

Linked Virtual Machines

vSphere Web Services SDK

VMware vSphere supports linked virtual machines, which are two or more related virtual machines that share virtual disk storage. Also called linked clones, linked virtual machines make efficient use of duplicated data. The vSphere Web Services SDK offers a number of interfaces to support linked virtual machines.

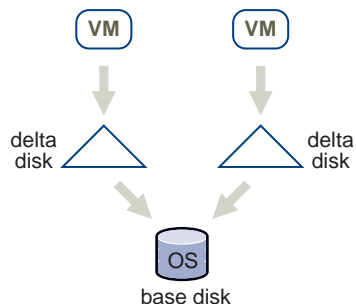
This technical note lists properties that support linked virtual machines and explains how you can create linked virtual machines using the vSphere Web Services SDK. This technical note uses pseudo code examples. See `SDK\samples\Axis\java\com\vmware\samples\vm\VMLinkedClone.java`, or in newer releases, `SDK\vsphere-ws\java\JAXWS\samples\com\vmware\vm\VMLinkedClone.java`, for real code examples.

Linked Virtual Machines and Disk Backings

In its simplest form, shared storage is achieved through the use of delta disk backings. A delta disk backing is a virtual disk file that sits on top of a standard virtual disk backing file. Each time the guest operating system on a virtual machine writes to disk, the data is written to the delta disk. Each time the guest operating system on a virtual machine reads from disk, the virtual machine first targets the disk block in the delta disk. If the data is not on the delta disk, the virtual machine looks for them on the base disk.

Linked virtual machines can be created from a snapshot or from the current running point. After you create linked virtual machines, they share the base disk backing. Each virtual machine has its own delta disk backing, as shown in [Figure 1](#). You can have up to eight virtual machines in a linked clone group.

Figure 1. Linked Virtual Machines with Shared Base Disk Backing and Separate Delta Disk Backing



Limitation for HA Clusters

Virtual machines in a linked clone group can be part of a VMware HA (high availability) cluster. The number of hosts in a cluster affects HA's ability to restart a failed virtual machine. If the cluster has eight or fewer hosts, then linked virtual machines restart properly. However, if the cluster has more than eight hosts, HA might not be able to restart a virtual machine after it fails. This is because HA is not aware that virtual machines in a linked clone group are subject to the eight host limit, so when responding to a failure, HA might try to restart the virtual machine on a host that cannot participate in the group due to the maximum host limit. HA will attempt failover five times to different hosts. Thus, in clusters with 13 or more hosts, it is possible that HA will never try a host that belongs to the linked clone group. This behavior is the same for vSphere 4.0, 4.1, and 5.0.

Creating a Linked Virtual Machine

You can create linked virtual machines in one of two ways:

- Clone a virtual machine from a snapshot.
- Clone a virtual machine from the current virtual machine state, which can differ from the snapshot point.

Create a Linked Virtual Machine From a Snapshot

You first create a snapshot, then create a linked virtual machine from the snapshot.

Create the Snapshot

To create the snapshot, call `CreateSnapshot` on the virtual machine. The virtual machine can be in any power state. The following pseudo code creates a snapshot named `snap1`, with description string. The snapshot does not include a memory dump (`False`). If the virtual machine is powered on, VMware Tools is called to quiesce the file system in the virtual machine (`True`).

```
myVm.CreateSnapshot("snap1", "snapshot for creating linked virtual machines", False, True)
```

Create the Linked Virtual Machine

To create a linked virtual machine, use a `VirtualMachineRelocateDiskMoveOptions.diskMoveType` of `createNewDeltaDiskBacking` and specify the snapshot you just created, as illustrated in [Example 1](#). The virtual machine with the snapshot can be in any power state.

Example 1. Creating a Linked Virtual Machine from a Snapshot

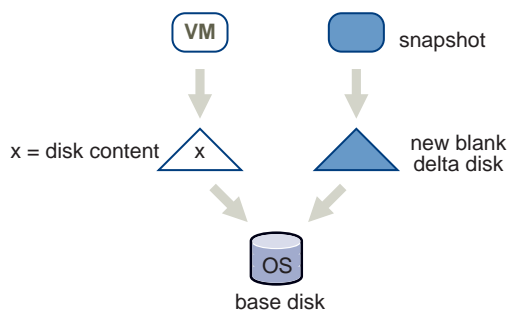
```
relSpec = new VirtualMachineRelocateSpec()
relSpec.diskMoveType = VirtualMachineRelocateDiskMoveOptions.createNewChildDiskBacking

cloneSpec = new VirtualMachineCloneSpec()
cloneSpec.powerOn = False
cloneSpec.template = False
cloneSpec.location = relSpec
cloneSpec.snapshot = myVm.snapshot.currentSnapshot

myVm.Clone(myVm.parent, myVm.name + "-clone", cloneSpec)
```

The result is a virtual machine with the same base disk as the original, but a new delta disk backing.

Figure 2. Creating a Virtual Machine from a Snapshot



The `createNewDeltaDiskBacking` option should be used only after taking a snapshot because the base disk remains writable. VMware View uses this method for creating linked clones to run desktop virtual machines.

Create a Linked Virtual Machine From the Current Running Point

To create a virtual machine from the current running point, clone the virtual machine, as in [Example 1](#), but use a `diskMoveType` of `moveChildMostDiskBacking`. The virtual machine can be in any power state. A snapshot is not needed for this operation.

Example 2. Creating a Linked Virtual Machine from the Current Running Point

```

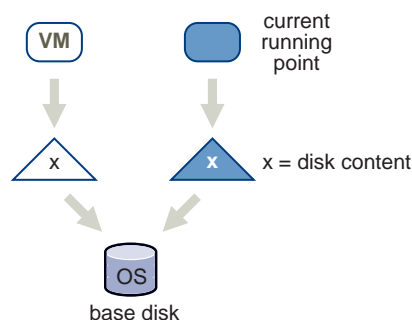
reSpec = new VirtualMachineRelocateSpec()
reSpec.diskMoveType = VirtualMachineRelocateDiskMoveOptions.moveChildMostDiskBacking

cloneSpec = new VirtualMachineCloneSpec()
cloneSpec.powerOn = False
cloneSpec.template = False
cloneSpec.location = reSpec

myVm.Clone(myVm.parent, myVm.name + "-clone", cloneSpec)

```

The result is a virtual machine with the same base disk as the running point, but a new delta disk backing.

Figure 3. Creating a Virtual Machine from the Current Running Point

A snapshot is not needed for the `moveChildMostDiskBacking` option because the base disk is read-only.

Removing Snapshots and Deleting Virtual Machines

After you have created a group of linked virtual machines, you might want to remove the snapshot that was the basis for a linked virtual machine, or delete a virtual machine. Those two actions affect the disks in the linked virtual machine group. For disk consolidation or deletion, it is preferable to perform the actions when connected to a vCenter Server.

- Snapshot removal – During snapshot removal, the snapshot metadata is also removed, and the virtual machine from which the snapshot was taken is no longer shown as having snapshots. If you remove a snapshot when connected directly to an ESX/ESXi host, shared disks are not consolidated and unnecessary levels of delta disks might result. If you remove a snapshot when connected to the vCenter Server, shared disks are not consolidated, but unshared disks are consolidated.
- Virtual machine deletion – If you delete a virtual machine when connected directly to an ESX/ESXi host, shared disks are not deleted. If you delete a virtual machine when connected to the vCenter Server, shared disks are not deleted, but unshared disks are deleted.

Relocating a Virtual Machine in a Linked Virtual Machine Group

You can move the virtual machines in a linked virtual machine group between datastores to save storage, as shown in [Figure 3](#). The contents of the delta disk might not be as important as the contents of the base, and you can save storage by removing the delta disk.

Example 3. Relocating a Linked Virtual Machine

```

reSpec = new VirtualMachineRelocateSpec()
reSpec.diskMoveType = VirtualMachineRelocateDiskMoveOptions.moveChildMostDiskBacking
reSpec.datastore = localDatastore

myVm.Relocate(reSpec)

```

You can relocate multiple linked virtual machines to a new datastore, but retain all shared storage during the relocation. To achieve the relocation, relocate the desired virtual machines one by one, giving the option to allow reattaching to an existing disk, as shown in [Example 4](#).

Example 4. Relocating Multiple Linked Virtual Machines

```
relSpec = new VirtualMachineRelocateSpec()
relSpec.diskMoveType = VirtualMachineRelocateDiskMoveOptions.moveAllDiskBackingsAndAllowSharing
relSpec.datastore = targetDatastore

myVm.Relocate(relSpec)
```

Promoting a Virtual Machine's Disk

Promoting a virtual machine's disk improves performance.

IMPORTANT You can use the `PromoteDisks` API only when connected to a vCenter Server system.

You can use `PromoteDisks` to copy disk backings or to consolidate disk backings.

- **Copy** – If the `unlink` parameter is `true`, any disk backing that is shared by multiple virtual machines is copied so that this virtual machine has its own unshared version. Files are copied into the home directory of the virtual machine. This setting results in improved read performance, but higher space requirements. The following call copies and shares disks, then collapses all unnecessary disks:

```
myVm.PromoteDisks(True, [])
```

- **Consolidate** – If the `unlink` parameter is `false`, any disk backing that is not shared between multiple virtual machines and not associated with a snapshot is consolidated with its child backing. The net effect is improved read performance at the cost of inhibiting future sharing. The following call eliminates any unnecessary disks:

```
myVm.PromoteDisks(False, [])
```

Promoting a virtual machine's disk might also be useful if you end up with disk backings that are not needed for snapshots or for sharing with other virtual machines.

Both uses of `PromoteDisks` take an optional second argument, which allows you to apply the operation to only a subset of disks. Pseudo code of [Example 6](#) shows how you could unshare and consolidate only the virtual disk with key 2001:

Example 5. Promoting Disk of a Linked Virtual Machine

```
for any of my VMs in dev
  if (dev.key == 2001)
    disk2001 = dev

myVm.PromoteDisks(True, [disk2001])
```

Performing Advanced Manipulation of Delta Disks

For advanced manipulation of delta disks, you can use `VirtualDeviceConfigSpec` operations such as the `VirtualDeviceConfigSpec.create` and `VirtualDeviceConfigSpec.add` operations.

Both `add` and `create` allow you to create a blank delta disk on top of an existing disk. You can specify `add` or `create` in the `VirtualDeviceSpec` and pass in a virtual disk whose parent property points to an existing disk. The operations create a new delta disk whose parent is the existing disk.

One use case is adding a delta disk on top of an existing virtual disk in a virtual machine without creating a snapshot. [Example 6](#) illustrates how to add the delta disk for the first virtual disk in the virtual machine.

Example 6. Creating a Virtual Machine

```
disk = None
for any of my VMs in dev
  if (VirtualDisk.isinstance == dev):
    disk = dev

# Remove the disk
removeDev = new VirtualDeviceConfigSpec()
removeDev.operation = "remove"
removeDev.device = disk

# Create a new delta disk which has the
# original disk as its parent disk
addDev = new VirtualDeviceConfigSpec()
addDev.operation = "add"
addDev.fileOperation = "create"
addDev.device = copy.copy(disk)
addDev.device.backing = copy.copy(disk.backing)
addDev.device.backing.fileName = "[" + disk.backing.datastore.name + "]"
addDev.device.backing.parent = disk.backing

spec = new VirtualMachineConfigSpec()
spec.deviceChange = [removeDev, addDev]
vm.Reconfigure(spec)
```

If you have comments about this documentation, submit your feedback to: docfeedback@vmware.com

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

Copyright © 2009, 2011 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Item: EN-000248-01
