virtual machines (VMs) from an easy-to-use interface integrated into VMware vCenter. The tool can be used to:

- Create clones of templates, virtual machines, or virtual machine snapshots, and deploy them into new or existing NetApp NFS and VMFS (FCP/iSCSI) datastores
- Apply guest customization specifications to the resulting virtual machines
- Provision, resize, deduplicate and deallocate datastores
- Deploy virtual machines for both server and desktop use
- Re-deploy virtual machines from a baseline image
- Monitor storage savings
- Import virtual machines into virtual desktop infrastructure connection brokers and management tools

The Rapid Cloning Utility can theoretically create thousands of virtual machine clones and hundreds of datastores in a single execution. In practice, however, multiple executions of smaller requests are recommended. The exact size of these requests depend on the size of the VMware Infrastructure 3 deployment and the hardware configuration of the vSphere Client managing the ESX hosts.

## System Requirements

See the *Rapid Cloning Utility Release Notes* at http://now.netapp.com for system requirements on the host running the RCU.

# Installing the Rapid Cloning Utility

Follow these steps to install the Rapid Cloning Utility:

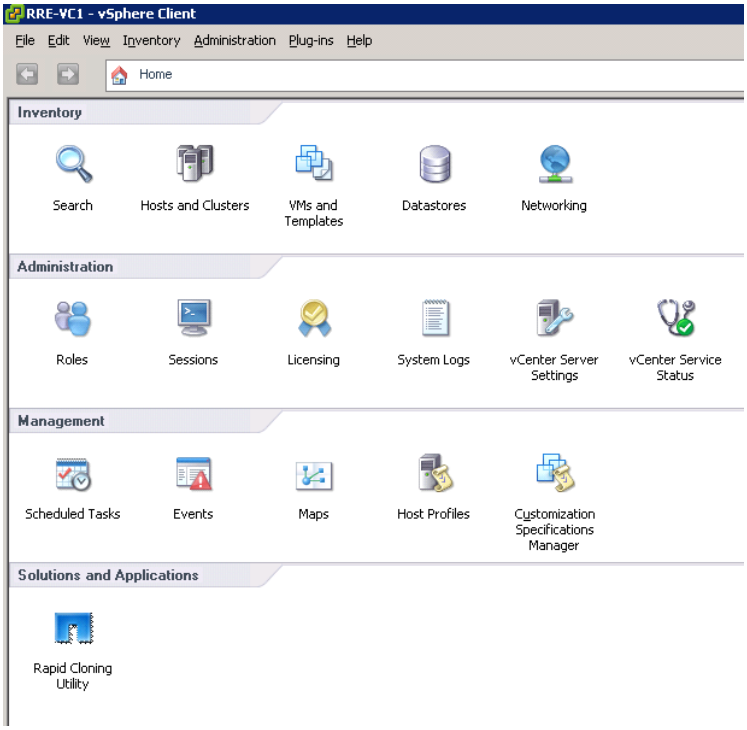
1. Make sure all prerequisites for the host system are met, as described in the *Rapid Cloning Utility Release Notes* at http://now.netapp.com.
2. Download the RCU setup executable appropriate for your system from http://now.netapp.com:
   - `RCUSetup_3_0_win32.exe`
   - `RCUSetup_3_0_win64.exe`
3. Save the file to your local file system.
4. Navigate to the folder containing the executable.
5. Double-click the executable and follow the on-screen directions.

# Launching the Rapid Cloning Utility

When you launch the RCU, you must import certificates and identify the controllers that will be used for cloning tasks.

1. In VMware vSphere Client, go to the **Solutions and Applications** tab and click the **Rapid Cloning Utility** icon.

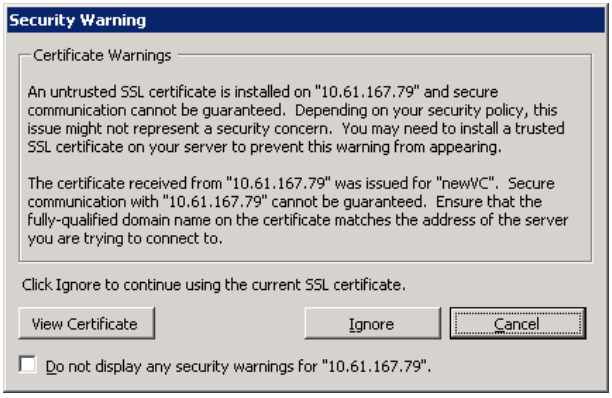
   You can also select **View > Solutions and Applications > Rapid Cloning Utility**.




### Import certificates

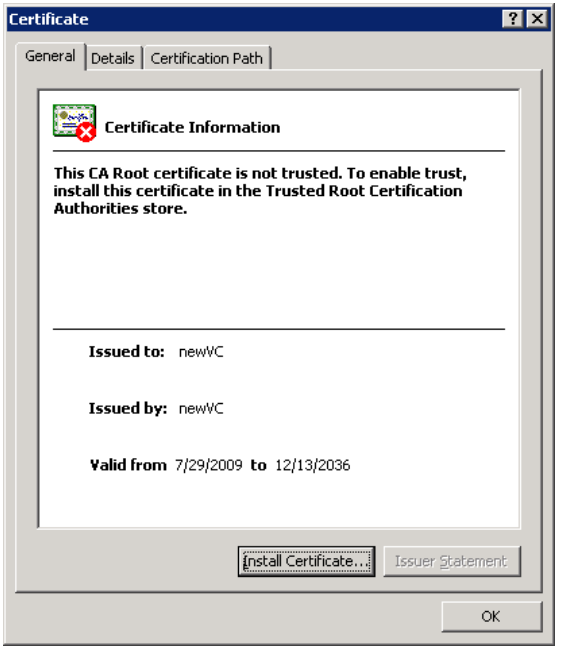
2. RCU uses SSL to encrypt communication to the vSphere Client. A “self signed” certificate is used in both cases. The first time the vSphere Client communicates with the RCU service, a certificate warning appears. To

prevent this warning from appearing again, select the **Do not display any security warnings...** check box and click **Ignore**.



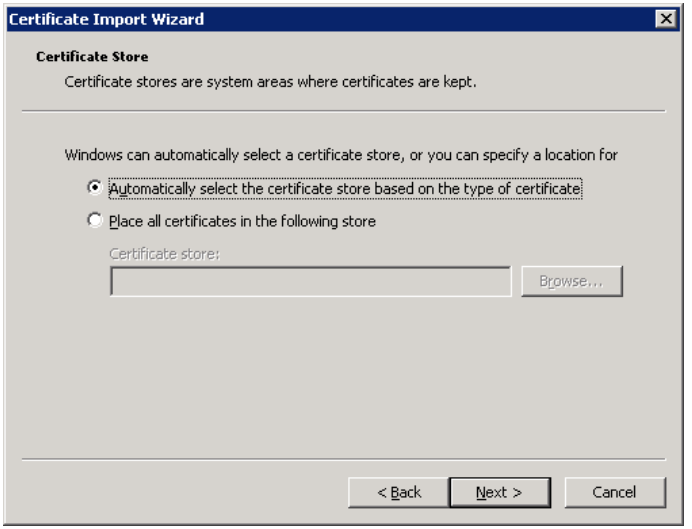

3. The first time you open Rapid Cloning Utility in vSphere Client, a similar warning displays. To prevent this warning from appearing each time, click **View Certificate**.

4. Click **Install Certificate…** to launch the Certificate Import Wizard.

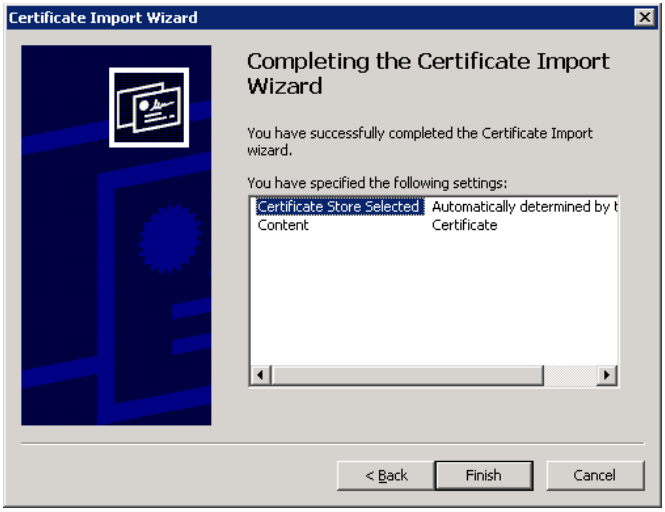



5. When the Certificate Import Wizard opens, click **Next**.

6. Accept the default and click **Next**.



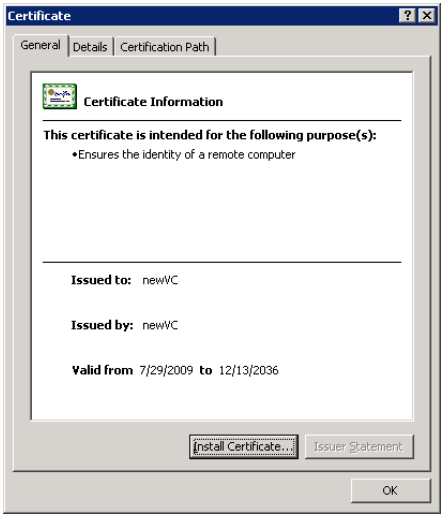
7. Click **Finish**.




8. Click **Yes** in the Security Warning window.

9. Click **OK**.

10. Click **OK** again.

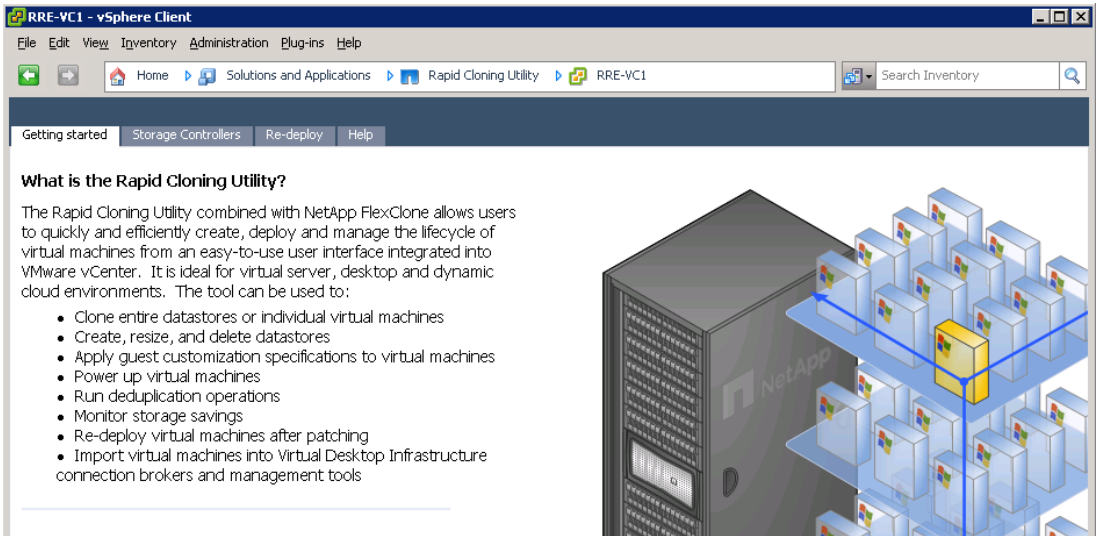



11. Click **Yes** in the Security Alert window.

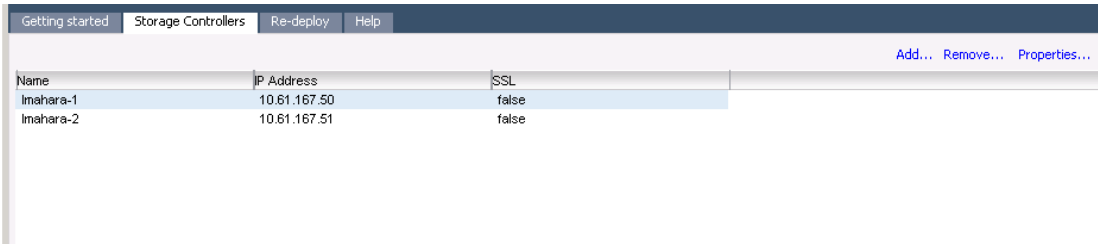
## Identify storage controllers

Before creating clones, identify the storage controllers to use for cloning.

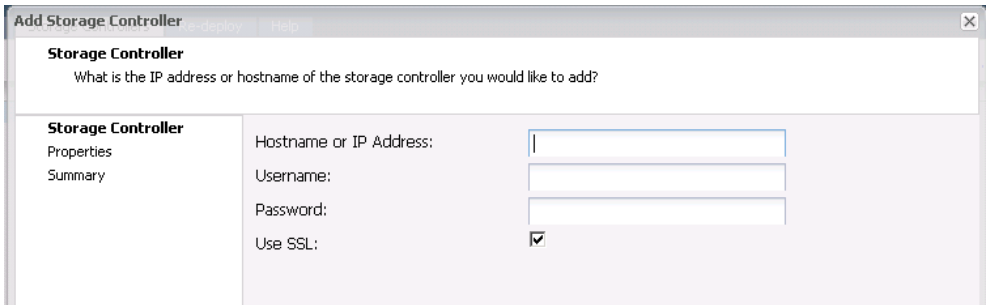
1. Click the **Storage Controllers** tab.




2. Click the blue **Add...** link.




3. Enter the hostname or IP address and credentials of the storage controller. Optionally deselect **Use SSL** if you do not want to encrypt communication between the RCU service and the storage controller. Click **OK** to continue.



4. Specify interfaces, volumes, and aggregates to use on the storage controller.

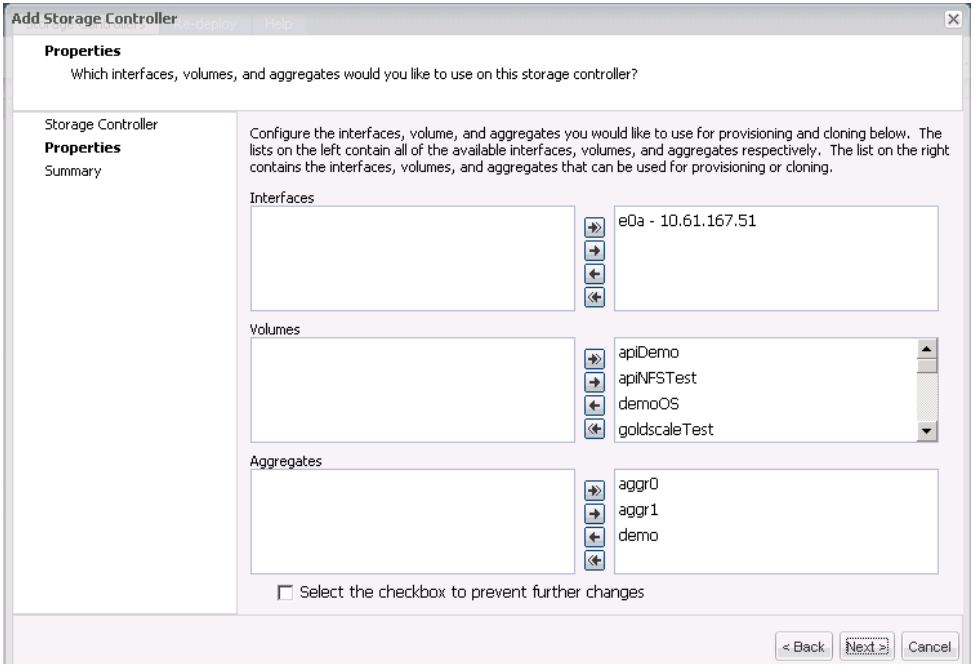
   a. Select components and click the directional buttons to move them to the desired column.

      - Items in the left column will *not* be used by RCU.
      - Items in the right column will be used by RCU.

**Note**

New volumes and aggregates created outside RCU appear in the left column as unused.

   b. To prevent users from modifying these settings without providing storage system credentials, select the check box at the bottom of the window.




**Note**

See “Managing storage controllers” on page 28 for additional information on managing storage controllers in the RCU.

# Cloning Virtual Machines 2

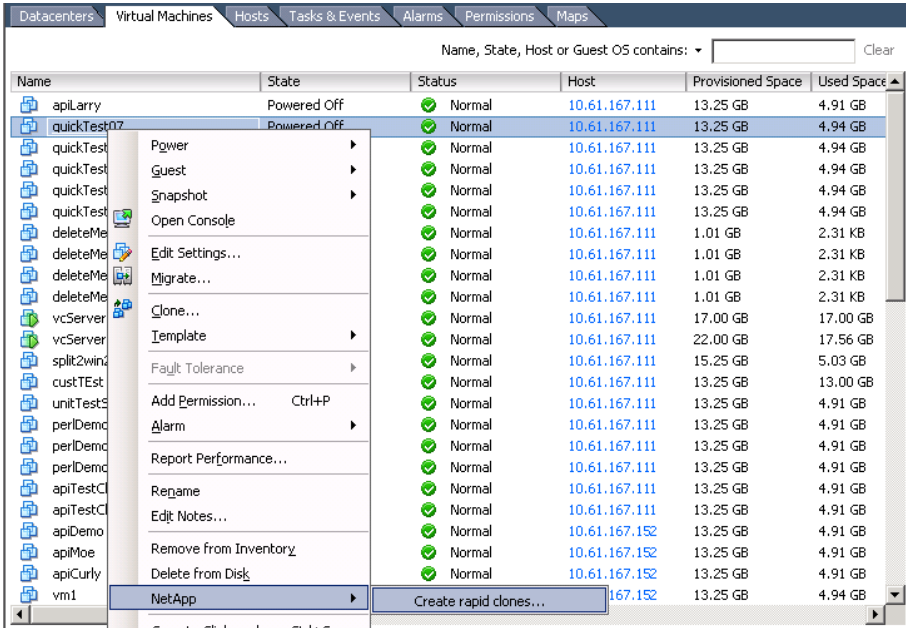
**Topics in this chapter**

This chapter includes the following topics.

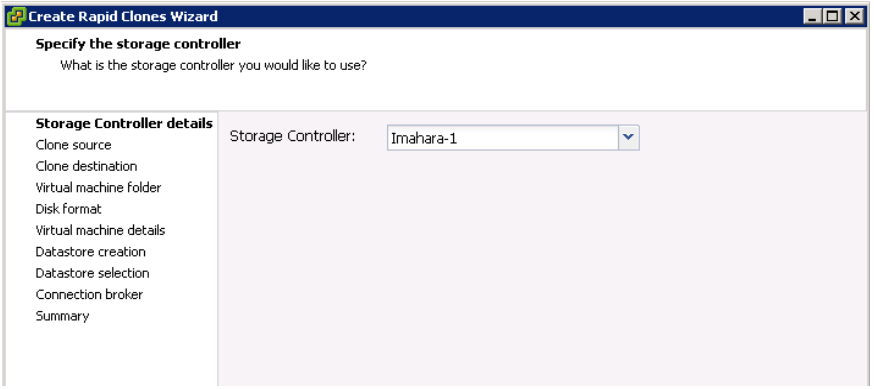

- "Cloning virtual machines" on page 12
- "Re-deploying clones" on page 23
- "Importing virtual machines into XenDesktop" on page 26

# Cloning virtual machines

1. In the vSphere Client Inventory, right-click a powered-down VM or template to clone, and select **NetApp > Create rapid clones…**.




2. The Create Rapid Clones Wizard launches. Select the storage controller to use for the new clones and click **Next**.



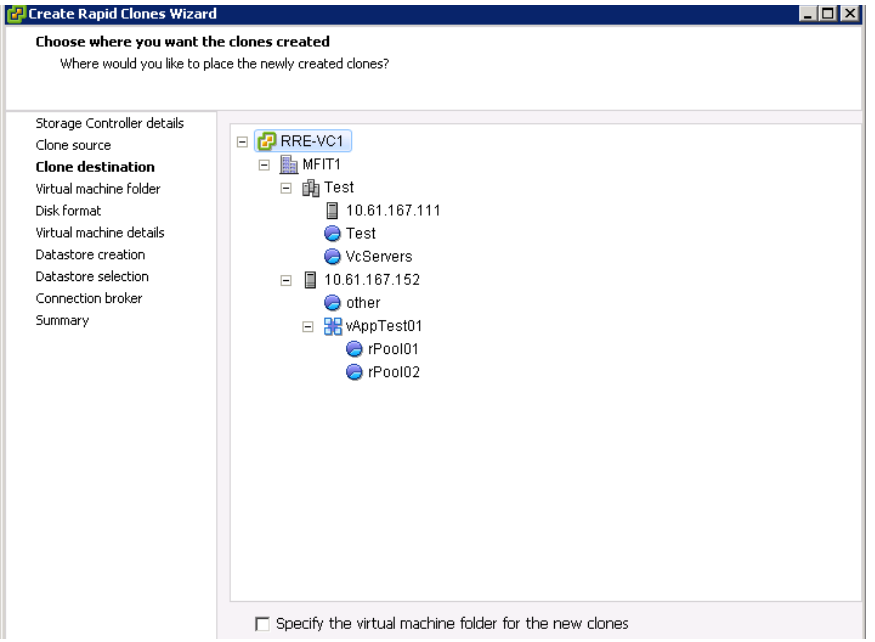
3. If the source VM or template has snapshots, the **Clone Source** window displays. Select the template or one of its snapshots as the source for the new clones and click **Next**

4. Select the destination for the new clones and click **Next**.

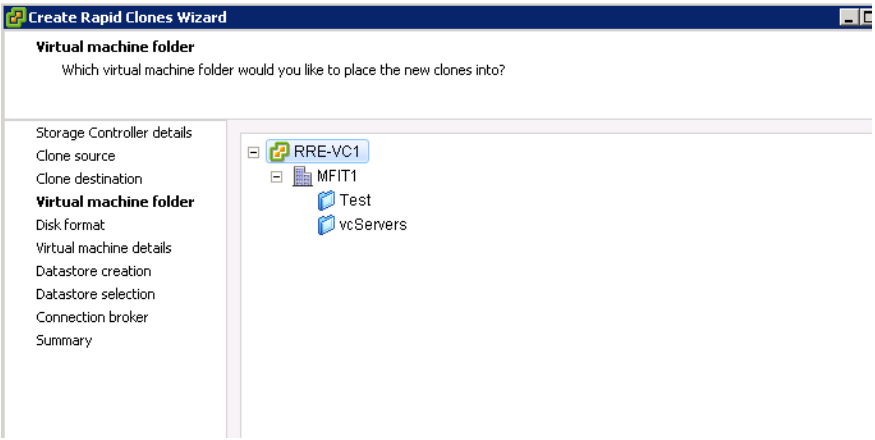
   If you want to specify the VM folder for the new clones, select the check box at the bottom of the window.

**Note**

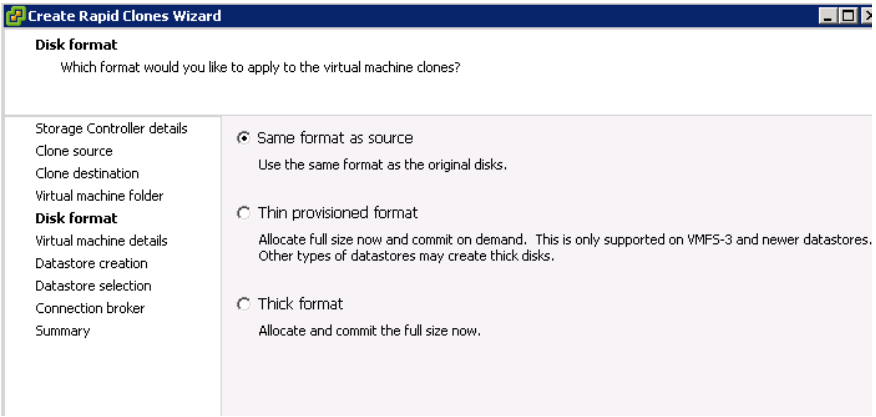
If the destination is a datacenter, cluster, vapp, or resource pool, new clones are distributed across available servers as quickly as possible. Therefore, new clones might be unevenly distributed, with more clones created on faster servers.



5. Identify the virtual machine folder in which to place the clones (if selected) and click **Next**.

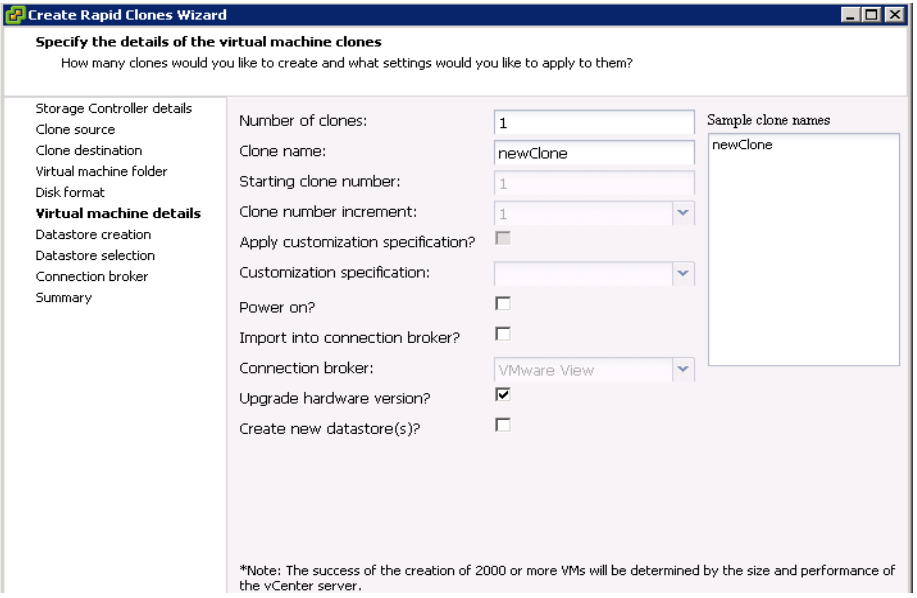



6. Select the disk format for the new clones and click **Next**.



7. Specify the following details for the clone operation, then click **Next**:
   - **Number of clones** - For new datastores, maximum 250. For existing datastores, maximum depends on available space. Number of clones must be evenly divisible by number of datastores being created.

     Success of 2,000 or more VMs depends on size and performance of the vCenter server.
   - **Clone name** - Used as a prefix for each clone. By default, the clone number is placed at the end of the clone name. To force clone number to a different position, use `%CLONE_NUM%` where you want the number to appear. For example, *new*`%CLONE_NUM%`*clone*.
   - **Starting clone number** - Maximum of eight digits.
   - **Clone increment** - Increment clone numbers by 1, 2, 3, 4 or 5.
   - **Apply customization specification?**- Applies a pre-defined specification to the new VMs. Select from specifications used for the native cloning process.
   - **Power on?** - Powers up the clones after creation.
   - **Import to connection broker?**- Automatically imports clone data into VMware View Server, or creates .csv file for Citrix XenDesktop.
   - **Connection broker** - Select the desired output type.
   - **Upgrade hardware version?** - If clone source was created on ESX 3.5 host and destination is a vSphere host, allows upgrade of clones to the new hardware version.
   - **Create new datastores?** - Creates new datastores for the virtual machine clones. Enabled for users with `create` role or higher.

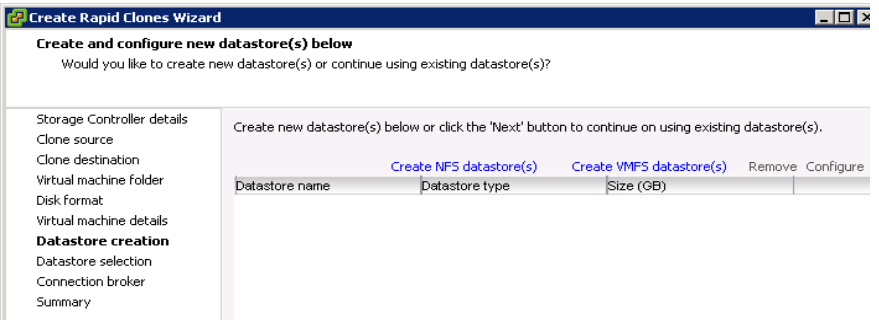


8. If you requested a new datastore, in the Datastore Creation window click one of the blue links to **Create NFS datastore(s)** or **Create VMFS datastore(s)**. To continue *without* creating a new datastore, click **Next**.

   Any restrictions for your configuration appear at the bottom of the window.

**Note**

You can create both NFS and VMFS datastores for the new VMs.




9. In the pop-up window, specify details for the new datastores and click **OK**.
   - **Protocol** (VMFS only) - FCP and iSCSI.
   - **Number of datastores** - Maximum 32 (NFS) or 256 (VMFS). The number of clones must be evenly divisible by the number of datastores.
   - **Datastore name** (disabled for multiple datastores) - Use the default or replace with a custom name.

**Note**

When creating multiple datastores, the golden volume name (NFS) or base name (VMFS) is used here and on the Summary tab as the datastore name.

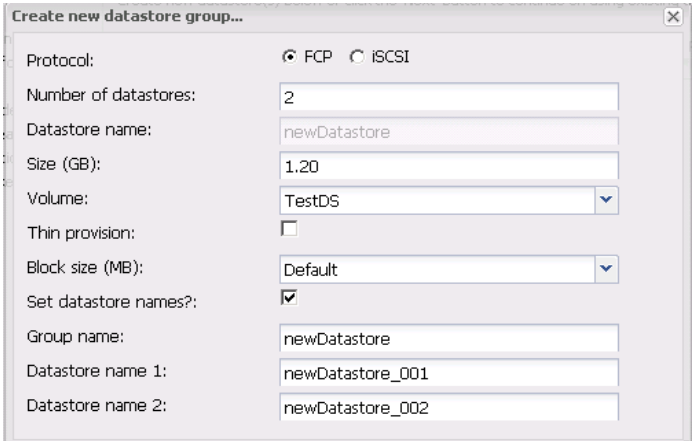
   - **Size** - Maximum datastore size depends on the controller and space available. For details, see the *Data ONTAP Storage Management Guide* for your Data ONTAP release.
   - **Volume** (VMFS only) - Select available volume from drop-down list.
   - **Aggregate** (NFS only) - Select available aggregate from drop-down list.
   - **Thin provision** - Capacity is not reserved for individual datastores. The aggregate is treated as a shared resource pool with capacity consumed as each datastore requires it. By eliminating unused but provisioned storage, more space is presented than is available. It is expected that all datastores will not utilize all of their provisioned storage at once.

**Note**

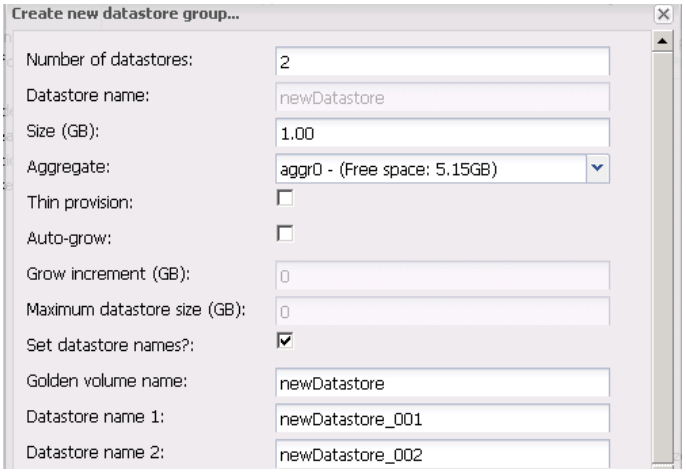
Thin provisioning sets space reserve to `none` and disables space checks. Therefore, cloning can fail if the size request uses too much of the aggregate.

- **Block size** (VMFS only) - Select from available block sizes.
- **Autogrow** (NFS only) - When space is needed, automatically expands the datastore by increment you specify, up to size limit you specify.
- **Grow increment** (NFS only) - Amount of storage added to datastore each time space is needed.
- **Maximum datastore size** (NFS only) - Limit at which Autogrow stops.
- **Set datastore names?** (enabled for multiple datastores) - Select to enable modification of default datastore names.
- **Group name** (VMFS) or **Golden volume name** (NFS) - Use the defaults or replace with custom names.

The illustration below shows the VMFS datastore creation pop-up window.



The illustration below shows the NFS datastore creation pop-up window.

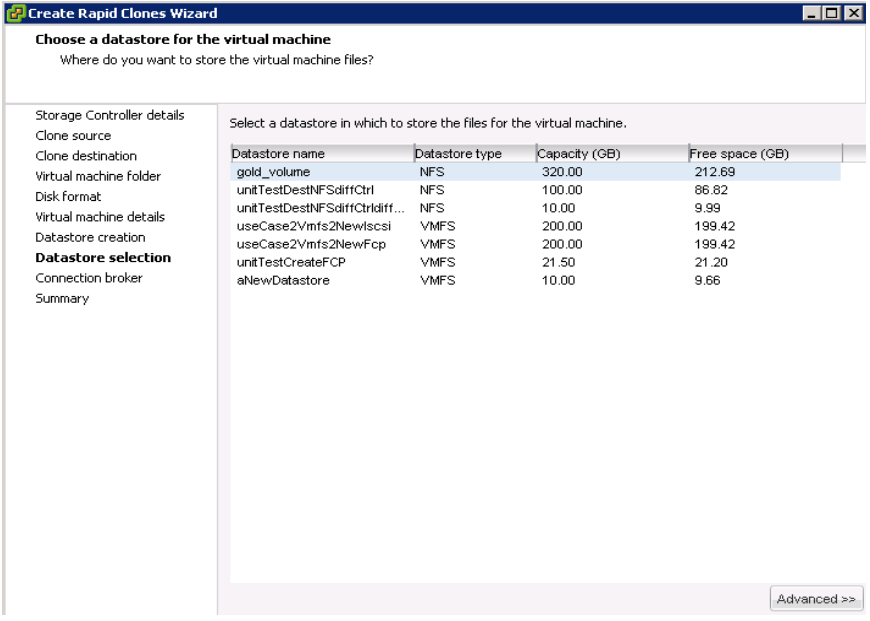



**10.** Select the datastore to house the new VMs.

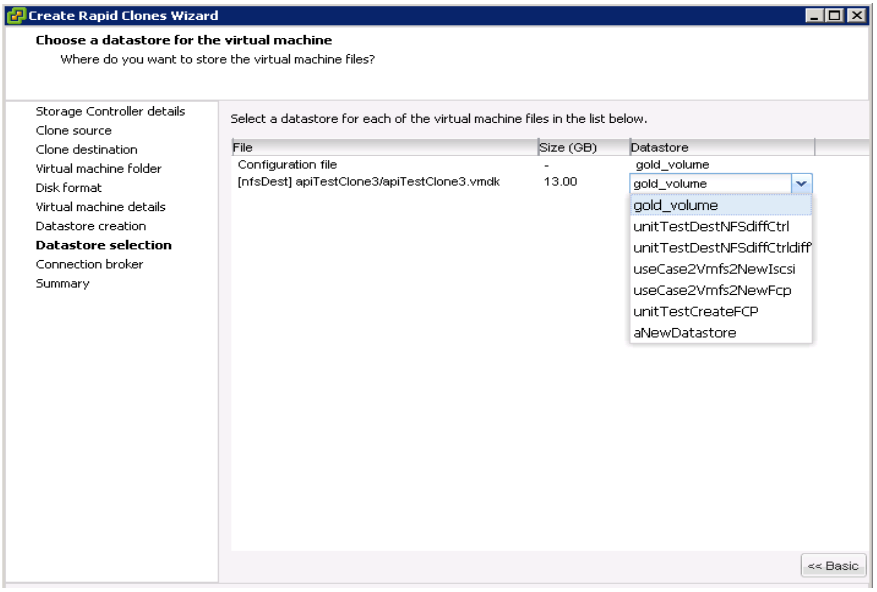
**Note**

You can clone the VMDK files that comprise the virtual machine to different datastores as described in Step b below.

**a.** To place all VMs in a single datastore, select the datastore and click **Next**.



**b.** To distribute VM files across multiple datastores, click **Advanced**. In the new window, select a VM file, open the corresponding Datastore pull-down list and select the datastore to house that file. Repeat for each VM file, then click **Next**.



11. If you specified a connection broker format, enter the details for the connection broker operation, then click **Next**.

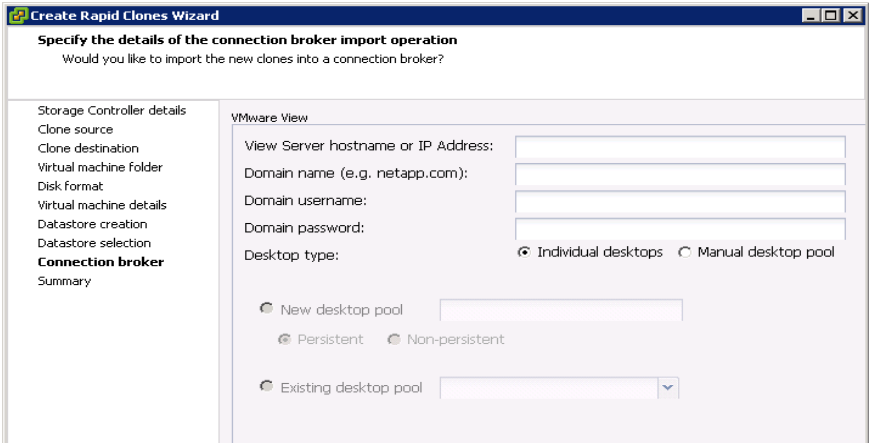
    a. For VMware View Server, RCU imports clone data into View Server at the end of the clone operation. You can specify unique desktop names and select from existing desktop pools.

       **Desktop type**: Choose **Individual desktops** to create new individual desktops. Select **Manual desktop pool** for a pool of virtual machines that are manually constructed by the View Manager administrator.

       **Access mode**: For manual desktop pools, **Persistent** access gives users connection to the same desktop for every session. **Non-persistent** access allocates desktops dynamically.

**Note**

Refer to the *View Manager Administration Guide* at http://www.vmware.com/pdf/viewmanager3_admin_guide.pdf for details on desktop types and access modes.

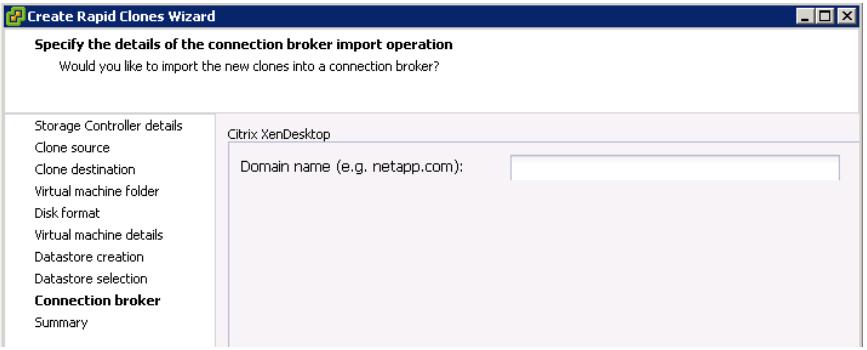


**b.** For Citrix XenDesktop, enter a fully qualified domain name. RCU creates a .csv file in the *install_directory*\exports directory with the following format:

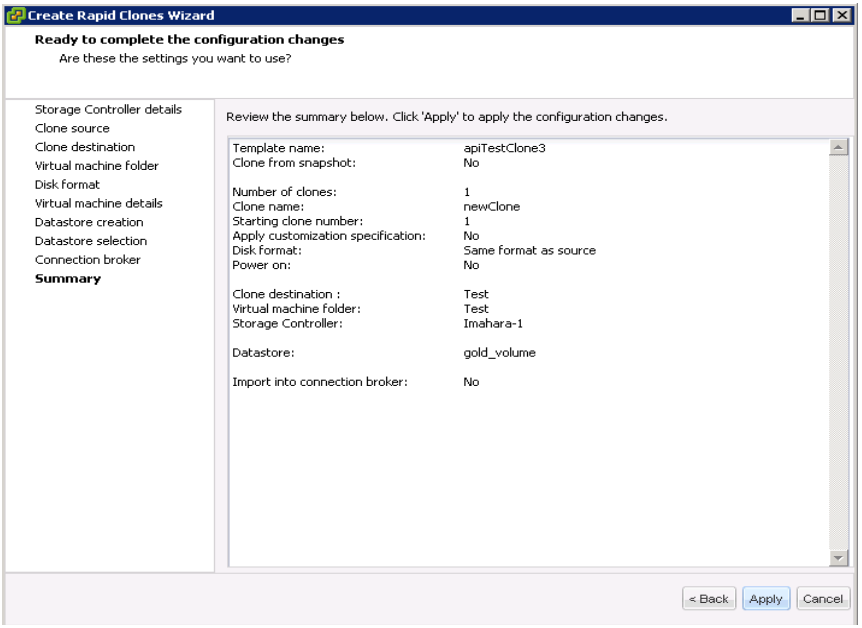
```
[ADComputerAccount],[AssignedUser],[VirtualMachine],[HostID]
```

**Note**

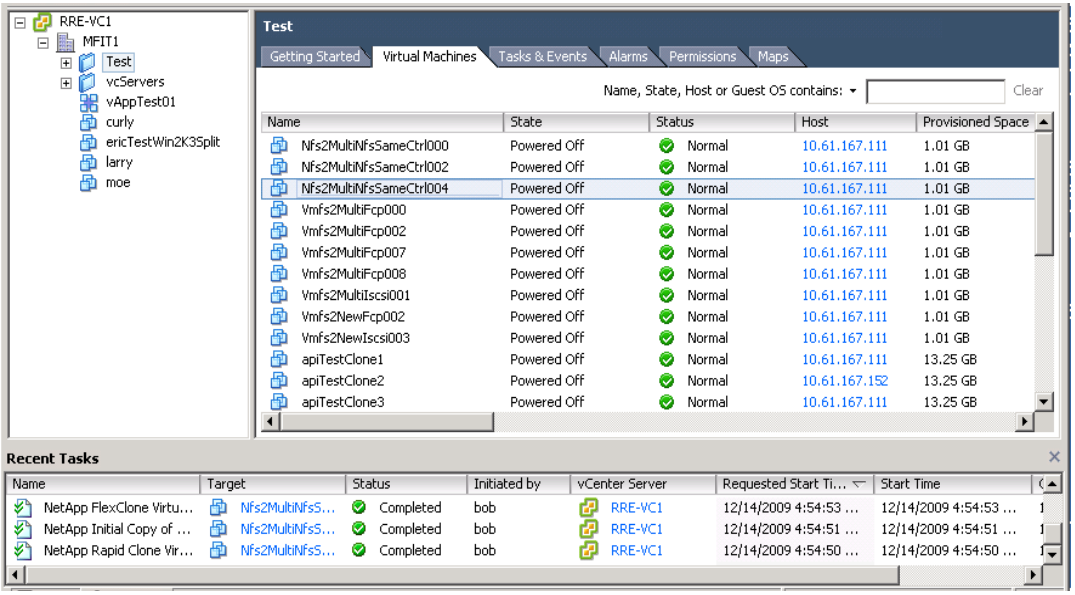
See "Importing virtual machines into XenDesktop" on page 26 for details on importing this file into Citrix XenDesktop.




**12.** Review the summary page and click **Apply** to proceed. To return to previous pages and modify settings, click **Back**.



13. The Recent Tasks pane of vSphere Client is populated as clone creation proceeds.




## Log of new VMs

A list of new VMs is written to a .csv file in the `<install_directory>\exports` directory. The filename is in the following format:

- `import_generic_date_time.csv`

# Re-deploying clones

The RCU re-deploy feature allows you to reset to their original state all VMs that are based on a selected gold VM or template. It also enables you to propagate changes made in the original gold VM or template to all of its clones. You can optionally reapply customization specifications as well.

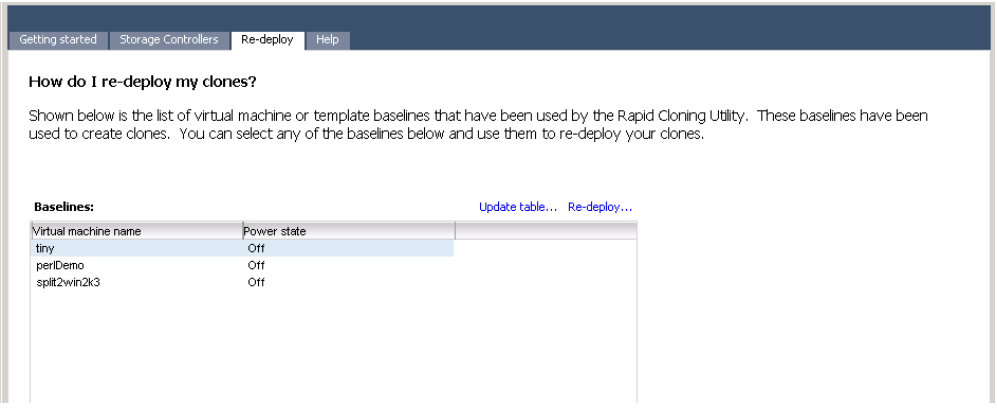
**Note**

This operation is available only for VMs that reside entirely on NFS destinations.

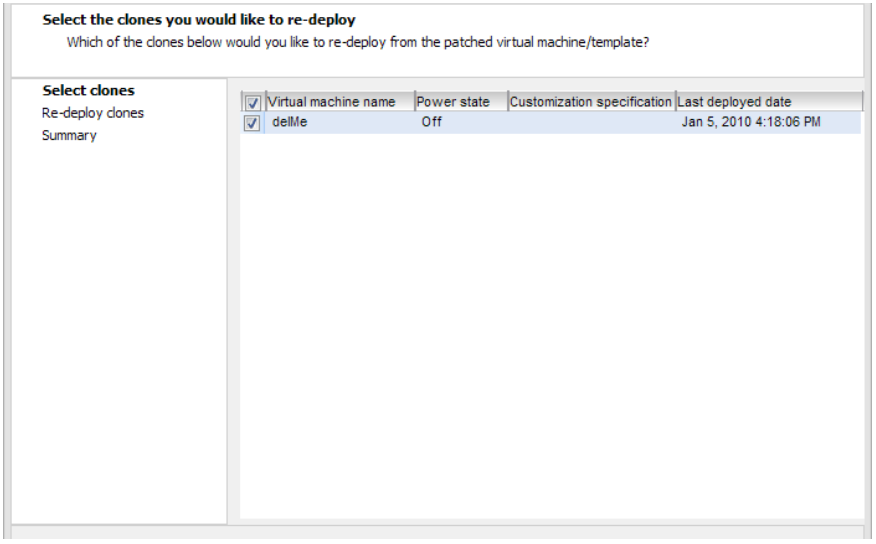
1. On the Rapid Cloning Utility **Re-deploy** tab, select the VM or template to use, then click the blue **Re-deploy**... link.

   **Note**

   Use the blue **Update table** link to refresh the list.



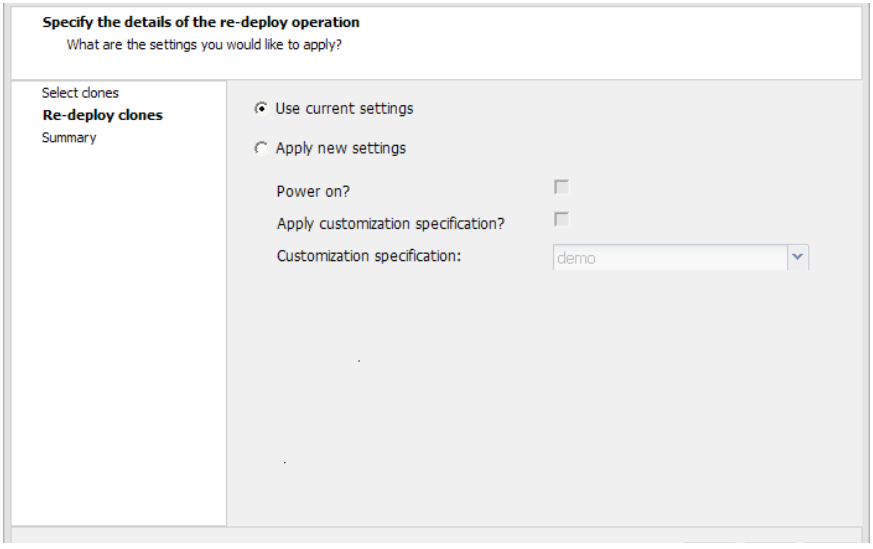
2. Select the check boxes for the clones to re-deploy, then click **Next**.



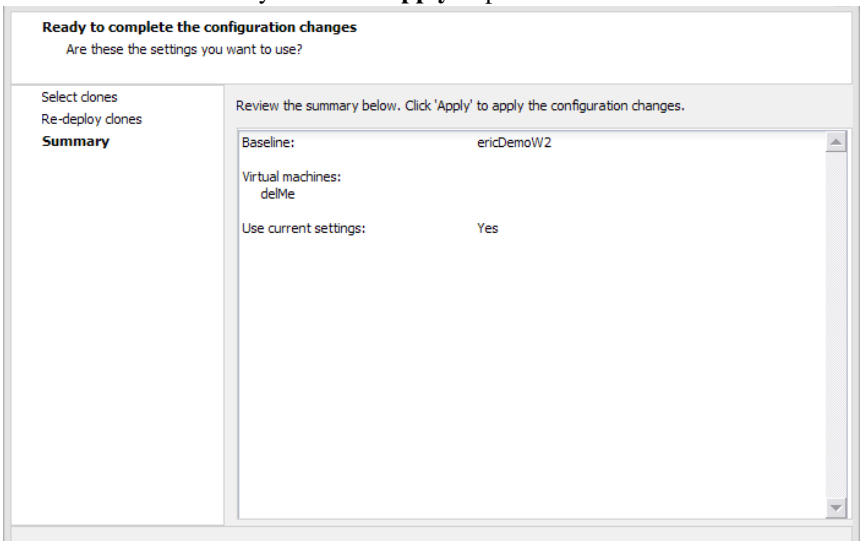

3. Specify the settings to apply to the re-deployed clones.

   a. **Power on?** - Powers on the clone.

   b. **Apply customization specification?** - Select from drop-down list.



4. Review the summary and click **Apply** to proceed.



# Importing virtual machines into XenDesktop

To create a Citrix XenDesktop import file, select **Citrix XenDesktop** as the Connection Broker in the Virtual Machine Details window of the Create Rapid Clones Wizard.

RCU creates the following XenDesktop import file:

- <*install_directory*>\exports\xenDesktop_*timestamp*.csv

The file contents are in the following format:

- [ADComputerAccount],[AssignedUser],[VirtualMachine],[HostID]

To import the file into XenDesktop:

1. Copy the export file to the XenDesktop system. Using the Citrix Access Management Console, choose to create a new desktop group.
2. A wizard is launched. Follow the wizard prompts to the fifth panel (Virtual Desktops page), and select the option to import the desktops from a .csv file.
3. Browse to the import file and click **OK**.

# Managing Storage Controllers and Datastores 3

**Topics in this chapter**

This chapter includes the following topics.

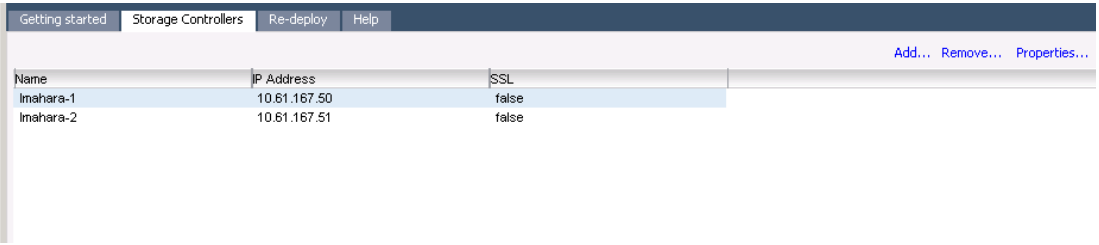

- "Managing storage controllers" on page 28
- "Managing datastores" on page 34

# Managing storage controllers

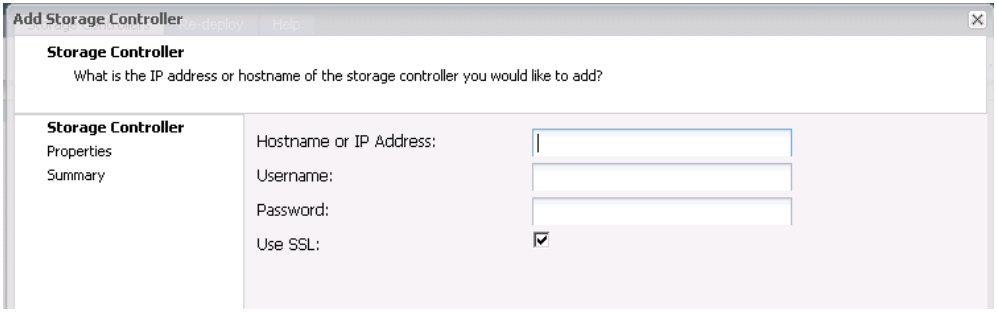
## Add storage controller

To add a storage controller to the list of controllers available to RCU:

1. Launch RCU in VMware vSphere Client.
2. On the **Storage Controllers** tab, click the blue **Add...** link.




3. Enter the hostname or IP address and credentials of the storage controller. Optionally deselect **Use SSL** if you do not want to encrypt communication between the RCU service and the storage controller. Click **OK**.



4. Specify interfaces, volumes, and aggregates to use on the storage controller.

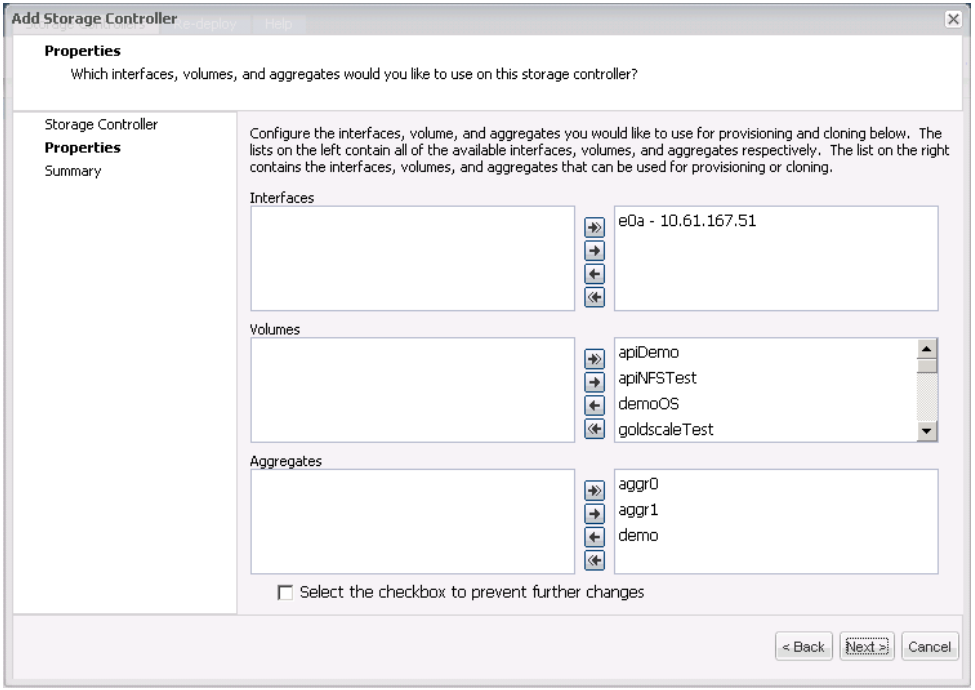
   a. Select components and click the directional buttons to move them to the desired column.

      - Items in the left column will *not* be used by RCU.
      - Items in the right column will be used by RCU.

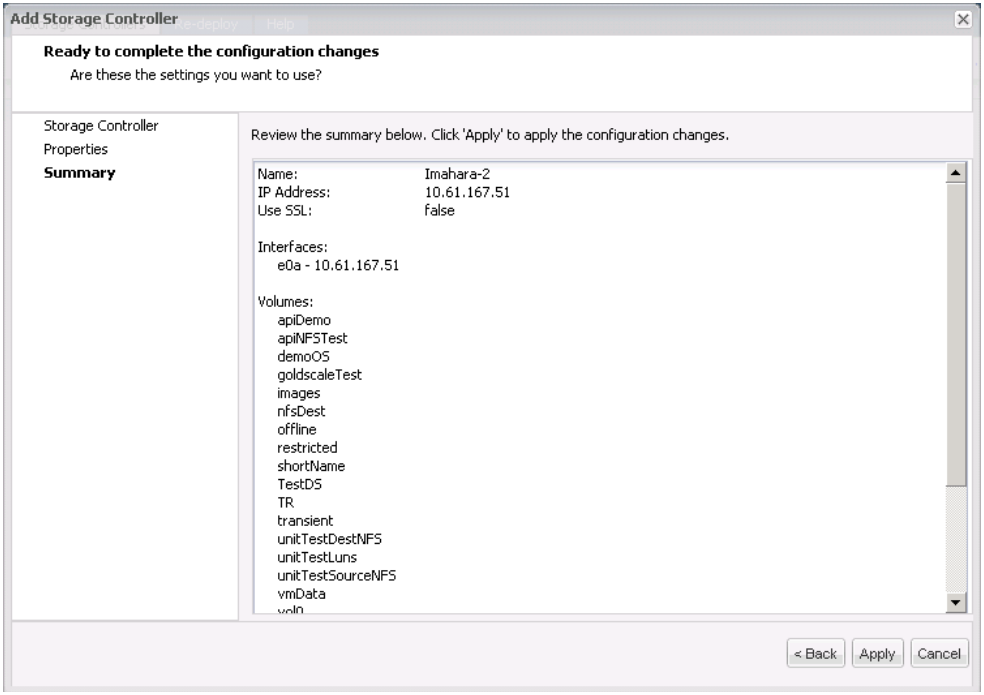
**Note**

New volumes and aggregates created outside RCU appear in the left column as unused.

   b. To prevent users from modifying these settings without providing storage system credentials, select the check box at the bottom of the windows.



5. Review the Summary and click **Apply** when finished.

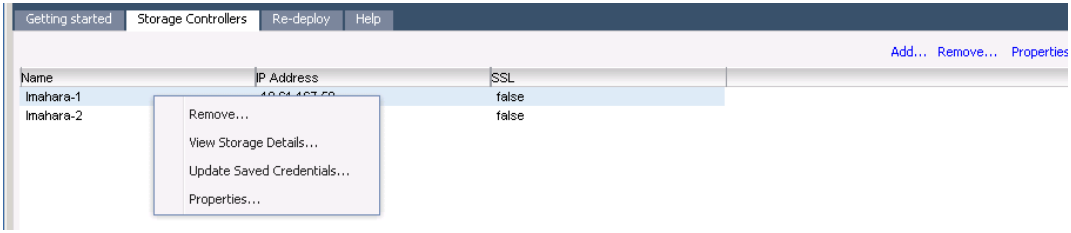



## Remove storage controller

If storage controller properties have been locked to prevent changes, the username and password will be required to remove the controller.

To remove a storage controller from the list of controllers available to RCU:

1. Launch RCU in VMware vSphere Client.
2. Right-click a storage controller and select **Remove**, or click the blue **Remove...** link

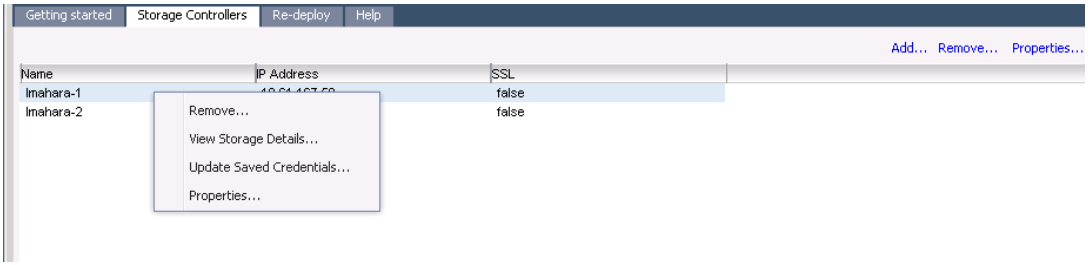



3. Click **Yes** to confirm removal of the storage controller from the list of controllers available to RCU.

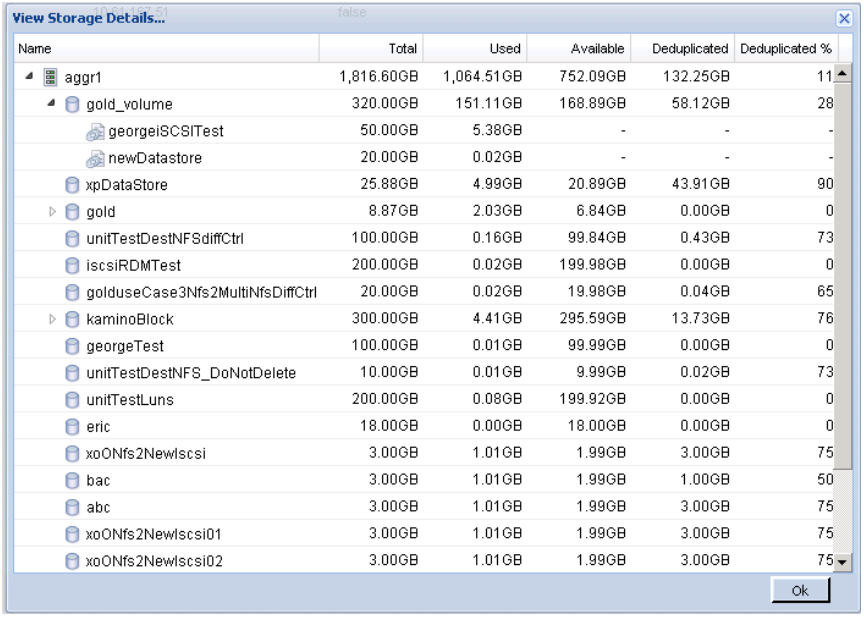
## View storage controller details

To view usage and deduplication statistics for a storage controller's aggregates, volumes, and LUNs:

1. Launch RCU in VMware vSphere Client.
2. Right-click a storage controller and select **View Storage Details**.




3. Click **OK** to close the window.
   - **Size** - Total storage on the controller.
   - **Used** - Storage used by RCU.
   - **Available** - Unused storage.
   - **Deduplicated** - Amount of total storage saved by deduplication.
   - **Deduplicated %** - Percent of total storage saved by deduplication.

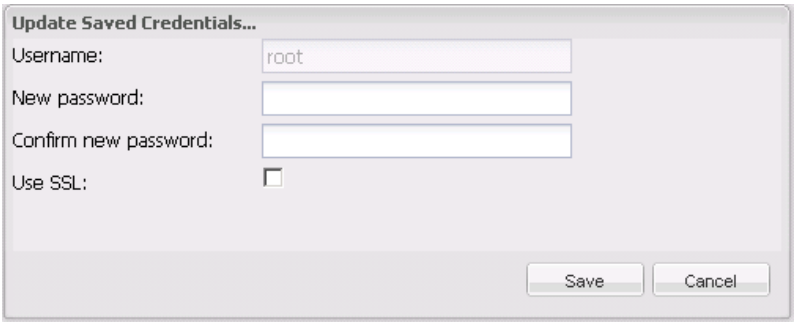


### Update saved credentials

If storage controller credentials are changed after the controller is added to RCU, use this feature to update the RCU settings to match the controller settings.

1. Launch RCU in VMware vSphere Client.
2. Right-click a storage controller and select **Update Saved Credentials**.

3. Enter the new credentials. Optionally select **Use SSL** if you want to encrypt communication between the RCU service and the storage controller. Click **Save** when finished.



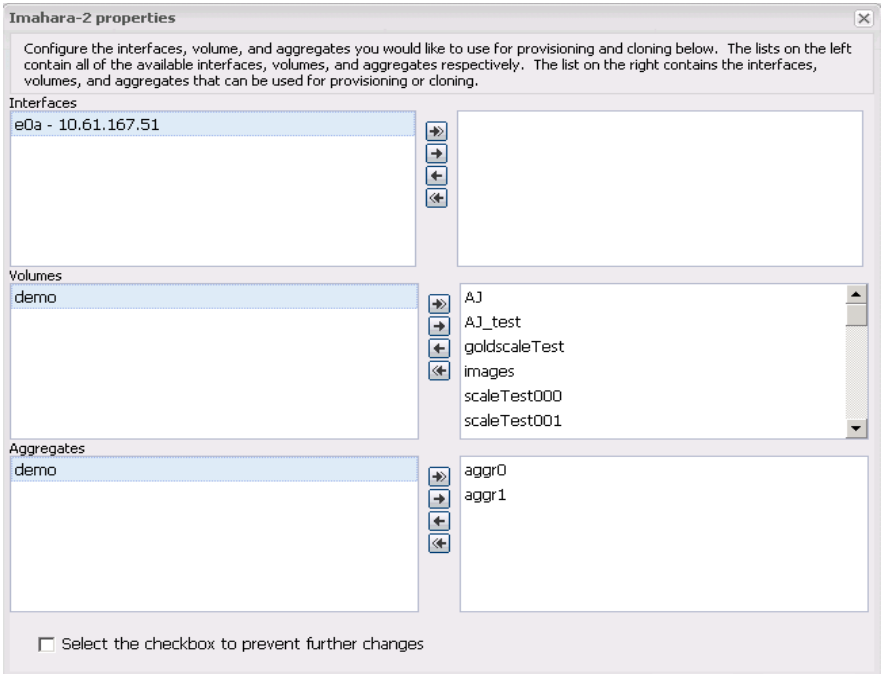
### Modify available interfaces, volumes, and aggregates

To restrict or expand the interfaces, volumes and aggregates available to RCU:

1. Launch RCU in VMware vSphere Client.
2. On the **Storage Controller** tab, right-click a storage controller and select **Properties.** Or click the blue **Properties...** link.
3. All available interfaces, volumes, and aggregates are presented.
   a. To restrict or expand availability to RCU, select items and click the directional buttons to move them to the desired column.
      - Items in the left column will *not* be used by RCU.
      - Items in the right column will be used by RCU.

   **Note**
   New volumes and aggregates created outside RCU appear in the left column as unused.

   b. To prevent users from modifying these settings without providing storage system credentials, select the check box at the bottom of the window.



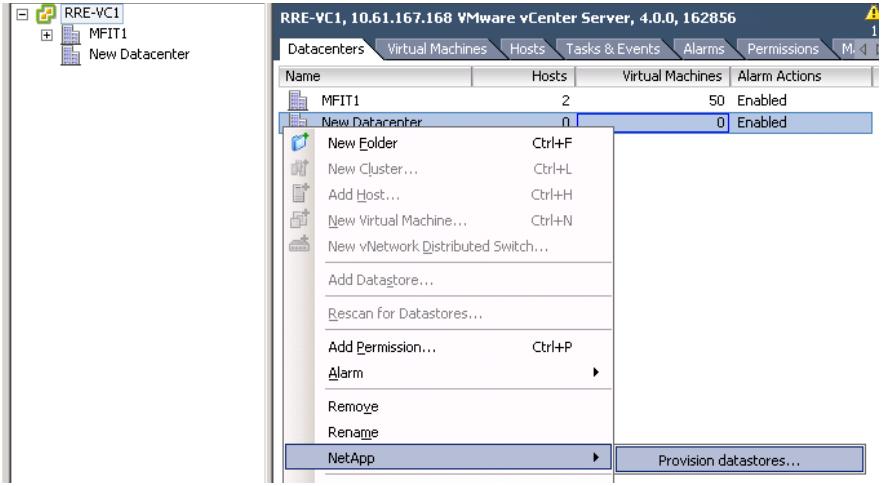

4. Click **Save** when finished.

# Managing datastores

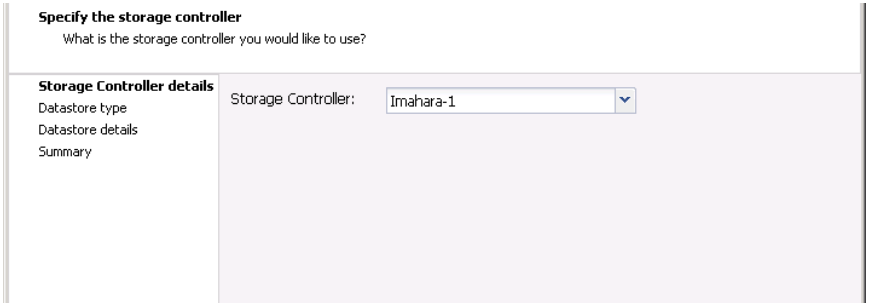
## Provision new datastore

You can create new datastores at the datacenter, cluster, or host level. The new datastore appears on every host in the datacenter or the cluster.

1. In vSphere Client **Inventory**, right-click a datacenter, cluster, or host and select **NetApp > Provision datastores**.




2. Select the storage controller to use for the new datacenter and click **Next**.

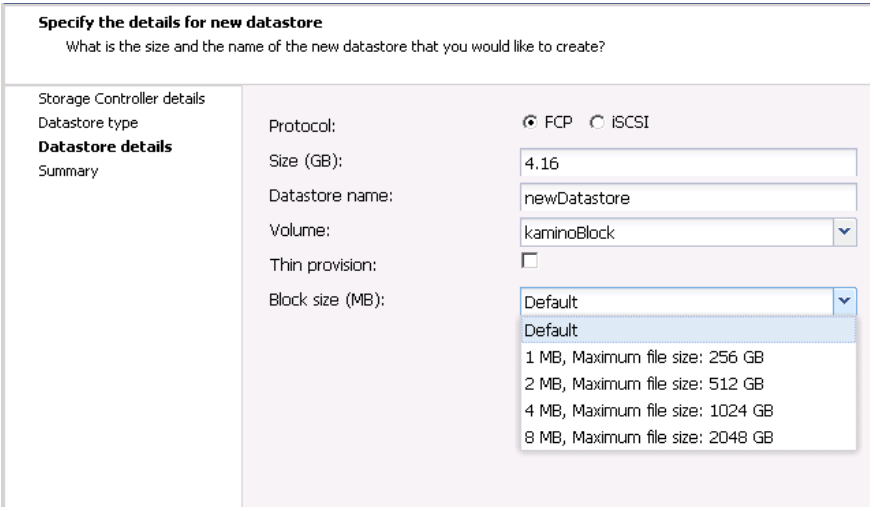


3. Select the desired datastore type and click **Next**.

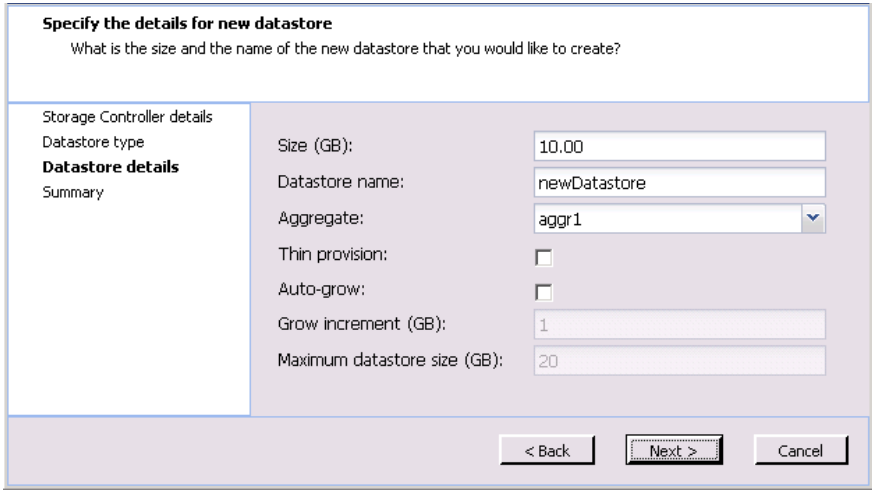
4. Specify details for the new datastore and click **Next**.
   - **Protocol** (VMFS only) - FCP or iSCSI
   - **Size** - Maximum datastore size depends on the controller and space available. For details, see the *Data ONTAP Storage Management Guide* for your Data ONTAP release.
   - **Datastore name** - Use the default or replace with a custom name.
   - **Volume** (VMFS only) - Select an available volume from drop-down list.
   - **Aggregate** (NFS only) - Select an available aggregate from drop-down list.
   - **Thin provision** - Capacity is not reserved for an individual datastore. The aggregate is treated as a shared resource pool, where capacity is consumed as each datastore requires it. By eliminating unused but provisioned areas of storage, more space is presented than is actually available. It is expected that all datastores will not utilize all of their provisioned storage at the same time.

     **Note**: Thin provisioning sets space reserve to `none`, and disables space checks. Cloning can fail if your size request uses too much of the aggregate.
   - **Block size** (VMFS only) - Select an available block size from drop-down list.
   - **Autogrow** (NFS only) - If more space is required, automatically expands the datastore by the increment you specify, up to the size limit you specify.
   - **Grow increment** (NFS only) - Amount of storage added to datastore each time space is needed.
   - **Maximum datastore size** (NFS only) - Limit at which Autogrow stops.

The illustration below shows the VMFS datastore details window.




The illustration below shows the NFS datastore details window.

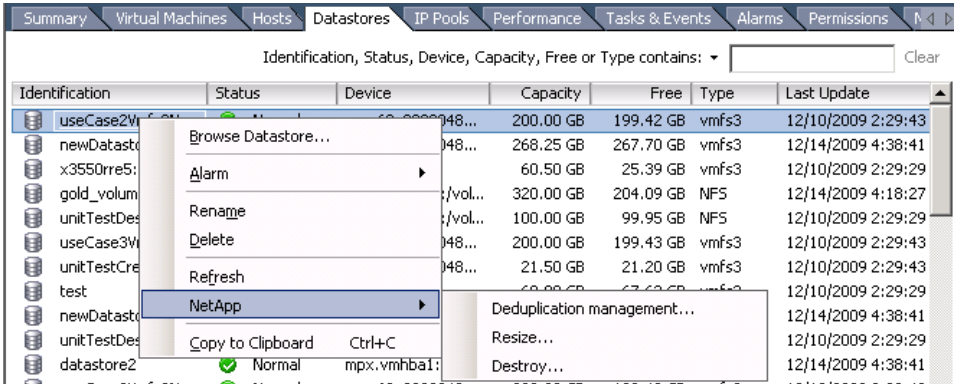



5. Review the datastore configuration summary and click **Apply** to begin provisioning the new datastore.

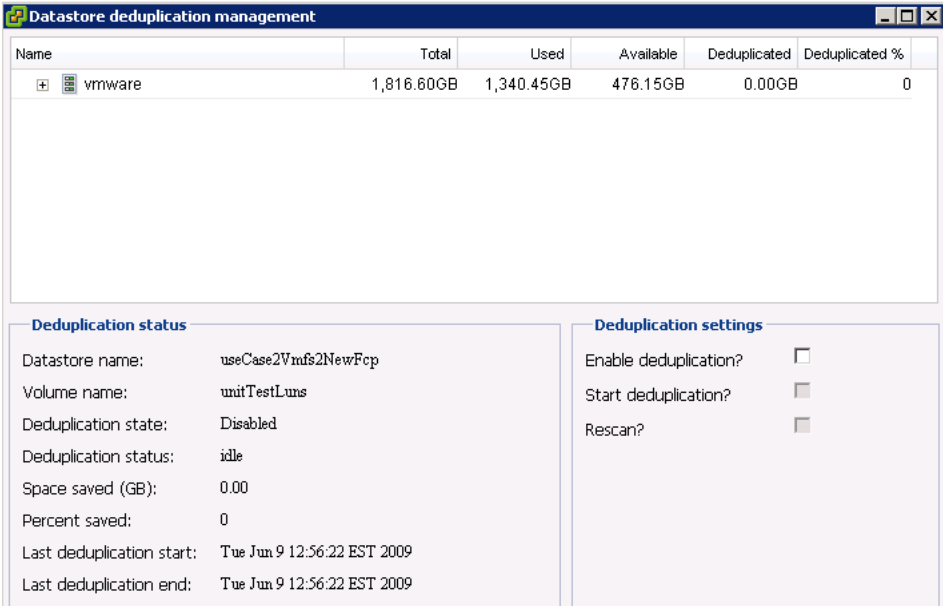
## Manage deduplication

Deduplication eliminates redundant objects on a selected datastore and references only the original object. For details on deduplication, refer to your *Data ONTAP Data Protection Online Backup and Recovery Guide*.

1. In the vSphere Client **Inventory**, select a host. On the **Summary** tab, right-click a datastore and select **NetApp > Deduplication management**.



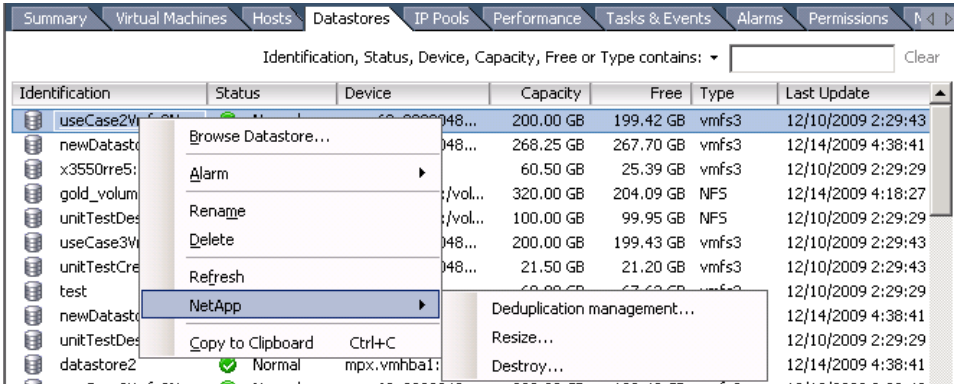

2. Select check boxes to **Enable/Disable** and **Start** deduplication. The **Deduplicated** column presents previously used space saved by this feature.
   - **Enable/Disable** toggles the deduplication feature on or off.
   - **Start** begins deduplication from the last marker position.
   - **Rescan** begins deduplication at the beginning of the volume.



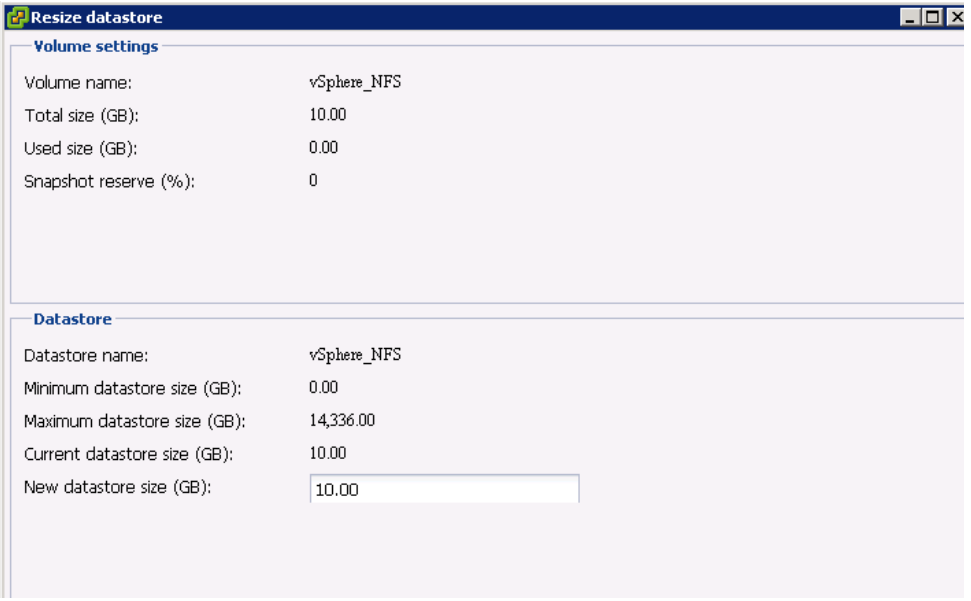
## Resize datastore

NFS datastore sizes can be increased or decreased. VMFS datastore sizes can be increased but *not* decreased. To resize a datastore:

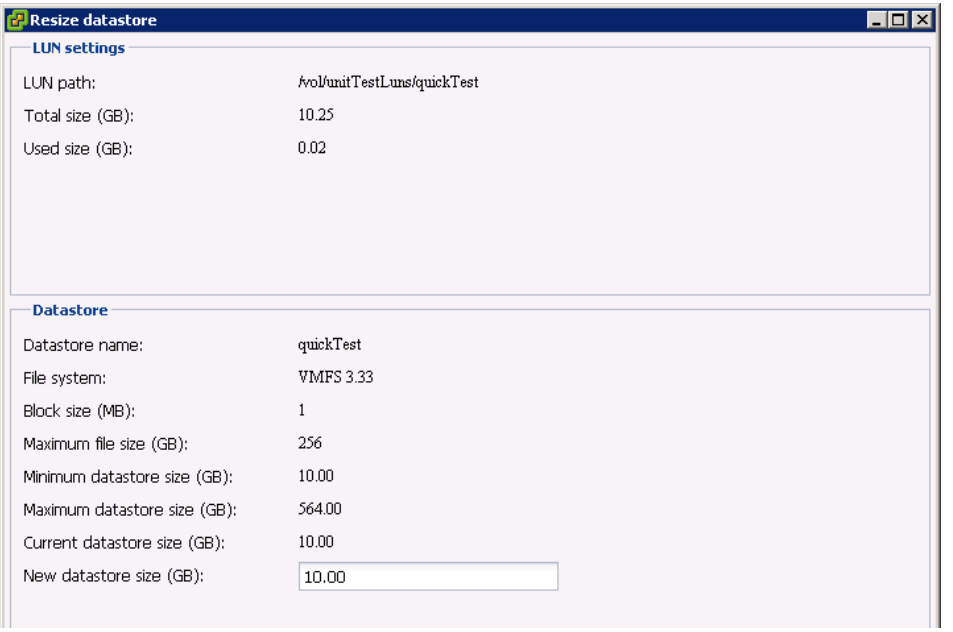
1. In the vSphere Client **Inventory**, right-click a datastore and select **NetApp** > **Resize**.




2. Enter the new datastore size and click **OK**. Then click **Yes** to confirm the operation. The illustration below shows an NFS datastore resize.



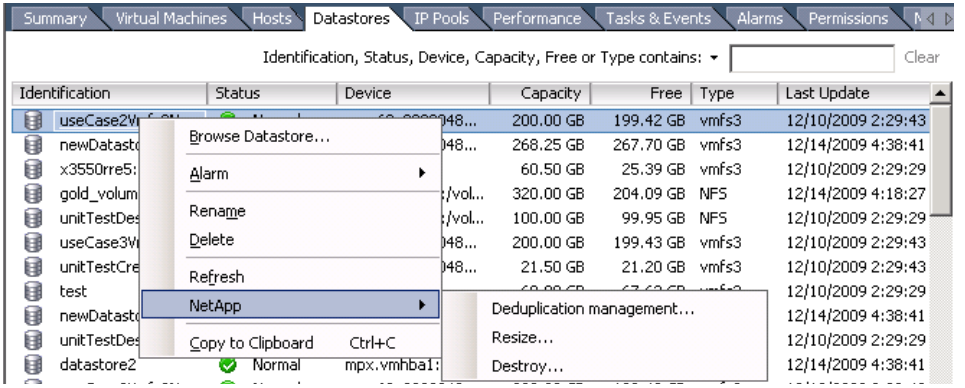
The illustration below shows a VMFS datastore resize.



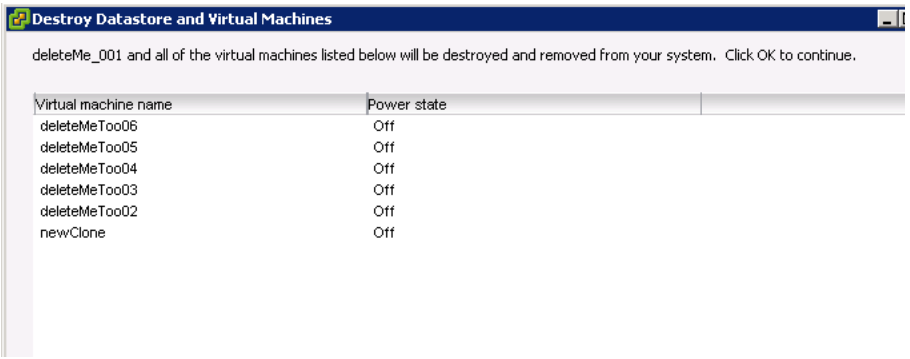

## Destroy datastore

The RCU can quickly destroy all virtual machines in a datastore, unregister and detach the datastore from the vSphere Client environment, and free the space on the storage controller. These actions permanently destroy all VMs, datastores, snapshots and FlexVolumes. To destroy a group of VMs:

1. In the vSphere Client **Inventory,** right-click a datastore and select **NetApp** > **Destroy**.



2. Click **OK** to proceed.




**Multiple datastore note (NFS only):** When a request is made to create more than one new NFS datastore, the RCU first creates a *golden volume* on the controller. The datastores that are then attached to vSphere Client will be FlexClones of the golden volume. Because the FlexClones share the storage with the golden volume, this space is not wasted. When the last datastore (FlexClone) of a golden volume is destroyed as described above, the golden volume is destroyed as well. This gives you complete control over each individual datastore.

# Troubleshooting 4

**Topics in this chapter**

This chapter includes the following topics.


- "Troubleshooting" on page 42

# Troubleshooting

For a list of known issues and resolutions, see the latest *Rapid Cloning Utility Release Notes* at http://now.netapp.com.

If you have a question that is not resolved by the *Release Notes*, contact NetApp Technical Support.

## Things to check before you call Technical Support

Before calling Technical Support, there are several things you can try in effort to solve the problem yourself. These include:

- Review the known issues and resolutions in the latest *Rapid Cloning Utility Release Notes* at http://now.netapp.com to see if they contain information on your problem.
- Review this guide to make sure you are performing the steps correctly.
- Review the System Requirements in the *Release Notes* to verify that your environment is in compliance with NetApp requirements.
- Check all cables to make sure that they are connected properly.

## Information you need to provide to Technical Support

Before you contact Technical Support, you need to gather the following information about your system and your problem:

- A valid NetApp storage controller serial number that is licensed and under warranty.
- Your contact information.
- A list of all the products you are using including software versions and licenses.
- All the error messages from the system.
- Information on your VMware environment.

## How to contact NetApp Technical Support

You can contact NetApp Technical Support by telephone or from the NetApp NOW site.

To contact Technical Support by telephone, use the following numbers:

- 888 4-NETAPP (US and Canada)
- 00.800.44.NETAPP (EMEA/Europe)
- +800.800.80.800 (Asia/Pacific)

To contact Technical Support from the Web, use the following address:

- http://now.netapp.com

For a detailed listing of Technical Support phone numbers, go to the NetApp Support page at

http://www.netapp.com/us/support/ngs-contacts.html

# Storage system permissions A

**Topics in this appendix**

This Appendix presents the storage system permissions required by the Rapid Cloning Utility. It includes the following topics.


- "Create clones role" on page 46
- "Create storage role" on page 48
- "Modify storage role" on page 49
- "Destroy storage role" on page 50

# Create clones role

A role and group must be created or modified as described below. Then a user must be created or modified to include this group. Finally, when the storage system is added to the Rapid Cloning Utility, the user just created or modified should be used.

The following permissions are required to use File Level FlexClone (flfc) to clone vmdk files in the same datastore while creating VMs.

- `api-system-get-version`
- `login-http-admin`
- `api-system-get-info`
- `api-system-cli`
- `api-license-list-info`
- `cli-ifconfig`
- `api-aggr-list-info`
- `api-volume-list-info`
- `api-lun-list-info`
- `api-lun-map-list-info`
- `api-igroup-list-info`
- `api-ems-autosupport-log`
- `api-file-get-file-info`
- `api-clone-*`
- `api-file-create-directory`
- `api-file-read-file`
- `api-file-delete-file`
- `api-file-write-file`
- `cli-mv`
- `api-file-delete-directory`
- `cli-ndmpd`
- `cli-ndmpcopy`
- `api-useradmin-user-list`
- `api-cf-status`
- `api-snapshot-list-info`
- `api-volume-autosize-get`
- `api-iscsi-session-list-info`
- `api-iscsi-portal-list-info`
- `api-fcp-service-status`
- `api-iscsi-service-status`
- `cli-df`
- `api-snapmirror-get-volume-status`
- `api-quota-report`

- `api-qtree-list`
- `api-system-api-list`
- `api-vfiler-list-info`

# Create storage role

The following access is required to create volumes and mount a datastore while creating VMs (includes previous rights).

- `api-volume-create`
- `api-volume-set-option`
- `api-volume-autosize-set`
- `api-sis-enable`
- `api-sis-start`
- `api-snapshot-create`
- `api-snapshot-set-reserve`
- `api-volume-clone-create`
- `api-nfs-exportfs-list-rules-2`
- `api-nfs-exportfs-modify-rule-2`
- `api-nfs-exportfs-load-exports`
- `api-igroup-create`
- `api-lun-create-by-size`
- `api-lun-map`
- `api-lun-set-comment`
- `api-igroup-add`
- `cli-qtree`
- `cli-iscsi`

# Modify storage role

The following access is required to manipulate storage objects (resize volume, dedupe volume) (includes previous rights).

- `api-volume-size`
- `api-sis-disable`
- `api-sis-stop`
- `api-lun-resize`

# Destroy storage role

The following access is required to destroy a datastore (volume/flexclone) (includes previous rights).

- `api-volume-offline`
- `api-volume-destroy`
- `api-lun-offline`
- `api-lun-destroy`

# Preferences File, Logs, and Export Files B

**Topics in this appendix**

This Appendix includes the following topics.


- "Preferences File" on page 52
- "Logs" on page 53
- "Export Files" on page 54

# Preferences File

The RCU preferences file contains options that control RCU operation. This file is stored in the following location:

- *install_directory*/conf/rcuPreferences.xml

The preferences file contains the following options:

| Option | Default Setting | Default Action |
|---|---|---|
| snapreserve | 0 | Sets the snapshot reserve size for new volumes to the specified percentage. |
| minra | off | Controls how many blocks are pre-fetched at each read operation. |
| no_atime_update | on | Disables updating of access time when files are read. Leave on for better performance in a heavy read traffic environment. |
| nosnap | on | Disables snapshots for new flexclone volumes. |
| sis | true | Enables deduplication on new volumes. |
| destroyParentIfLastClone | true | When last flexclone is destroyed, the golden volume is destroyed. |
| checkPossibleCrossMount | true | If an administrator mounts a volume for use as a datastore to an ESX host using a different ip address or volume path than what was used to mount the volume to other ESX hosts, there is a potential for the RCU Destroy feature to incorrectly report which VMs are present on the datastore. This is because the datastore uuid is generated by the ESX host based on the ip address and volume path. checkPossibleCrossMount addresses this issue and therefore should be left set to true. The only time this should be set to false is if the Destroy window stays in the "Loading..." state for an extended period of time and there is no chance of the issue described above. |
| onlyOffline | false | By default, the Destroy feature takes offline and destroys the volume that was mounted as a datastore. Setting onlyOffline to true leaves the volume intact, just offline. |

# Logs

The RCU creates the following log files.

- *install_directory*/logs/rcu.log
- *install_directory*/logs/rcuSupport.xml

The log files include RCU version, build number, and build date. Each log file is rotated at a size of 8192 KB. Up to nine log files are retained.

# Export Files

### Network configuration file

After each cloning procedure, RCU creates a .csv network configuration file in the *install_directory*\exports directory. The filename is in the following format:

- import_generic_*date_time*.csv

For example: import_generic_2009_03_06_11_04.csv

The file includes the following information for each VM:

- VM Name
- UUID, VMX Path
- Number of CPUs
- Memory in MB
- Guest Full Name
- Guest Alternate Name
- Datastore Name

The file also includes the following information for up to ten NICs:

- NIC x
- NIC x Network Label
- NIC x Address Type
- Manual - Statically assigned MAC address.
- Generated - Automatically generated MAC address.
- Assigned - MAC address assigned by VirtualCenter.
- NIC x WOL Enabled
- Wake-on-LAN enabled or disabled on this virtual network adapter
- NIC x MAC Address

### XenDesktop

When XenDesktop connection broker output is requested in the cloning wizard, RCU creates a .csv file in the *install_directory*\exports directory. The filename is in the following format:

- xenDesktop_*timestamp*.csv

The file includes the following information:

[ADComputerAccount],[AssignedUser],[VirtualMachine],[HostID]

# Programmable API

# C

**Topics in this appendix**

This Appendix describes the RCU programmable API. It includes the following topics.


- "Overview" on page 56
- "Methods" on page 58
- "Specifications and messages" on page 77
- "Example" on page 92
- "Java" on page 94

The RCU Application Programming Interface (API) is accessed using Simple Object Access Protocol (SOAP). The API is a layer over the NetApp API (called ZAPI) and the VMware VI SDK, but does not require either in the consumer application or script. The RCU API is designed to be leveraged with the VI SDK to offload the intricacies of storage object cloning while cloning virtual machines. To this end, the managed object reference returned by the VMware VI SDK is used to identify components in the vCenter Inventory. You can view this information using the Managed Object Browser on the vCenter server.

This version of the API exposes the clone creation engine (which includes the new re-deploy feature), the datastore management engine (create, resize, destroy), and the file copy/clone offload engine. There are also two general purpose utility methods included:

- `getVmFiles` returns a list of files that make up the virtual machine. This is useful for creating the list of files required in the `cloneSpec` described below.
- `getMoref` returns the managed object reference of the requested object based on name and type. The `getMoref` returns the first object that matches the name and type combination. For this reason, this method should not be used in production environments unless all object names are unique.

### The virtual machine clone engine

The virtual machine clone engine provides two methods:

- `createClones` can be used to create VM clones on new or existing datastores. In the case that more than one datastore is created, the FlexClone feature on the controller is leveraged to create clones of the datastore.
- `redeployVMs` provides the ability to re-deploy the virtual hard drives of the source virtual machine to the virtual machines specified. This feature leverages the FlexClone feature on the controller as well.

### The datastore management engine

The datastore management engine provides three methods:

- The `createDatastore` method provides the ability to provision storage on the controller, present it to the ESX host(s), and create a datastore.
- The `resizeDatastore` method provides the ability to grow and shrink NFS based datastores and grow VMFS based datastores.
- The `destroyDatastore` method provides the ability to delete all virtual machines associated with the datastore, unmount it from ESX host(s) and destroy the storage objects on the controller (freeing the space).

### The file copy/clone offload engine

The file copy/clone offload engine provides four methods. These methods provide the ability to execute and monitor file copy and clone operations. This engine provides the ability to offload file copy and clone operations to the controller for NFS-based datastores. This functionality is unique, when compared to that provided by the other engines, in that it does not require a Virtual Center session. An ESX host session can be used instead.

The input to the methods are a combination of complex (specification and message) and simple (string, int, long, boolean, etc.) datatypes. The specifications and messages are described below.

Very little verification or validation is done in the API. For example, if the username and password combination is not correct, the API will fail. If there is not enough space to create the requested datastore(s), the API method will fail.

### NetApp Communities

General questions related to the Rapid Cloning Utility API can be submitted to the NetApp Communities Forum located at http://communities.netapp.com/community/products_and_solutions/virtualization.

# Methods

This section contains the following topics:


- "Virtual machine clone creation and re-deploy engine" on page 58
- "Datastore management engine" on page 61
- "Copy/Clone offload engine" on page 65
- "Utility methods" on page 72


# Virtual machine clone creation and re-deploy engine

### createClones

Create VM clones on new or existing datastores. The source can be a virtual machine or a virtual machine template. The source can be further refined by specifying a virtual machine snapshot.

The following options cause at least one native clone (built into Virtual Center) operation to occur:

- clone source is powered on
- virtual machine snapshot is specified
- hard drive transformation is specified

The virtual or template must not contain any RDMs, must not contain any devices that use VMDirectPath, and must be in a good state (connected). The mix of VirtualIDEController attached hard drives and VirtualSCSIController hard drives in the same VM may result in the drives being reordered in the resulting clones, therefore this is not supported. The creation of virtual machines based on hardware version `vmx-07` will fail on ESX 3.5 hosts.

**Status:** Current (added in version 2.1)

**Type:** Asynchronous

**Parameters:**

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| `requestSpec` | RequestSpec | See "Request Spec" on page 78 | Request can specify a vCenter server only. This method does not support direct connections to ESX hosts. |

- XML

```
complexType name="createClones">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="arg0"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

**Return:**

- Description

| Type | Example | Description |
|---|---|---|
| String | `Task:task-2` | A managed object reference to a vCenter task. This task can be monitored and altered using the VI SDK |

- XML

```
<complexType name="createClonesResponse">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="return"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

## Re-deploy VMs

Re-deploy the virtual hard drives associated with a virtual machine to other virtual machines. The source can be a virtual machine, a virtual machine template, or a virtual machine snapshot.

The following options cause a native (built into Virtual Center) clone operation before it can use the Rapid Clone methodology:

- clone source is powered on
- virtual machine snapshot is specified
- hard drive transformation is specified

The virtual machine or template must not contain any RDMs or any devices that use VMDirectPath, and must be in a good state (connected).

**Status:** Current (added in version 3.0)

**Type:** Asynchronous

### Parameters:

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| requestSpec | RequestSpec | See "RequestSpec" on page 78 | Request can specify a vCenter server only. This method does not support direct connections to ESX hosts. |

- XML

```
<complexType name="redeployVMs">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="arg0"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
        <element name="arg1"
type="{http://server.kamino.netapp.com/}controllerSpec"
maxOccurs="unbounded" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

### Return:

- Description

| Type | Example | Description |
|---|---|---|
| String | `Task:task-2` | Managed object reference to a vCenter task. Task can be monitored and altered using the VI SDK. |

- XML

```
<complexType name="redeployVMsResponse">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
```

```
        <sequence>
          <element name="return"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

# Datastore management engine

### createDatastore

Provision storage on the controller, attach it the ESX host(s) and create a datastore. More than one ESX host can be chosen by specifying the managed object reference of a cluster or datacenter in the DatastoreSpec.

**Status:** Current (added in version 3.0)

**Type:** Synchronous

**Parameters:**

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| dsSpec | DatastoreSpec | See "DatastoreSpec" on page 83 | The specification describing the datastore to create. |
| requestSpec | RequestSpec | See "RequestSpec" on page 78 | Request can specify a vCenter server only. This method does not support direct connections to ESX hosts. |

- XML

```
  <complexType name="createDatastore">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="arg0"
type="{http://server.kamino.netapp.com/}datastoreSpec"
minOccurs="0"/>
```

```
          <element name="arg1"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

**Return:**

- Description

| Type | Example | Description |
|---|---|---|
| String | `newDatastore` | The name of the new datastore that as created. |

- XML

```
  <complexType name="createDatastoreResponse">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="return"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

### resizeDatastore

Grow or shrink NFS-based datastores (and associated storage object on the controller) and grow VMFS-based datastores (and associated storage object on the controller).

**Status:** Current (added in version 3.0)

**Type:** Synchronous

**Parameters:**

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| dsSpec | DatastoreSpec | See “DatastoreSpec” on page 83 | Specification describing datastore resize request. |

| Name | Type | Example | Description |
|---|---|---|---|
| requestSpec | RequestSpec | See “RequestSpec” on page 78 | Request can specify a vCenter server only. This method does not support direct connections to ESX hosts. |

XML

```
 <complexType name="resizeDatastore">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="arg0"
type="{http://server.kamino.netapp.com/}datastoreSpec"
minOccurs="0"/>
         <element name="arg1"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

### Return:

- Description

| Type | Example | Description |
|---|---|---|
| String | `Task:task-2` | A managed object reference to a vCenter task. |

- XML

```
 <complexType name="resizeDatastoreResponse">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="return"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

### destroyDatastore

Delete any VM with a file on the specified datastore, delete the datastore (after detaching from each ESX host), offline the storage objects and destroy it (freeing space).

**Status:** Current (added in version 3.0)

**Type:** Synchronous

#### Parameters:

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| dsSpec | DatastoreSpec | See "DatastoreSpec" on page 83 | Specification describing datastore resize request. |
| requestSpec | RequestSpec | See "RequestSpec" on page 78 | Request can specify a vCenter server only. This method does not support direct connections to ESX hosts. |

- XML

```
  <complexType name="destroyDatastore">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="arg0"
type="{http://server.kamino.netapp.com/}datastoreSpec"
minOccurs="0"/>
          <element name="arg1"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

#### Return:

- Description

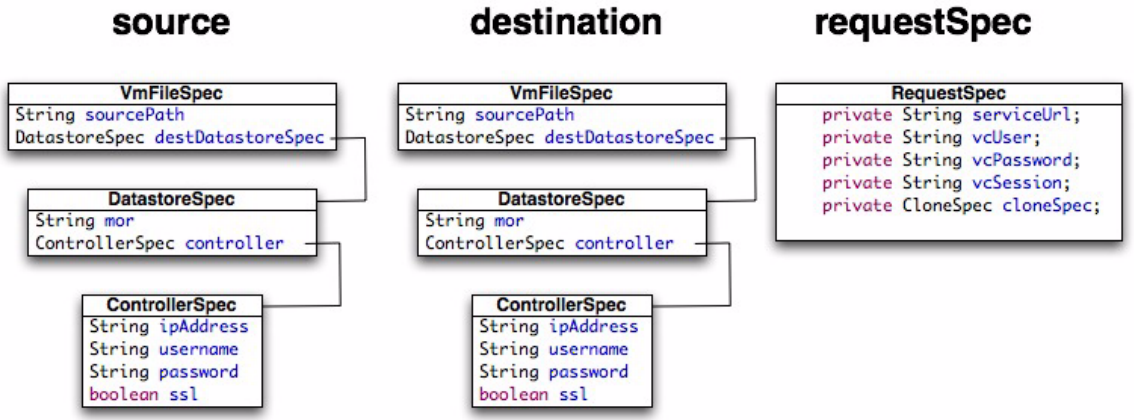
| Type | Example | Description |
|---|---|---|
| String | `Task:task-2` | A managed object reference to a vCenter task. |

- XML

```
<complexType name="destroyDatastoreResponse">
  <complexContent>
    <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="return"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

## Copy/Clone offload engine

The `fileCopyOffload` and `fileCloneOffload` take `VmFileSpec` and `RequestSpec` as arguments. The `DatastoreSpec` should contain only the datastore managed object reference and a reference to the Controller. This is defined as 'Existing Datastore' in the formulas section of the `DatastoreSpec` documentation.




### fileCopyOffload

Offload the copy of an NFS datastore file to the controller. This method should be used in cases where a full copy (all unique blocks) is required. In all other cases, the `fileCloneOffload` should be used.

This process involves a start-up time, which is quickly recovered when copying large files (because the offloaded controller base copy is very efficient). This start-up time may cause the offloaded copy of small files to take longer than using a host based copy.

This method supports copying a file within the same controller. The `VmFileSpec` for the source and destination must specify the same controller.

**Status:** Current (added in version 3.0)

**Type:** Asynchronous

**Parameters:**

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| source | VmFileSpec | See "VmFile Spec" on page 82 | Specification describing the source file (datastore and controller). |
| destination | VmFileSpec | See "VmFile Spec" on page 82 | Specification describing the destination file (datastore and controller). |
| requestSpec | RequestSpec | See "Request Spec" on page 78 | Request can specify a vCenter server or ESX host. |

- XML

```
 <complexType name="fileCopyOffload">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="arg0"
type="{http://server.kamino.netapp.com/}vmFileSpec"
minOccurs="0"/>
         <element name="arg1"
type="{http://server.kamino.netapp.com/}vmFileSpec"
minOccurs="0"/>
         <element name="arg2"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

**Return:**

- Description

| Type | Example | Description |
|---|---|---|
| Integer | `876234` | The operation identifier to monitor using `getFileOpOffloadStatus`. |

- XML

```
 <complexType name="fileCopyOffloadResponse">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="return"
type="{http://www.w3.org/2001/XMLSchema}int"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

### fileCloneOffload

Offload the space efficient clone of an NFS datastore file to the controller. This process uses the file level FlexClone feature of the controller. This process automatically falls back to the controller offloaded copy as needed if the `fallBackToCopy` parameter is set to `true`.

This method supports only cloning the file within the same volume. If `fallBackToCopy` is set to `true`, this method supports copying file within the same controller. In both cases, the `VmFileSpec` for the source and destination must specify the same controller.

The most effective use of this method is to employ a strategy where the output of the first operation (the destination file) becomes the input (the source file) for the next operation. For example, to create three clones of `test-flat.vmdk`, the following process (pseudo code) will be the most efficient:

```
clone(test-flat.vmdk,test1-flat.vmdk)
clone(test1-flat.vmdk,test2-flat.vmdk)
clone(test2-flat.vmdk,test3-flat.vmdk)
```

**Status:** Current (added in version 3.0)

**Type:** Asynchronous

**Parameters:**

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| source | VmFileSpec | See "VmFile Spec" on page 82 | Specification describing source file (datastore and controller). |
| destination | VmFileSpec | See "VmFile Spec" on page 82 | Specification describing destination file (datastore and controller). |
| fallBackToCopy | boolean | `true` | If set to `true`, engine runs in "fully automatic mode" which falls back to an offloaded copy as needed. If `false`, conditions that would normally fall back to a copy result in an error (which the caller must deal with). |
| requestSpec | RequestSpec | See "Request Spec" on page 78 | Request can specify a vCenter server or ESX host. |

- XML

```
 <complexType name="fileCloneOffload">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="arg0"
type="{http://server.kamino.netapp.com/}vmFileSpec"
minOccurs="0"/>
         <element name="arg1"
type="{http://server.kamino.netapp.com/}vmFileSpec"
minOccurs="0"/>
         <element name="arg2"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
         <element name="arg3"
type="{http://www.w3.org/2001/XMLSchema}boolean" minOccurs="0"/>
       </sequence>
```

```
    </restriction>
  </complexContent>
</complexType>
```

**Return:**

- Description

| Type | Example | Description |
|---|---|---|
| Integer | `876234` | The operation identifier to monitor using `getFileOpOffloadStatus`. |

- XML

```
<complexType name="fileCloneOffloadResponse">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="return"
type="{http://www.w3.org/2001/XMLSchema}int"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

### getFileOpOffloadStatus

Use this method to track the progress of a `fileCopyOffload` or `fileCloneOffload` operation. The status will be complete, failed, or running. When this method returns a `StatusMessage` with a status of complete or failed, the operation information is marked for cleanup which will occur in five minutes. After the operation information is cleaned up, it is no longer visible via this method. The percent of the operation is also present in the progress field.

**Status:** Current (added in version 3.0)

**Type:** Synchronous

**Parameters:**

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| opId | Integer | `876234` | The operation identifier returned from `fileCopyOffload` or `fileCloneOffload`. |

XML

```
 <complexType name="getFileOpOffloadStatus">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="arg0"
type="{http://www.w3.org/2001/XMLSchema}int"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

**Return:**

- Description

| Type | Example | Description |
|---|---|---|
| StatusMessage | See "StatusMessage" on page 91 | Information describing status, progress, and reason for error (if operation fails). |

- XML

```
 <complexType name="getFileOpOffloadStatusResponse">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="return"
type="{http://server.kamino.netapp.com/}statusMessage"
minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

### clearAllFinishedOpOffloadStatus

This method starts the cleanup timer described in `getFileOpOffloadStatus` for all operations that have a status of complete or failed.

**Status:** Current (added in version 3.0)

**Type:** Synchronous

### Parameters:

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| opId | Integer | `876234` | The operation identifier returned from `fileCopyOffload` or `fileCloneOffload`. |

- XML

```
  <complexType name="clearAllFinishedOpOffloadStatus">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

### Return:

- Description

Void

- XML

```
  <complexType name="clearAllFinishedOpOffloadStatusResponse">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

### getVms

Get the list of VMs that were created using `createClones` method. This is a list that can be used in the `redeployVMs` method.

**Status:** Current (added in version 3.0)

**Type:** Synchronous

**Parameters:**

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| vmMorRef | String | `VirtualMachine :vm-40` | The managed object reference of the VM. |
| requestSpec | RequestSpec | See "RequestSpec" on page 78 | Request can specify a vCenter server or ESX host. |

- XML

```
 <complexType name="getVms">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="arg0"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
         <element name="arg1"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

### Return:

- Description

| Type | Example | Description |
|---|---|---|
| List< VmSpec> | See "VmSpec" on page 88 | List of `VmSpec` based on the VM specified. A `VmSpec` is returned for each VM created using `createClones` method. This is used as input to the `redeployVMs` method. |

- XML

```
 <complexType name="getVmsResponse">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="return"
type="{http://server.kamino.netapp.com/}vmSpec"
maxOccurs="unbounded" minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

## getVmFiles

Get a skeleton list of `VmFileSpec` to be completed and used in the submission to `createClones`.

**Status:** Current (added in version 2.1)

**Type:** Synchronous

### Parameters:

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| `vmMorRef` | String | `VirtualMachine:vm-40` | The managed object reference of the VM. |
| `requestSpec` | Request Spec | See "RequestSpec" on page 78 | Request can specify a vCenter server or ESX host |

- XML

```
 <complexType name="getVmFiles">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="arg0"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
         <element name="arg1"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

**Return:**

- Description

| Type | Example | Description |
|---|---|---|
| List< VmFileSpec> | See "VmFile Spec" on page 82 | A list of `VmFileSpec` based on the VM specified. This information should be modified and submitted using the `CloneSpec`. |

- XML

```
 <complexType name="getVmFilesResponse">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="return"
type="{http://server.kamino.netapp.com/}vmFileSpec"
maxOccurs="unbounded" minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

### getMoref

Returns the managed object reference of the requested object based on name and type. The `getMoref` returns the first object that matches the name and type combination. For this reason, this method should not be used in production environments unless all object names are unique.

**Status:** Current (added in version 2.1)

**Type:** Synchronous

### Parameters:

- Description

| Name | Type | Example | Description |
|---|---|---|---|
| `name` | String | `winXP` | Name of object to look for. |
| `type` | String | `VirtualMachine` | Managed object type. |
| `requestSpec` | RequestSpec | See “RequestSpec” on page 78 | Request can specify a vCenter server or ESX host. |

- XML

```
 <complexType name="getMoref">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="arg0"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
         <element name="arg1"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
         <element name="arg2"
type="{http://server.kamino.netapp.com/}requestSpec"
minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

### Return:

- Description

| Type | Example | Description |
|---|---|---|
| String | `VirtualMachine:vm-40` | Managed object reference in a string format. |

- XML

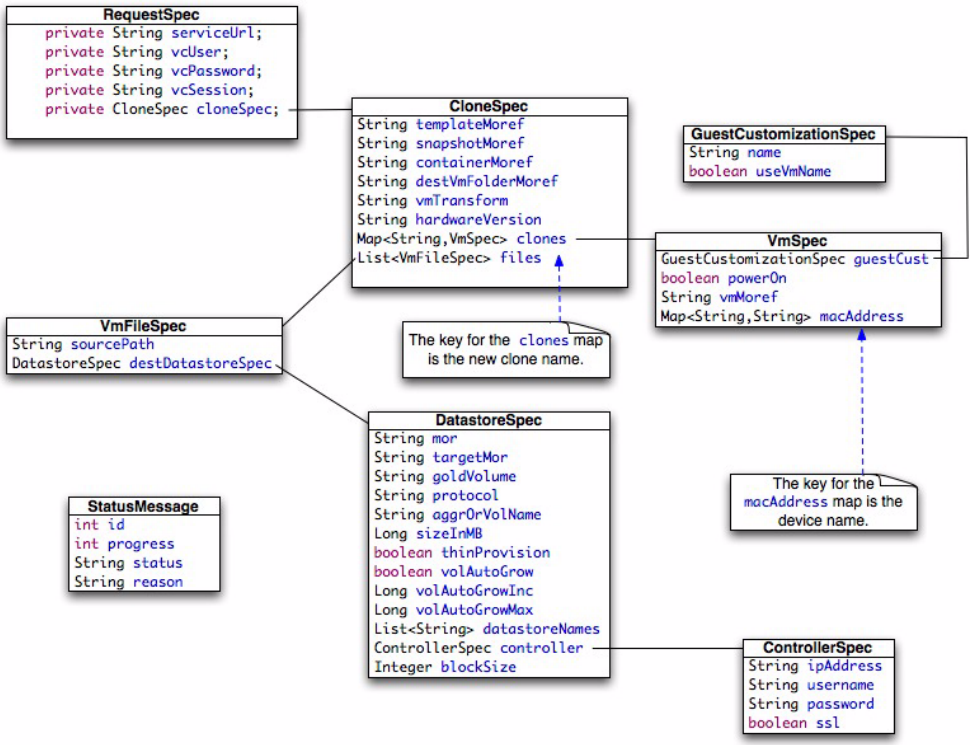
```
<complexType name="getMorefResponse">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="return"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
       </sequence>
```

```
    </restriction>
  </complexContent>
</complexType>
```

# Specifications and messages

This section includes the following topics


- "RequestSpec" on page 78
- "CloneSpec" on page 79
- "VmFile Spec" on page 82
- "DatastoreSpec" on page 83
- "ControllerSpec" on page 87
- "VmSpec" on page 88
- "GuestCustomizationSpec" on page 90
- "StatusMessage" on page 91

# RequestSpec

The Request Specification describes the URL of the VMware vCenter SDK or ESX Host as well as the authentication information. The authentication information may be in the form of a user name and password combination or a VMware Session ID. An optional Clone Specification may also be present.

### Properties

| Name | Type | Example | Description |
|---|---|---|---|
| serviceUrl | String | `https://10.10.10.1/sdk` | URL for the VMware vCenter SDK |
| vcUser | String | `Administrator` | VMware vCenter username (null ok if using vcSession) |
| vcPassword | String | `pass123` | VMware vCenter password (null ok if using vcSession) |
| vcSession | String | `CDF37576-C2E2-461A-8797-D33FBA6D5D82` | VMware session (null ok if using vcUser/vcPassword) |
| cloneSpec | CloneSpec | | A clone specification |

### Notes

- `cloneSpec` may be null when using this spec with anything other than `redeployVMs` or `createClones`.
- `vcSession` should be null if `vcUser` and `vcPassword` are used.
- `vcUser` and `vcPassword` should be null if `vcSession` is used.

### XML

```
 <complexType name="requestSpec">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="cloneSpec"
type="{http://server.kamino.netapp.com/}cloneSpec" minOccurs="0"/>
         <element name="serviceUrl"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
         <element name="vcPassword"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
```

```
        <element name="vcSession"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        <element name="vcUser"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
 </complexType>
```

# CloneSpec

The Clone Specification describes a request to create clones of a VM or template or to re-deploy the virtual hard drives. When used with the `redeployVMs` method, each VM in the map named `clones` will have its virtual hard drives replaced with those of the source. The String in this Map is the name of the Virtual Machine to be re-deployed and the `VmSpec` describes this Virtual Machine.

When used with the `createClones` method, a new VM is created for each entry in the Map named `clones`. The String in this Map is the name of the clone and the `VmSpec` describes the new clone configuration. The List named `files` describes the files that make up the source VM or template. This list can be used to specify different destinations for each file as well as create new datastores.

### Properties

| Name | Type | Example | Description |
|---|---|---|---|
| `templateMoref` | String | `VirtualMachine:vm-255` | Source VM or template of cloning operation. String representation of type and value of `ManagedObjectReference` from VMware VI API. |
| `snapshotMoref` | String | `VirtualMachineSnapshot: snapshot-10244` | The managed object reference for a snapshot of the source virtual machine to base the clones on. |

| Name | Type | Example | Description |
|---|---|---|---|
| containerMoref | String | Datacenter:datacenter-2 | Destination for resulting clones. Valid destination types: Datacenter, ResourcePool, ClusterComputeResource, and ComputeResource, A string representation of type and value of ManagedObjectReference from the VMware VI API. |
| destVmFolderMoref | String | Folder:group-v1266 | VM folder the clones should be created in. If null, clones are created at the root VM folder. |
| vmTransform | String | sparse | Transforms all virtual hard drives to specified format. Should be specified only when there is actual work to do. Specifying a transform when one is not required causes unnecessary work. Options are null, flat, and sparse. |
| hardwareVersion | String | vmx-07 | Upgrade hardware version from a previous version to vmx-04 or vmx-07. NOTE: vmx-04 is supported by ESX 3.5 and both are supported by ESX 4.0. |
| clones | Map<String,VmSpec> | | Map of new VM name to VM specification (VmSpec). |
| files | List<VmFileSpec> | | List of files that make up source VM or template specified in templateMoref. |

**XML**

```
<complexType name="cloneSpec">
  <complexContent>
    <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="clones">
```

```
            <complexType>
              <complexContent>
                <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
                  <sequence>
                    <element name="entry" maxOccurs="unbounded"
minOccurs="0">
                      <complexType>
                        <complexContent>
                          <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
                            <sequence>
                              <element name="key"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
                              <element name="value"
type="{http://server.kamino.netapp.com/}vmSpec" minOccurs="0"/>
                            </sequence>
                          </restriction>
                        </complexContent>
                      </complexType>
                    </element>
                  </sequence>
                </restriction>
              </complexContent>
            </complexType>
          </element>
          <element name="connBroker"
type="{http://server.kamino.netapp.com/}connectionBrokerSpec"
minOccurs="0"/>
          <element name="containerMoref"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="destVmFolderMoref"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="files"
type="{http://server.kamino.netapp.com/}vmFileSpec"
maxOccurs="unbounded" minOccurs="0"/>
          <element name="hardwareVersion"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="snapshotMoref"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="templateMoref"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="vmTransform"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

# VmFile Spec

This describes the source configuration file (vmx) or the source virtual hard disk files (vmdk) as well as the destination datastore specification.

### Properties

| Name | Type | Example | Description |
|---|---|---|---|
| `sourcePath` | String | `[testSource] demoSource/demoSource.vmdk` | Path to vmx or vmdk file. The string `Configuration File` can be passed in place of an actual vmx file. |
| `destDatastoreSpec` | DatastoreSpec | See "DatastoreSpec" on page 83 | Destination datastore specification. |

### XML

```
<complexType name="vmFileSpec">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="destDatastoreSpec"
type="{http://server.kamino.netapp.com/}datastoreSpec"
minOccurs="0"/>
         <element name="sourcePath"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
```

# DatastoreSpec

DatastoreSpec describes the destination datastore. This can describe new datastores to be created as well as existing datastores. See the "Required parameters" section below for valid parameter combinations. The number of clones must be evenly divisible by the number of datastores.

### Properties

| Name | Type | Example | Description |
|---|---|---|---|
| `mor` | String | `Datastore:datastore-31` | Destination datastore. String representation of type and value of `ManagedObjectReference` from the VMware VI API. |
| `targetMor` | String | `Datacenter:datacenter-2` | The managed object reference of the vCenter object in which to attach the new datastore. Can be an ESX host, cluster or datacenter. |
| `goldVolume` | String | `goldVolume` | Name of volume used when creating more than one NFS-based datastore. This volume is not permanently presented to the ESX hosts. This volume becomes the parent of the FlexClones. |
| `protocol` | String | `FCP` | Valid values are NFS, FCP, ISCSI. |
| `aggrOrVolName` | String | `vmwareLuns` | Name of the aggregate for new NFS datastores or name of volume for new VMFS datastores. |
| `sizeInMB` | Long | `204800` | Size of the datastore in MB. An additional 256MB is added for VMFS datastores to cover metadata overhead. |
| `thinProvision` | Boolean | `true` | If `true`, space will not be reserved for the storage object. For NFS, the volume will guarantee=none. For VMFS, LUN will be created with '-o noreserve'. |
| `volAutoGrow` | Boolean | `true` | If true, the `volAutoGrowInc` and `volAutoGrowMax` values are applied to the volume. |

| Name | Type | Example | Description |
|---|---|---|---|
| volAutoGrowInc | Long | 204800 | Increment in which to grow volume automatically as needed in MB. |
| volAutoGrowMax | Long | 204800 | Maximum size to which to grow the volume automatically in MB. |
| datastoreNames | List<String> | newDatastore | List of datastore names. Care should be taken by the application to prevent duplicate datastore, volume or LUN names. For NFS, datastore name is used as volume name. For VMFS, datastore name is used as LUN name. |
| numDatastores | Int | 1 | Number of datastores. This should indicate the size of the list of names in datastoreNames. |
| blockSize | Int | 1 | VMFS block size in MB. |
| controller | ControllerSpec | | The NetApp controller. |

**Required parameters**

**Specifying an existing datastore:**

- mor
- controller

**Specifying new NFS datastores using createClones or createDatastore:**

- targetMor - only required for createDatastore
- aggrOrVolName
- sizeInMB
- thinProvision
- volAutoGrow
- volAutoGrowInc
- volAutoGrowMax
- protocol - must be NFS
- controller
- datastoreNames - only one name in the list
- numDatastores - should be 1

**Specifying new NFS datastores using createClones:**

- goldVolume
- aggrOrVolName
- sizeInMB

- thinProvision
- volAutoGrow
- volAutoGrowInc
- volAutoGrowMax
- protocol - Must be NFS
- controller
- datastoreNames
- numDatastores

**Specifying new VMFS datastores using createClones or createDatastore:**

- targetMor - only required for createDatastore
- aggrOrVolName
- sizeInMB
- thinProvision
- protocol - must be FCP or ISCSI
- controller
- datastoreNames - only one name in the list
- numDatastores - should be 1

**Specifying new VMFS datastores using createClones:**

- aggrOrVolName
- sizeInMB
- thinProvision
- protocol - must be FCP or ISCSI
- controller
- datastoreNames
- numDatastores

**XML**

```
 <complexType name="datastoreSpec">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="aggrOrVolName"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="blockSize"
type="{http://www.w3.org/2001/XMLSchema}int" minOccurs="0"/>
          <element name="controller"
type="{http://server.kamino.netapp.com/}controllerSpec"
minOccurs="0"/>
```

```
          <element name="datastoreNames"
type="{http://www.w3.org/2001/XMLSchema}string"
maxOccurs="unbounded" minOccurs="0"/>
          <element name="goldVolume"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="mor"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="numDatastores"
type="{http://www.w3.org/2001/XMLSchema}int"/>
          <element name="protocol"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="sizeInMB"
type="{http://www.w3.org/2001/XMLSchema}long" minOccurs="0"/>
          <element name="targetMor"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="temp"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="thinProvision"
type="{http://www.w3.org/2001/XMLSchema}boolean"/>
          <element name="volAutoGrow"
type="{http://www.w3.org/2001/XMLSchema}boolean"/>
          <element name="volAutoGrowInc"
type="{http://www.w3.org/2001/XMLSchema}long" minOccurs="0"/>
          <element name="volAutoGrowMax"
type="{http://www.w3.org/2001/XMLSchema}long" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
 </complexType>
```

# ControllerSpec

This describes the controller connection and authentication data. This information is used by the RCU to connect to the controller using the ZAPI interface. No other protocol is used to connect to the controller.

### Properties

| Name | Type | Example | Description |
|---|---|---|---|
| `ipAddress` | String | `10.10.10.2` | IP or host name of the NetApp controller. |
| `username` | String | `root` | User name (does not need to be root). |
| `password` | String | `pass123` | Password. |
| `ssl` | Boolean | `true` | If `true`, use HTTPS. If `false`, use HTTP to connect to the controller. |

### XML

```
  <complexType name="controllerSpec">
    <complexContent>
      <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="ipAddress"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="password"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="ssl"
type="{http://www.w3.org/2001/XMLSchema}boolean"/>
          <element name="username"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

# VmSpec

This describes configuration and action options for each VM created. The MAC address must be in a range defined by VMware. Refer to VMware documentation for more information.

### Properties

| Name | Type | Example | Description |
|---|---|---|---|
| macAddress | Map<String,String> | "Network Adapter 1", "00:0c:29:39:8c:c1" | Virtual network adapter to MAC address information (optional). |
| custSpecName | String | winXP_HR | Name of guest customization specification to be applied (optional). |
| vmMoref | String | VirtualMachine:vm-255 | The managed object reference of the virtual machine to be re-deployed. |
| powerOn | Boolean | false | If true, the new VMs are powered on after all have been created. |

### XML

```
<complexType name="vmSpec">
   <complexContent>
     <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
       <sequence>
         <element name="custSpec"
type="{http://server.kamino.netapp.com/}guestCustomizationSpec"
minOccurs="0"/>
         <element name="domain"
type="{http://server.kamino.netapp.com/}domainSpec"
minOccurs="0"/>
         <element name="macAddress">
           <complexType>
             <complexContent>
               <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
                 <sequence>
                   <element name="entry" maxOccurs="unbounded"
minOccurs="0">
                     <complexType>
                       <complexContent>
                         <restriction
base="{http://www.w3.org/2001/XMLSchema}anyType">
                           <sequence>
```

```
                              <element name="key"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
                              <element name="value"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
                            </sequence>
                          </restriction>
                        </complexContent>
                      </complexType>
                    </element>
                  </sequence>
                </restriction>
              </complexContent>
            </complexType>
          </element>
          <element name="powerOn"
type="{http://www.w3.org/2001/XMLSchema}boolean"/>
          <element name="vmMoref"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

# GuestCustomizationSpec

### Properties

| Name | Type | Example | Description |
|---|---|---|---|
| name | String | `WindowsSysprep2` | Name of the guest customization specification. |
| useVmName | Boolean | `true` | If guest customization specification is of type `CustomizationSysprepText`, this option can be used make the guest hostname match the virtual machine name. Note: It is the responsibility of the implementer to ensure the virtual machine name will result in a valid host name. |

### XML

```
<complexType name="guestCustomizationSpec">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="name"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        <element name="useVmName"
type="{http://www.w3.org/2001/XMLSchema}boolean"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

# StatusMessage

### Properties

| Name | Type | Example | Description |
|---|---|---|---|
| id | Int | `123` | Operation identifier. |
| progress | Int | `51` | Valid values are 0-100. Indicates how much of the copy or clone process has completed at the time of the query. |
| status | String | `running` | Valid values are; `complete` (finished with out error), `failed` (finished with error), or `running` (operation in progress). |
| reason | String | `Clone operation on file '/vol/test/dir/file1-flat.vmdk' failed with error 'Stale donor inode'. The clone operation ID is 911.` | If the status is `failed`, this will contain the reason for the failure. |

### XML

```
<complexType name="statusMessage">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="id"
type="{http://www.w3.org/2001/XMLSchema}int"/>
        <element name="progress"
type="{http://www.w3.org/2001/XMLSchema}int"/>
        <element name="reason"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        <element name="status"
type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

# Example

```
<ns2:createClonesxmlns:ns2="http://server.rcu.netapp.com/">
<arg0>
   <cloneSpec>
       <clones>
           <entry>
               <key>apiTestClone1</key>
               <value>
                   <powerOn>false</powerOn>
               </value>
           </entry>
          <entry>
               <key>apiTestClone2</key>
               <value>
                   <powerOn>false</powerOn>
               </value>
           </entry>
          <entry>
               <key>apiTestClone3</key>
               <value>
                   <powerOn>false</powerOn>
               </value>
           </entry>
       </clones>
       <containerMoref>Datacenter:datacenter-2</containerMoref>
       <files>
           <destDatastoreSpec>
               <controller>
                   <ipAddress>10.10.10.2</ipAddress>
                   <password></password>
                   <ssl>false</ssl>
                   <username>root</username>
               </controller>
               <mor>Datastore:datastore-17</mor>
               <numDatastores>0</numDatastores>
               <thinProvision>false</thinProvision>
               <volAutoGrow>false</volAutoGrow>
           </destDatastoreSpec>
           <sourcePath>[unitTestSourceNFS]
demoSource/demoSource.vmx</sourcePath>
       </files>
       <files>
           <destDatastoreSpec>
                <controller>
                   <ipAddress>10.10.10.2</ipAddress>
                   <password></password>
                   <ssl>false</ssl>
                   <username>root</username>
```

```
                </controller>
                <mor>Datastore:datastore-17</mor>
                <numDatastores>0</numDatastores>
                <thinProvision>false</thinProvision>
                <volAutoGrow>false</volAutoGrow>
            </destDatastoreSpec>
            <sourcePath>[unitTestSourceNFS]
demoSource/demoSource.vmdk</sourcePath>
        </files>
        <templateMoref>VirtualMachine:vm-255</templateMoref>
    </cloneSpec>
    <serviceUrl>https://10.10.10.2/sdk</serviceUrl>
    <vcPassword>pass123</vcPassword>
    <vcUser>Administrator</vcUser>
</arg0>
</ns2:createClones>
```

# Java

This information is for use with Java-based tools as well as the Java programming language.

## Java SSL Security

**Issue:** Java SSL security demands a hostname in the CN to work properly. The IP address that RCU configures during application installation is not adequate.

**Solution:**

1. Stop the RCU service.
2. On the RCU server, delete the keystore from the `conf` directory in the installation path. For example:

   ```
   del "C:\Program Files\netapp\Rapid Cloning Utility\conf\keyStore"
   ```

3. Create a keystore using the hostname (instead of ip address) as the hostname.

   The following example uses `vc123.domain.com` as the name of the VC server where RCU was installed:

   ```
   "C:\Program Files\netapp\Rapid Cloning
   Utility\jre\bin\keytool.exe" -genkeypair -alias RCU -keyalg RSA
   -keysize 1024 -dname "CN=vc123.domain.com, OU=Department,
   O=Company, L=Somewhere, S=Somestate, C=US,
   EMAILADDRESS=root@localhost" -validity 9999 -storepass
   RCUVersion2 -keypass RCUVersion2  -keystore "C:\Program
   Files\netapp\Rapid Cloning Utility\conf\keyStore"
   ```

4. Start the RCU service.
5. Import the cert into the keystore that the client application (the consumer of the RCU API) uses. This keystore is in a different location than the RCU keystore and is likely even on a separate machine. There are several references on the internet for how to do this; these can be found using "keytool import" as search criteria.
6. Make sure the application uses the host name in the URL. For example:

   https://vc123.domain.com:61921/rcu/api?wsdl

If, after completing the steps described above, you are still having trouble, see the"Https java security debugger" recipe for more information: http://communities.netapp.com/docs/DOC-3578

### Java Samples

The examples provided in the API directly were built using Java 6. The following steps were taken to create the supporting classes:

```
cd "C:\Program Files\netapp\Rapid Cloning
Utility\api\Samples\java\src"
wsimport -keep -p com.netapp.rcu.api
https://vc123.domain.com:61921/rcu/api?wsdl
```