Provisioning and Cloning Capability for Virtual Storage Console 4.1 for VMware® vSphere™ **Programmable API 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

October 2012

Contents

Provisioning and Cloning programmable APIs

Virtual Storage Console for VMware vSphere provides a programmable application interface (API) for Provisioning and Cloning.

The API is exposed using Simple Object Access Protocol (SOAP). It provides a layer above the NetApp Controller API (called Manage ONTAP) and the VMware VI SDK, but does not require either of these in the consumer application or script.

The Provisioning and Cloning API is designed to be leveraged with the VI SDK. It provides end-to-end automated datastore provisioning and offloads the intricacies of storage object cloning while cloning virtual machines.

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 Provisioning and Cloning API exposes the virtual machine clone creation engine (which includes the redeploy feature), the datastore management engine (create, destroy, resize), 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` API.
- `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 clone creation and routing methods: createClones and redeployVMs.

- `createClones` can be used to create virtual machine clones on new or existing datastores. When 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 redeploy 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 for managing datastores: createDatastore, resizeDatastore, and destroyDatastore.

- The `createDatastore` method provides the ability to provision storage on the controller, present it to the ESX hosts, 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 hosts, destroy the storage objects on the controller, and free 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 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 is a combination of complex (specification and message) and simple (string, int, long, boolean, and so on) data types. The specifications and messages are described below.

Note: Very little verification or validation is done in the API. For example, if there is not enough space to create the requested datastore(s), the API method will fail.

Provisioning and Cloning methods

This section describes all the available Provisioning and Cloning methods.

Virtual machine clone creation and redeploy engine

This section describes the APIs for interfacing with the virtual machine clone creation and redeploy engine.

createClones

You can use the `createClones` method to create virtual machine 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 machine or template must not contain any RDMs, must not contain any devices that use `VMDirectPath`, and must be connected.

The mix of `VirtualIDEController` attached hard drives and `VirtualSCSIController` hard drives in the same virtual machine 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

Name	Type	Value	Description
requestSpec	Object	RequestSpec	Request can specify a vCenter server only. This method does not support direct connections to ESX hosts. **Note:** See *RequestSpec* on page 28.

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>
```

Returns

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

Return 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>
```

RedeployVMs

You can redeploy 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

Name	Type	Value	Description
requestSpec	Object	RequestSpec	**Note:** See *RequestSpec* on page 28.

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>
```

Returns

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

Return 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

This section describes the APIs for interfacing with the datastore management engine.

createDatastore

You can use the `createDatastore` method to provision storage on the storage controller, attach it to one or more ESX hosts 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

Name	Type	Value	Description
dsSpec	Object	DatastoreSpec	The specification describing the datastore to create. **Note:** See *DatastoreSpec* on page 33.

Name	Type	Value	Description
requestSpec	Object	RequestSpec	Request can specify a vCenter server only. This method does not support direct connections to ESX hosts. **Note:** See *RequestSpec* on page 28.

XML

```
complexType name="createDatastore">
    <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" minOccurs="0"/>
            </sequence>
        </restriction>
    </complexContent>
</complexType>
```

Returns

Type	Value	Description
String	`newDatastore`	The name of the new datastore that was created.

Return 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

You can use the `resizeDatastore` method to grow or shrink NFS-based datastores (and associated storage objects on the controller), and grow VMFS-based datastores (and associated storage objects on the controller).

Status

Current (added in version 3.0)

Type

Synchronous

Parameters

Name	Type	Value	Description
dsSpec	Object	DatastoreSpec	Specification describing datastore resize request.
requestSpec	Object	RequestSpec	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/}requestSpec" minOccurs="0"/>
                <element name="arg1" type="{http://
server.kamino.netapp.com/}datastoreSpec" minOccurs="0"/>
            </sequence>
        </restriction>
```

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

Returns

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

Return 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

You can use the `destroyDatastore` method to delete any virtual machine with a file on the specified datastore, delete the datastore (after detaching from each ESX host), take the storage objects offline, and destroy the datastore (to free space).

Status

Current (added in version 3.0)

Type

Synchronous

Parameters

Name	Type	Value	Description
dsSpec	Object	DatastoreSpec	Specification describing datastore resize request. **Note:** See *DatastoreSpec* on page 33.
requestSpec	Object	RequestSpec	Request can specify a vCenter server only. This method does not support direct connections to ESX hosts. **Note:** See *RequestSpec* on page 28.

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>
```

Returns

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

Return 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>
```

Connection Broker features

This section describes the APIs for interfacing with connection broker features.

performViewImport

You can use the `performViewImport` method to import the specified virtual machines into a VMware View Server.

Status

Current (added in version 3.2)

Type

Synchronous

Parameters

Name	Type	Value	Description
dsSpec	Object	DatastoreSpec	The specification describing the connection broker information.
requestSpec	Object	RequestSpec	Request can specify a vCenter server only. This method does not support direct connections to ESX hosts.

Name	Type	Value	Description
vmsForImport	List <String>		A list of the virtual machines (by name) that should be imported into the View server.

XML

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

Returns

Type	Value	Description
Void	N/A	Nothing returned

Return XML

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

createXenImportFile

You can use the `createXenImportFile` method to create a file for importing virtual machines into a Citrix XenDesktop server.

Status

Current (added in version 3.2)

Type

Synchronous

Parameters

Name	Type	Value	Description
dsSpec	Object	ConnectionBrokerSpec	The specification describing the connection broker information.
requestSpec	Object	RequestSpec	Request can specify a vCenter server only. This method does not support direct connections to ESX hosts.
vmsForImport	List		A list of the virtual machines (by name) that should be imported into the View server.

XML

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

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

Returns

Type	Value	Description
Void	N/A	Nothing returned

Return XML

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

Copy/Clone offload engine

This section describes the APIs for interfacing with the 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

You can use the `fileCopyOffload` method to 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

Name	Type	Value	Description
source	Object	VmFileSpec	Specification describing the source file (datastore and controller).
destination	Object	VmFileSpec	Specification describing the destination file (datastore and controller).
requestSpec	Object	RequestSpec	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>
```

Returns

Type	Value	Description
Integer	`876234`	The operation identifier to monitor using `getFileOpOffloadStatus`.

Return 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

You can use the `fileCloneOffload` method to offload the 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) is 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

Name	Type	Value	Description
source	Object	VmFileSpec	Specification describing the source file (datastore and controller).
destination	Object	VmFileSpec	Specification describing the destination file (datastore and controller).
fallBackToCopy	Boolean		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	Object	RequestSpec	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>
```

Returns

Type	Value	Description
Integer	`876234`	The operation identifier to monitor using `getFileOpOffloadStatus`.

Return 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

You can use the `getFileOpOffloadStatus` 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 occurs five minutes later. After the operation information has been cleaned up, it is no longer visible using this method. The progress field displays information about the progress of the operation.

Status

Current (added in version 3.0)

Type

Synchronous

Parameters

Name	Type	Value	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>
```

Returns

Type	Value	Description
String	StatusMessage	Information describing status, progress, and reason for error (if operation fails). **Note:** See *StatusMessage* on page 42.

Return 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

You can use the `clearAllFinishedOpOffloadStatus` method to start 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

Name	Type	Value	Description
opId	Integer		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>
```

Returns

Void

Return XML

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

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

Utility methods

This section describes utility methods that return output, such as a list of created virtual machines and the managed object reference of each virtual machine.

getVms

You can use the `getVms` method to obtain the list of virtual machines that were created using the `createClones` method. This list can be used in the `redeployVMs` method.

Status

Current (added in version 3.0)

Type

Synchronous

Parameters

Name	Type	Value	Description
vmMorRef	Object	The managed object reference of the VM.	The managed object reference of the VM.

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>
```

Returns

Type	Value	Description
vmMorRef	String	The managed object reference of the VM.
requestSpec	RequestSpec	Request can specify a vCenter server or ESX host.

Return XML

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

getVmFiles

You can use the `getVmFiles` method to obtain 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

Name	Type	Value	Description
vmMorRef	String		The managed object reference of the VM.
requestSpec	Object	RequestSpec	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>
```

Returns

Type	Value	Description
List	Object <VmFileSpec>	A list of VmFileSpec based on the VM specified. This information should be modified and submitted using the CloneSpec.

Return 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

You can use the `getMoref` method to obtain the managed object reference of the requested object based on name and type.

The `getMoref` method 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

Name	Type	Value	Description
name	String		Name of object to look for.
type	String		Managed object type.
requestSpec	String		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>
```

Returns

Type	Value	Description
String		Managed object reference in string format.

Return 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>
```

Provisioning and Cloning specifications and messages

The Provisioning and Cloning API provides several specifications and messages.

RequestSpec

The `RequestSpec` 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

Type	Value	Description
serviceUrl	String	URL for the VMware vCenter SDK
vcUser	String	VMware vCenter username (null ok if using vcSession)

Type	Value	Description
vcPassword	String	VMware vCenter password (null ok if using vcSession)
vcSession	String	VMware session (null ok if using vcUser/vcPassword)
cloneSpec	Object `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.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 `CloneSpec` specification describes a request to create clones of a virtual machine or template or to redeploy the virtual hard drives.

When used with the `redeployVMs` method, each virtual machine 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 redeployed and the `VmSpec` describes this virtual machine.

When `CloneSpec` is used with the `createClones` method, a new virtual machine is created for each entry in the clones map. 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 virtual machine or template. This list can be used to specify different destinations for each file as well as to create new datastores.

Type	**Value**	**Description**
templateMoref	String	Source VM or template of cloning operation. String representation of type and value of ManagedObjectReference from VMware VI API.
snapshotMoref	String	The managed object reference for a snapshot of the source virtual machine to base the clones on.
containerMoref	String	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	Virtual machine folder the clones should be created in. If `null`, clones are created at the root virtual machine folder.
vmTransform	String	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`.

Type	Value	Description
hardwareVersion	String	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 virtual machine name to virtual machine specification (VmSpec).
files	List <VmFileSpec>	List of files that make up source virtual machine or template specified in templateMoref.
memMB	Long	Override the source virtual machine (or template) amount of memory during cloning process. Value is in MB.
numberCPU	Int	Override the source virtual machine (or template) number of CPUs during cloning process.

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.netapp.com/}vmSpec"
minOccurs="0"/>
                                                        </sequence>
                                                    </restriction>
                                                </complexContent>
                                            </complexType>
                                        </element>
                                    </sequence>
                                </restriction>
                            </complexContent>
                        </complexType>
                    </element>
                    <element name="connBroker" type="{http://
server.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"/>
                    <element name="memMB" type="{http://www.w3.org/2001/
XMLSchema}long" minOccurs="0"/>
                    <element name="numberCPU" type="{http://www.w3.org/2001/
XMLSchema}int" minOccurs="0"/>
                </sequence>
            </restriction>
        </complexContent>
        </complexType>
```

VmFileSpec

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

Properties

Type	Value	Description
sourcePath	String	Path to `vmx` or `vmdk` file. The string Configuration File can be passed in place of an actual `vmx` file.
destDatastoreSpec	DatastoreSpec	Destination datastore specification.

XML

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

DatastoreSpec

The `DatastoreSpec` specification 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

Type	Value	Description
mor	String	Destination datastore. String representation of type and value of `ManagedObjectReference` from the VMware VI API.
targetMor	String	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	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	The protocsl being used. Valid values are NFS, FCP, iSCSI.
containerName	String	Name of the aggregate for new NFS datastores, or name of volume for new VMFS datastores.
sizeInMB	Long	Size of the datastore in MB. An additional 256 MB is added for VMFS datastores to cover metadata overhead.
thinProvision	Boolean	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	If `true`, the volAutoGrowInc and volAutoGrowMax values are applied to the volume.

Type	Value	Description
volAutoGrowInc	Long	Increment in which to grow volume automatically as needed in MB.
volAutoGrowMax	Long	Maximum size to which to grow the volume automatically in MB.
datastoreNames	List <String>	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	Number of datastores. This should indicate the size of the list of names in datastoreNames.
blockSize	Int	VMFS block size in MB.
controller	ControllerSpec	The controller.
wrapperVol	Boolean	When `true`, new volume is created to contain the new LUN to be used for a new VMFS datastore (containerName must contain aggregate name if `true`). When `true`, the volume containing the LUN (VMFS datastore) will be resized to make room for the new size of the LUN (if required).

Required parameters

Some actions require the use of multiple parameters.

Specifying an existing datastore

- mor
- controller

Specifying new NFS datastores using createClones or createDatastore

- targetMor - only required for `createDatastore`
- containerName
- 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
- containerName
- 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`
- containerName
- 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

- containerName
- 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="containerName" 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

The `ControllerSpec` specification describes the controller connection and authentication data. This information is used by the Provisioning and Cloning capability to connect to the controller using the ZAPI interface. No other protocol is used to connect to the controller.

Properties

Type	Value	Description
ipAddress	String	IP or host name of the controller.
username	String	User name (does not need to be root).
password	String	Password.
ssl	Boolean	If `true`, use HTTPS. If `false`, use HTTP to connect to the controller.
passthroughContext	String (optional)	Name of the vFiler or Vserver on which to create the new storage.
actuallyOpsMgr	Boolean (optional)	If `true`, connects to Operations Manager. If `false`, connects to 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

The `VmSpec` specification describes configuration and action options for each virtual machine created.

The MAC address must be in a range defined by VMware. Refer to VMware documentation for more information.

Properties

Type	Value	Description
macAddress	Map <String, String>	Virtual network adapter to MAC address information (optional).
custSpecName	String	Name of guest customization specification to be applied (optional).
vmMoref	String	The managed object reference of the virtual machine to be redeployed.
powerOn	Boolean	If `true`, the new virtual machines 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

The `GuestCustomizationSpec` specification identifies the guest customization specification.

Properties

Type	Value	Description
name	String	Name of the guest customization specification.
useVmName	Boolean	If guest customization specification is of type `CustomizationSysprepText`, this option can be used to make the guest hostname match the virtual machine name. **Note:** It is the responsibility of the implementer to ensure that the virtual machine name results 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>
```

ConnectionBrokerSpec

The `ConnectionBrokerSpec` specification describes the connection broker and authentication data. This information is used by VSC for VMware vSphere to connect to the connection broker (VMware View or Citrix XenDesktop).

Properties

Type	Value	Description
type	ConnectionBrokerType	The type of connection broker to import into. This can be `VMWARE_VIEW_4_0`, `VMWARE_VIEW_4_5`, or `XEN_DESKTOP`.
host	String	Hostname or IP address of the connection broker (used only for View import).
username	String	A user who can access the View Server (used only for View import).
password	String	Password of the specified user (used only for View import).
domain	String	FQDN where the connection broker resides.
desktopType	DesktopType	`INDIVIDUAL_DESKTOP` or `DESKTOP_POOL` (used only for View import).

Type	Value	Description
accessMode	AccessMode	`PERSISTENT` or `NON_PERSISTENT`. This corresponds to dedicated and floating in View 4.5 and higher, respectively (used only for View import).
poolType	PoolType	`NEW` or `EXISTING`. Create a new pool or use an existing one (used only for View import).
poolName	String	The name of the new pool if the PoolType is set to `NEW` (used only for View import).

StatusMessage

You can use the `StatusMessage` specification to obtain the progress and status of an operation.

Properties

Type	Value	Description
id	Int	Operation identifier.
progress	Int	Valid values are 0-100. Indicates how much of the copy or clone process has completed at the time of the query.
status	String	Valid values are `complete` (finished without error), `failed` (finished with error), or `running` (operation in progress).
reason	String	If the status is `failed`, this contains 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>
```

Provisioning and Cloning sample code

```
<ns2:createClonesxmlns:ns2="http://server.kamino.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>
```

Provisioning and Cloning client-side programming

Various client-side programming environments allow you to access the SOAP service. You can use this service for your own client-side programming.

Accessing the SOAP API through Java

You can access the SOAP API by using Java-based tools or the Java programming language.

Generating a certificate for use with the wsimport tool

You must generate a certificate for use with `wsimport` so that the tool will be able to read the WSDL that is generated by the build process.

About this task

Complete the following steps to generate a certificate for use with `wsimport`.

Steps

1. Stop the NVPF service. This step is optional if you have already generated an SSL certificate for the environment.
2. Run the following command in the VSC for VMware vSphere installation directory: `c:\Program Files\NetApp\Virtual Storage Console>bin\vsc ssl setup -domain <domain`

   For <*domain*>, enter the host name of the system running VSC for VMware vSphere or a fully qualified domain name of the system running VSC for VMware vSphere.

   Note: This step is optional if you have already generated an SSL certificate for the environment.

3. From the VSC for VMware vSphere installation directory, change to `\etc` and run the command: `keytool -export -alias nvpf -keystore nvpf.keystore -file nvpf.cer`

   Note: If you moved the keystore file from the `c:\Program Files\NetApp\Virtual Storage Console\etc` directory, enter the path to the keystore file.

   This command creates a new file called `nvpf.cer`.

   This certificate will be imported to the local Java keystore. If you have the Java JRE version 1.6.0_21 installed in Program Files, the command to execute will look like this:

   ```
   c:\Program Files\NetApp\Virtual Storage Console\etc>keytool -import -
   alias nvpf -file nvpf.cer -keystore "c:\Program Files\Java
   \jdk1.6.0_21\jre\lib\security\cacerts"
   ```

4. Enter keystore password:

   `changeit`

5. Run `wsimport` to grab the WSDL and generate the Java classes to write your own client. Enter the commands:

   `cd \dev`

   `c :\dev\wsimport -verbose -s . -p com.netapp.kamino.api https://<domain>:8143/kamino/public/api?wsdl`

   Be sure to change <*domain*> to a valid host or domain name. If that name does not resolve through DNS, add it to your hosts file (not localhost).

Accessing SOAP through C#

To begin using the SOAP API, you must first add the web reference to your project.

After the web service reference has been added, you can start accessing the client side objects to make API calls into the SOAP service.

Copyright information

Trademark information

How to send your comments

You can help us to improve the quality of our documentation by sending us your feedback.

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have suggestions for improving this document, send us your comments by email to *doccomments@netapp.com*. To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277

Index

W

X

Z