



Technical Report

## Using FlexClone to clone Files and LUNs

Shree Reddy, NetApp  
September 2011 | TR-3742

### **FLEXCLONE ENHANCEMENTS TO ENABLE FILE AND LUN CLONING**

This document will cover the enhancements made to the FlexClone® functionality at the file and LUN levels in Data ONTAP® 8.1.

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
<b>2</b>	<b>CONFIGURATION AND OPERATION .....</b>	<b>6</b>
2.1	PLATFORMS.....	6
2.2	DATA ONTAP.....	6
2.3	LICENSING.....	7
2.4	CREATING A FLEXCLONE INSTANCE OF A FILE IN A NAS ENVIRONMENT .....	8
2.5	CREATING A FLEXCLONE INSTANCE OF A LUN .....	8
2.6	CREATING A FLEXCLONE INSTANCE OF A FILE IN A SAN ENVIRONMENT .....	9
2.7	CREATING A FLEXCLONE INSTANCE FROM A SNAPSHOT COPY .....	9
2.8	DELETING THE SOURCE FILE OR LUN.....	10
2.9	OVERWRITING THE DATA IN THE SOURCE OR CLONE FILE OR LUN.....	10
2.10	VOLUME STATUS.....	10
2.11	VIEWING THE SPACE SAVINGS DUE TO FLEXCLONE FILE OR LUN .....	10
2.12	STATUS OF CLONE OPERATION.....	10
2.13	COMMAND SUMMARY .....	11
<b>3</b>	<b>BEHAVIOR OF FILE AND LUN CLONE .....</b>	<b>13</b>
3.1	SPACE RESERVATION .....	13
3.2	QUOTAS.....	13
3.3	ACCESS CONTROL LISTS (ACLs) AND STREAMS .....	13
3.4	ROLE-BASED ACCESS CONTROL (7-MODE ONLY).....	13
<b>4</b>	<b>RECOMMENDATIONS .....</b>	<b>13</b>
4.1	CREATING A LARGE NUMBER OF CLONE FILES OR LUNS .....	14
4.2	CREATING A LARGE NUMBER OF CLONE FILE OR LUNS IN THE SAME VOLUME .....	14
<b>5</b>	<b>INTEROPERABILITY WITH OTHER NETAPP PRODUCTS.....</b>	<b>14</b>
5.1	VOLUME SNAPMIRROR.....	14
5.2	QTREE SNAPMIRROR AND SNAPVAULT (7-MODE ONLY).....	15
5.3	NDMP AND DUMP .....	15
5.4	SYNCHRONOUS SNAPMIRROR (7-MODE ONLY) .....	16
5.5	DATA MOTION FOR VOLUMES (7-MODE AND CLUSTER-MODE) .....	16
5.6	SNAPSHOT .....	16
5.7	VOLUME OPERATIONS .....	16
5.8	MULTISTORE (7-MODE).....	17
5.9	DEDUPLICATION.....	17
5.10	REBOOT.....	17
5.11	CLUSTER FAILOVER .....	17
5.12	FLEXSHARE (7-MODE ONLY).....	17
5.13	SNAPLOCK .....	18
<b>6</b>	<b>PERFORMANCE.....</b>	<b>18</b>

<b>7</b>	<b>EXAMPLE USE CASE (7-MODE)</b> .....	<b>18</b>
<b>8</b>	<b>FLEXCLONE IN VIRTUALIZED ENVIRONMENTS</b> .....	<b>26</b>
<b>9</b>	<b>CONCLUSION</b> .....	<b>27</b>
<b>10</b>	<b>REFERENCES</b> .....	<b>28</b>
<b>10.1</b>	<b>NETAPP TECHNICAL REPORTS AND WHITE PAPERS</b> .....	<b>28</b>
<b>10.2</b>	<b>DATA ONTAP PRODUCT DOCUMENTATION</b> .....	<b>28</b>

**LIST OF TABLES**

Table 1)	Data ONTAP 8.1 enhancements .....	7
Table 2)	LUN clone support in various Data ONTAP versions .....	9
Table 3)	Comparison of LUN cloning methodologies .....	9
Table 4)	Command summary .....	11

**LIST OF FIGURES**

Figure 1)	Overview of FlexClone .....	5
Figure 2)	Combining FlexClone at FlexVol and file or LUN level to create multiple copies of the same file or LUN all based on the same physical storage. ....	6
Figure 3)	Volume SnapMirror transfer of FlexClone instance of file or LUN. ....	15
Figure 4)	SnapMirror transfer of FlexClone instance of file or LUN.....	15
Figure 5)	Volume mapped to the client using CIFS. ....	19
Figure 6)	Properties of the CIFS share.....	20
Figure 7)	Files as seen from the client.....	22
Figure 8)	Properties of the share. ....	22
Figure 9)	Properties of all the files. ....	23
Figure 10)	Properties of the share. ....	25
Figure 11)	Workflow of the VM cloning process .....	27

## 1 INTRODUCTION

NetApp® FlexClone technology was enhanced in Data ONTAP 7.3.1 to provide space-efficient cloning at different granularities. FlexClone technology now gives the user the ability to clone individual files that are present in a FlexVol® volume in a NAS environment or inside a LUN in a SAN environment. This is also a newer way to clone LUNs without the need for a backing Snapshot™ copy. The pre-8.1 implementation had certain limitations such as speed, scalability, asynchronous operation, and delay in availability of clones for read/write access. These limitations have been fixed in Data ONTAP 8.1 so that FlexClone volumes at the file/LUN level can now support a broader set of use cases. This document gives a detailed description of FlexClone at the file and LUN levels in Data ONTAP 8.1.

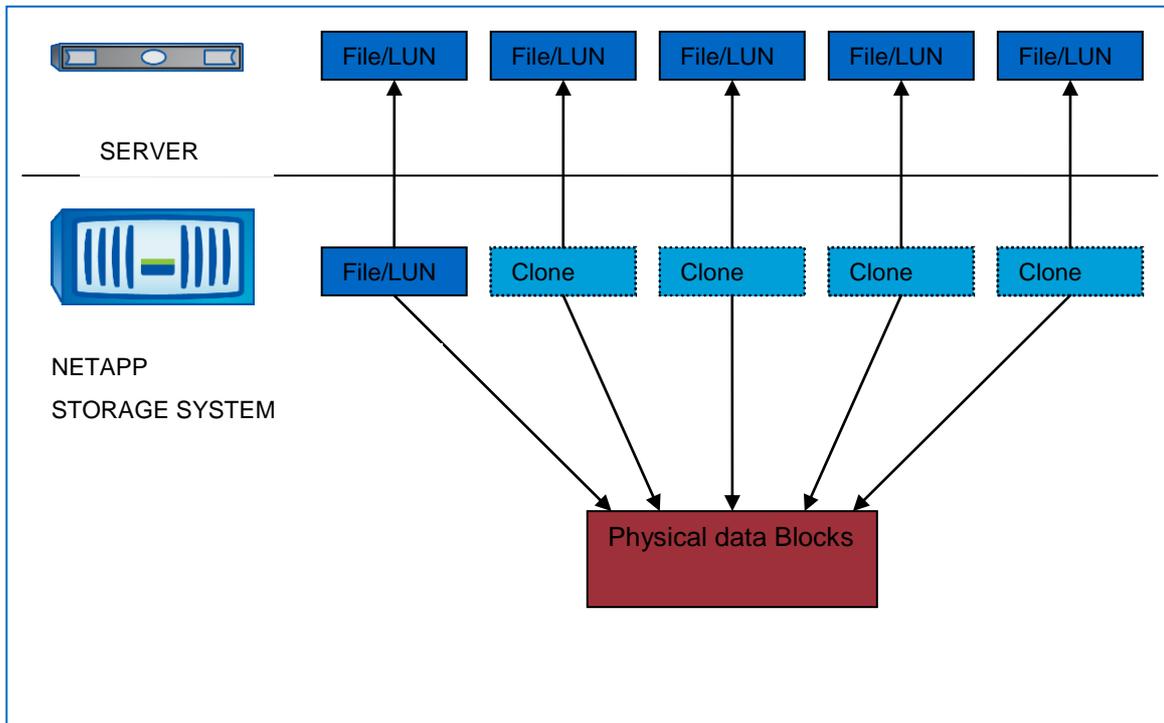
FlexClone at the FlexVol level has been in Data ONTAP since Data ONTAP 7.0. A FlexClone volume is a writable point-in-time image based on a Snapshot copy of a FlexVol volume or another FlexClone volume. FlexClone volumes use space very efficiently, leveraging the Data ONTAP architecture to store only data that changes between the source and clone. For more information on FlexClone at the FlexVol level, refer to NetApp TR-3347

Starting with Data ONTAP 7.3.1, you can create a clone of a file that is present either in a FlexVol volume in a NAS environment or inside a LUN in a SAN environment. Using the enhanced FlexClone technology, LUNs can now be cloned without the need of a backing Snapshot copy.

Cloning of files and LUNs using FlexClone technology is highly space efficient since the cloned copies share the same physical data space with the source and occupy negligible extra space in the storage system for their initial metadata. Cloned files or LUNs start occupying extra space only when the data in the source or the clone is overwritten. FlexClone creation is also a fast and time-efficient process since there is no physical copy of data involved. Cloning using FlexClone at file and LUN levels is Snapshot independent and so does not need a backing Snapshot copy.

The process of creating a clone of an existing LUN or file has no impact on the client access to the source file or LUN either during the creation process or after the cloning is done. Clients that are accessing the source file or LUN will experience no disruption or outage during the clone creation process and can write to the source file or LUN while the cloning process is in progress. The source file or LUN and the clone files or LUNs appear to the client systems as regular files or LUNs but share the physical storage space on the storage system.

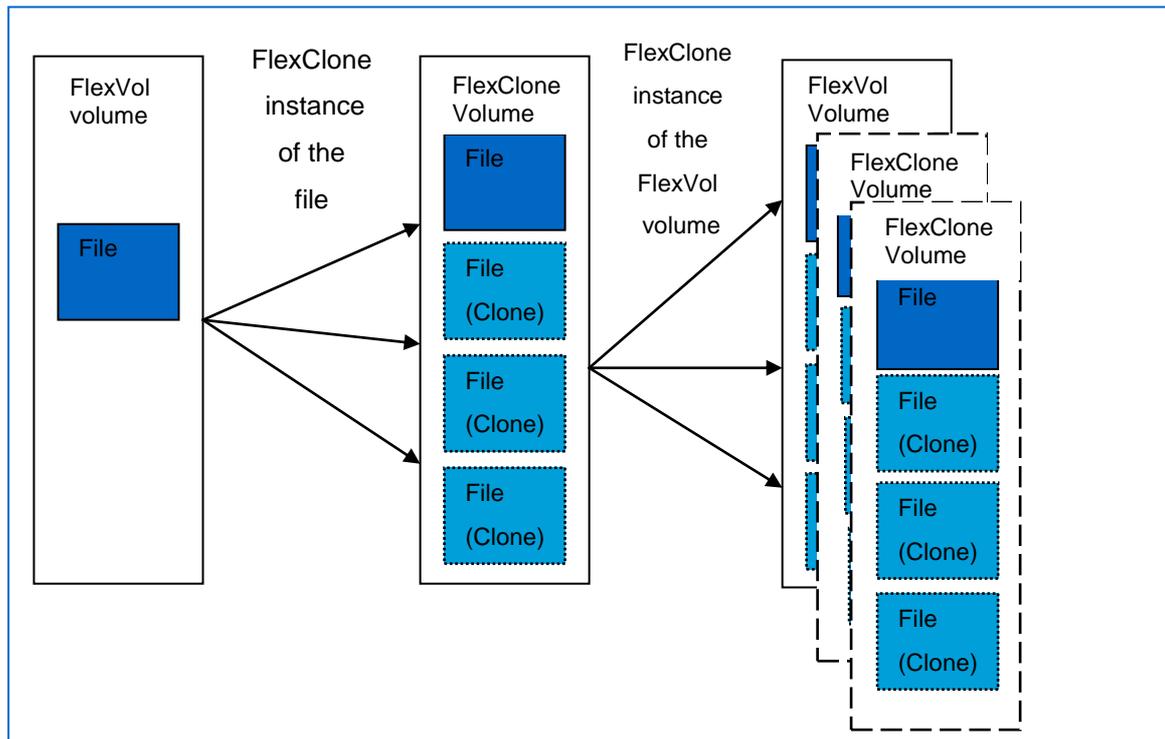
Figure 1) Overview of FlexClone



With the enhancements in Data ONTAP 8.1, the clone file or LUN can be accessed from the client as soon as the clone operation is triggered. The clients see the clone files or LUNs as regular files and LUNs. All regular file and LUN operations are supported on both the source and the clones. If a file or LUN has clone files or LUNs, either the source file or LUN or any of the clone files or LUNs can be deleted with no effect on the source file or the clone files.

The new cloning abilities combined with the existing functionality to create FlexClone copies of FlexVol volumes provide a space- and time-efficient solution for many data center problems that involve multiple copies of the same dataset and drastically reduce the storage space needed to store duplicate copies of data. FlexClone at the FlexVol volume, file, and LUN levels can be combined to create a powerful time- and space-efficient solution to store redundant datasets since all the redundant files or LUNs share the same underlying physical storage.

**Figure 2) Combining FlexClone at FlexVol and file or LUN level to create multiple copies of the same file or LUN all based on the same physical storage.**



As shown in Figure 2, we can use FlexClone at the file or LUN level to create clone files or LUNs within the same FlexVol volume, all of which share the same underlying storage. Then we can use FlexClone at the FlexVol level to create FlexClone volumes of the parent FlexVol volume. The FlexClone volumes share the data blocks with the parent FlexVol volume. So, we have lots of copies of the same file or LUN all sharing the same underlying physical storage. These clone files or LUNs start occupying extra space only when data in them is overwritten.

## 2 CONFIGURATION AND OPERATION

This section gives an overview of the requirements for enabling FlexClone at the file and LUN levels and the process of creating a file or LUN clone on a NetApp storage system.

### 2.1 PLATFORMS

FlexClone at the file and LUN granularities is supported on all the platforms supported in Data ONTAP 8.1.

### 2.2 DATA ONTAP

The enhanced ability of FlexClone to clone files and LUNs without the need for a pinned backing Snapshot copy is available in Data ONTAP 7.3.1 and later. Data ONTAP 8.1 introduces significant enhancements to the existing FlexClone implementation.

Table 1) Data ONTAP 8.1 enhancements

	Data ONTAP 7.3.x and Data ONTAP 8.0.x	Data ONTAP 8.1
<b>Maximum clone count per image</b>	255	32,767
<b>Clone creation time</b>	Dependent on size of clone	Constant time; not dependent on size of clone  (Time taken for sub-file/LUN cloning still depends on the size of the range being cloned. There are no constant time guarantees for it.)
<b>Read/write access to clone</b>	Delayed access to clone till clone operation is complete	Clone is available immediately while the cloning operation completes in the background
<b>API</b>	Asynchronous	Synchronous
<b>Scalability</b>	Not scalable; each clone requires a WAFL® consistency point	No dependency on a WAFL consistency point per operation
<b>Number of simultaneous clone operations</b>	16	No limit
<b>7-Mode/Cluster-Mode</b>	7-Mode only	Both modes supported
<b>Maximum shared limit for FlexClone data</b>	Same as dedupe limit	640TB*
<b>Single file SnapRestore® operation during a FlexClone operation</b>	Not allowed	Allowed
<b>Ability to create a clone from a Snapshot copy</b>	No	Yes

\* **Note:** If a user backs up data in a volume with such high shared limits to tape, the user will not be able to restore it to a single volume since it will exceed the volume size limit. The same is true for other logical backup mechanisms like SnapVault/QSM. Also any system process (for example: compression/deduplication processes, splitting a cloned volume) that needs to backup or process this much data may be impacted. In general, NetApp recommends using multiple volumes instead of putting such high amounts of shared data in one volume for better performance.

## 2.3 LICENSING

The enhanced ability of FlexClone to clone files and LUNs without the need for a backing Snapshot copy, along with the existing ability to create FlexClone volume of a FlexVol volume is available under

the FlexClone feature license. You need to license `flex_clone` to enable the FlexClone feature on the storage system using the following command:

```
license add <<flex_clone license key>>
```

In an HA pair environment, you need to add the license on both the nodes to enable the feature on both of them.

If you already have a FlexClone license and are running an older version of Data ONTAP, then upgrading to a Data ONTAP version that supports FlexClone at file and LUN levels will give you access to this enhanced functionality.

## 2.4 CREATING A FLEXCLONE INSTANCE OF A FILE IN A NAS ENVIRONMENT

FlexClone at the file level can be used to create clones of individual files stored in a FlexVol volume in a NetApp storage system for both NFS and CIFS environments. The clone file must reside in the same FlexVol volume as the source file. All the cloned files refer to the same physical data blocks on the disk. There is a minimal amount of space overhead for each clone for storing its metadata. A FlexClone instance of a file can be created in a NAS environment with the following steps:

- Issue a `clone start (7-Mode)/volume file clone create (Cluster-Mode)` command with source and destination file paths. The source and destination paths must be in the same FlexVol volume. The clone operation is synchronous, and the clone is available for reads and writes as soon as the command returns. The source file can be accessed by the client and can be written to while the cloning operation is in progress.
- In order to create a point-in-time image of the source file or LUN, clone command by default creates a temporary hidden copy of the source file/LUN. This is done so that the clone is a point-in-time image of the source file or LUN and that any active writes to the source file do not affect the process of clone creation. The clone file is now ready to be used by NFS or CIFS clients. The clients see the clone and source as separate files. These source and clone files can also be accessed by any of the data access protocols that can access files, including http, ftp, and NDMP.
- The sharing process completes in the background. Note that there is no interruption to any read, write, or other operations on the source file while a clone operation is in progress.

## 2.5 CREATING A FLEXCLONE INSTANCE OF A LUN

FlexClone can be used to clone a complete LUN that is present in a FlexVol volume on the storage system. The process to clone a LUN is similar to cloning a file. The clone LUN must reside in the same FlexVol volume as the source LUN. Data ONTAP has the ability to recognize the source as a LUN and takes the necessary steps to clone a LUN. Though the process to share physical blocks between a LUN and its clone is similar to file cloning, cloning a LUN involves extra metadata operations that must be performed to present the LUN to the SAN subsystem of Data ONTAP. Data ONTAP automatically performs these operations in the background when the source is a LUN. Also, a FlexClone instance of a LUN requires slightly more space for its metadata due to the SAN subsystem-related metadata. Once the operation is triggered, the clone LUN is ready to be presented to a SAN client.

Note that Data ONTAP has had the ability to clone a LUN using a backing Snapshot copy since Data ONTAP 7.0. This method of creating a LUN clone still exists and is supported in Data ONTAP 8.1 operating in 7-Mode. Data ONTAP 8.1 operating in Cluster-Mode does not support cloning a LUN clone using a backing Snapshot copy.

Using FlexClone to clone a LUN is an enhancement that was introduced in Data ONTAP 7.3.1 and does not require the backing snapshot to be pinned. This method continues to be supported in both modes in Data ONTAP 8.1.

Table 2) LUN clone support in various Data ONTAP versions

LUN Clone Method	Data ONTAP 7.3	Data ONTAP 8.0.x 7-Mode	Data ONTAP 8.0.x Cluster-Mode	Data ONTAP 8.1 7-Mode	Data ONTAP 8.1 Cluster-Mode
Using a backing Snapshot copy	Yes	Yes	No	Yes	No
Using FlexClone	Yes	Yes	No	Yes	Yes

Table 3) Comparison of LUN cloning methodologies

	LUN Clone Using Backing Snapshot Copy	LUN Clone Using FlexClone
Command	<code>lun clone create</code>	<code>clone start</code>
Dependency on a backing Snapshot copy	Yes	No
Uses block-sharing infrastructure	No	Yes
Clone needs to be split explicitly	Yes; clone will share blocks with the source till a split is triggered using the <code>lun clone split</code> command	No; clone blocks are shared with source using a background process.

## 2.6 CREATING A FLEXCLONE INSTANCE OF A FILE IN A SAN ENVIRONMENT

FlexClone can be used to clone files that are present inside a LUN in a SAN environment. Data ONTAP provides the API to be able to do this. However, host-side support is needed to integrate the clone file into the host file system and to make the clone file usable by the client. The API can be used to create a clone of a file on the storage system as follows:

- To clone a file that is present inside a LUN, the user needs to specify the address of the data blocks of the file in logical block addressing (LBA) format as seen by the host. The `-r` flag of the `clone start (7-Mode)/volume file clone create (Cluster-Mode)` command must be used to specify the source and destination LBA addresses and the count of number of data blocks to be cloned.
- The rest of the process of clone creation is the same as the process outlined in the NAS section.
- Host-side tool support is needed to integrate the cloned data blocks into the host file system and to present the cloned file to the host file system.

## 2.7 CREATING A FLEXCLONE INSTANCE FROM A SNAPSHOT COPY

Starting with Data ONTAP 8.1, FlexClone can now be used to clone files/LUNs present in a Snapshot copy. When cloning from a backing Snapshot copy, new snapshot copies are disallowed till the background cloning process is complete.

When cloning from a Snapshot copy, the backing Snapshot copy is locked till the clone operation is completed. Once the clone operation is complete, the clone is no longer dependent on the backing Snapshot copy. If the Snapshot copy gets destroyed due to snapshot auto-delete, access to the dependent clone will be denied.

## 2.8 DELETING THE SOURCE FILE OR LUN

Once clone files or LUNs have been created, the source file or LUN or any of the clones can be deleted. Deleting the source file or LUN has no impact on the clone files or LUNs. All the clone files or LUNs still exist and show up to the clients as normal files or LUNs. Similarly, deleting a clone file or LUN has no impact on the other clones or the source. The clone LUNs or files still exist and they still share the same physical data blocks on the disk.

When a file or LUN or its clone that has shared blocks is deleted, the shared blocks continue to be used by the remaining clone files or LUNs. So, deleting a clone will only free up the space that is being used by its metadata. No space used by the shared data blocks is returned to free space until the source file or LUN and all clone files or LUNs of that source are deleted. Once a source file or LUN and all its clone files or LUNs are deleted, the data blocks are freed and the space returned to the free space pool.

## 2.9 OVERWRITING THE DATA IN THE SOURCE OR CLONE FILE OR LUN

When a clone file or LUN is initially created, both the source and the clone share the same data blocks. Once data in either the source or the clone is overwritten with new data, more space will start to be consumed since the clone and the source do not share the newly written data. The newly written data is stored separately for each individual entity, the source and the clone. Note that even if the same data is written to both the source and the clone, they still will point to different blocks on disk and will not share the physical blocks. The physical blocks are only shared immediately after the cloning process. If the newly written data to the source and clone is similar, deduplication can be run on the FlexVol volume that contains the clone to get further space savings since deduplication can eliminate duplicate blocks.

## 2.10 VOLUME STATUS

The `vol status (7-Mode)/volume show (Cluster-Mode)` command of Data ONTAP shows the status of all volumes that contain file and LUN clones created using the clone command as having the attribute `sis`. The `sis` attribute is added to the FlexVol volume when the first FlexClone instance of a file or LUN is created. Note that volumes that have deduplication enabled on them also show up with the attribute `sis` in the output of `vol status (7-Mode)/volume show (Cluster-Mode)` command.

## 2.11 VIEWING THE SPACE SAVINGS DUE TO FLEXCLONE FILE OR LUN

The `df -s` command with the FlexVol volume name as argument will list how much space in the volume has been saved because the clone files or LUNs share the same physical blocks instead of each having their own physical copy of the blocks. Note that if deduplication is enabled on the same FlexVol volume that has FlexClone files or LUNs, the output of `df -s` will display the total saving due to the combination of both deduplication and FlexClone. The `df -s` command displays the space savings due to FlexClone as a percentage of the sum of the used and saved space on the FlexVol volume.

For example, if you have a FlexVol volume of size 100GB with 50GB used space and then create a file of 10GB and then create one clone of it, the total used physical space is approximately 60GB (50GB + 10GB for file and its clone). If the clone was a full physical copy, you would have used 70GB (50GB + 10GB for file + 10GB for the clone). So, we saved space of 10GB by cloning. So the savings are 14%  $((10/70)*100)$ .

## 2.12 STATUS OF CLONE OPERATION

With the improved FlexClone implementation in Data ONTAP 8.1, clone creation is instantaneous and constant in time. So, operations such as `clone status`, `clone stop`, or `clone clear` have been deprecated.

## 2.13 COMMAND SUMMARY

Here is a brief summary of the `clone` command and its subcommands. For a more detailed description of the commands, refer to the Data ONTAP documentation on the NetApp Support (formerly NOW®) site at <http://now.netapp.com>.

Table 4) Command summary

Command	Purpose
<p><b>7-Mode:</b></p> <pre>clone start &lt;src_path&gt; &lt;dest_path&gt; [-n]</pre> <p><b>Cluster-Mode:</b></p> <pre>volume file clone create          [- vserver &lt;vserver name&gt;] -volume &lt;volume name&gt; -source-path &lt;text&gt; [-destination-path &lt;text&gt;]</pre>	<p>Starts the process of creating a FlexClone instance of the file or LUN. The file or LUN at source indicated by the <code>src_path/source-path</code> argument is cloned to the destination path indicated by the <code>dest_path/destination-path</code> argument. The clone file or LUN must be in the same FlexVol volume as the source.</p> <p><b>NOTE:</b> The <code>-n</code> option has no effect in Data ONTAP 8.1 and is only supported for backward compatibility.</p>
<p><b>7-Mode:</b></p> <pre>clone start &lt;src_path&gt; [dest_path] &lt;-r &lt;src_fbn&gt;:&lt;dest_fbn&gt;:&lt;fbn_cnt&gt; ...&gt;</pre> <p><b>Cluster-Mode:</b></p> <pre>volume file clone create          [- vserver &lt;vserver name&gt;] -volume &lt;volume name&gt; -source-path &lt;text&gt; [-destination-path &lt;text&gt;] { [- range -r} &lt;src_fbn&gt;:&lt;dest_fbn&gt;:&lt;fbn_cnt&gt; ...] }</pre>	<p>Starts the clone process for cloning a file present inside a LUN. For cloning a file inside a LUN, the <code>src_fbn</code> and <code>dest_fbn</code> are the LBA addresses of the data blocks in the source file and the destination address where the blocks will be referenced. The <code>fbn_cnt</code> parameter is the count of number of blocks starting with the source LBA address that need to be cloned. The LBA addresses that are used here are the LBA addresses within the LUN as seen by the host.</p> <p>Multiple ranges can be specified with a single command. If <code>dest_path</code> is not specified, it is always considered to be a clone creation request from one block range to another within the same LUN (block ranges must be specified in this case).</p> <p>This can be used to create clones from LUN/sub-LUN to a regular file or from a regular file to a LUN/sub-LUN.</p> <p>If no range triplets are specified, it is considered as full file clone.</p> <p>For regular files, the blocks are considered as FBNs and are 4k in size. For LUNs the blocks are considered as LBAs and are 512B in size. In case of clone creation from LUN/File to file/LUN the ranges must be specified as LBA ranges.</p>

Command	Purpose
<p><b>7-Mode:</b></p> <pre>clone start &lt;src_path&gt; &lt;dest_path&gt; -s &lt;snapshot_name&gt;</pre> <p><b>Cluster-Mode:</b></p> <pre>volume file clone create          [- vserver &lt;vserver name&gt;] -volume &lt;volume name&gt; -source-path &lt;text&gt; [-destination-path &lt;text&gt;] [- snapshot-name -s &lt;snapshot name&gt;]</pre>	<p>The clone start/volume file clone create command can be used to create a clone from a source in a Snapshot copy as well. Snapshot name needs to be specified with the -s option. Sub-file/LUN clone from a specified Snapshot copy can also be created. If not specified, the clone source is always considered to be in the active file system.</p>
<p><b>7-Mode:</b></p> <pre>clone start &lt;src_path&gt; &lt;dest_path&gt; [-o]</pre> <p><b>Cluster-Mode:</b></p> <pre>volume file clone create          [- vserver &lt;vserver name&gt;] -volume &lt;volume name&gt; -source-path &lt;text&gt; [-destination-path &lt;text&gt;] [-no- reserve -o [true]]</pre>	<p>When invoked with the “-o” flag, the clone will not be space reserved.</p>
<p><b>7-Mode:</b></p> <pre>clone status [vol-name [ID]]</pre> <p><b>Cluster-Mode:</b></p> <p>N/A</p>	<p>Not supported in Data ONTAP 8.1</p>
<p><b>7-Mode:</b></p> <pre>clone stop &lt;vol-name&gt; &lt;ID&gt;</pre> <p><b>Cluster-Mode:</b></p> <p>N/A</p>	<p>Not supported in Data ONTAP 8.1</p>
<p><b>7-Mode:</b></p> <pre>clone clear &lt;vol-name&gt; &lt;ID&gt;</pre> <p><b>Cluster-Mode:</b></p> <p>N/A</p>	<p>Not supported in Data ONTAP 8.1</p>
<p><b>7-Mode:</b></p> <pre>man clone</pre> <p><b>Cluster-Mode:</b></p> <pre>man volume file clone</pre>	<p>Gives the man page for the clone command</p>

## 3 BEHAVIOR OF FILE AND LUN CLONE

This section discusses in detail how file or LUN clone behaves with different options of Data ONTAP.

### 3.1 SPACE RESERVATION

FlexClone for files/LUNs inherit the space guarantee settings of their source. So, if there is not enough space in the FlexVol volume to create a FlexClone instance of a file/LUN with space guarantee same as that of the source, the cloning process will fail. Note that the source and the clone will share blocks on the disk even with space guarantee enabled.

### 3.2 QUOTAS

Quota usage for clones is charged at the logical level. So, the amount of extra space usage charged to the quota for creating a clone is equal to the total logical size of the clone. For example, if you create a clone of a 10GB file, the total used space charged to the quota for the source file and the clone file is 20GB (10GB for the source and 10GB for the clone).

If the total logical used space occupied after a clone creation will be more than the allowed quota for that user or group or qtree, the clone operation will still succeed if the FlexVol volume has enough space to hold the metadata or the data for the clone. However, after the success of the clone operation, the quota for that user or group or qtree will be oversubscribed.

### 3.3 ACCESS CONTROL LISTS (ACLs) AND STREAMS

Starting with Data ONTAP 8.1, the clone will now inherit the access control lists of the parent file or LUN.

If the file that is the source of a FlexClone operation has streams, the clone operation will be disallowed.

### 3.4 ROLE-BASED ACCESS CONTROL (7-MODE ONLY)

The access to the clone command can be controlled by using role-based access control (RBAC) capabilities provided by Data ONTAP using the `useradmin` command. You can create roles that have access to only the commands that are needed to use the clone functionality. Here is an example set of commands to use to enable RBAC for clone command:

- Create a role that has access to all login methods but can use only the clone commands after logging in:  
`netapp01> useradmin role add cloneadmin -a login-*,cli-clone*,api-clone-*`
- Add a group that has the role of cloneadmin just created:  
`netapp01> useradmin group add cloneadmin_group -r cloneadmin`
- Add users to the group:  
`netapp01> useradmin user add cloneu1 -g cloneadmin_group`

The user, cloneu1 in this case, will have access only to the clone command of Data ONTAP. For more details on roles, groups, and users, refer to the Data ONTAP system administration guide on the NOW site at <http://now.netapp.com>.

## 4 RECOMMENDATIONS

This section outlines some of the recommended steps for creating a large number of clone LUNs or files while using the least amount of space. For detailed discussion around the feature related thresholds and recommendations for the FlexClone feature, refer to the “About FlexClone Files and FlexClone LUNs” section of the Data ONTAP storage management guide on NOW site at <http://now.netapp.com>.

## 4.1 CREATING A LARGE NUMBER OF CLONE FILES OR LUNS

To create a large number of clone files or LUNs on the storage system it is recommended to combine the different granularities of FlexClone copies rather than using the same granularity.

Suppose that you want to create 2,000 clone files from a single file. Here is how it can be achieved:

1. Create 99 FlexClone files from the source file for a total of 100 logical files within the same volume.
2. Create 19 FlexClone volumes of the FlexVol volume that the files reside in for a total of 20 FlexVol volumes. We now have 2,000 logical files all sharing the same physical storage.

## 4.2 CREATING A LARGE NUMBER OF CLONE FILE OR LUNS IN THE SAME VOLUME

The maximum number of times a block can be cloned is 32,767. All clones of the block beyond that will each be a physical copy. To minimize the space consumed by the clone files, you can follow this process:

1. Create 32,767 clone files or LUNs.
2. Then create the 32,768th clone file or LUN. This clone will be a full physical copy of the source file or LUN.
3. Now, create the clone files or LUNs from the 32,769th clone onward using the 32,768th clone file or LUN created in the previous step as the source until you create 32,767 more clone files or LUNs. In this way you can create 32,767 more clone files or LUNs based on the one physical copy created in the previous step. So, you have 65,536 logical files or LUNs all sharing two physical on disk copies. Repeat this process till you reach your target file or LUN clone count.

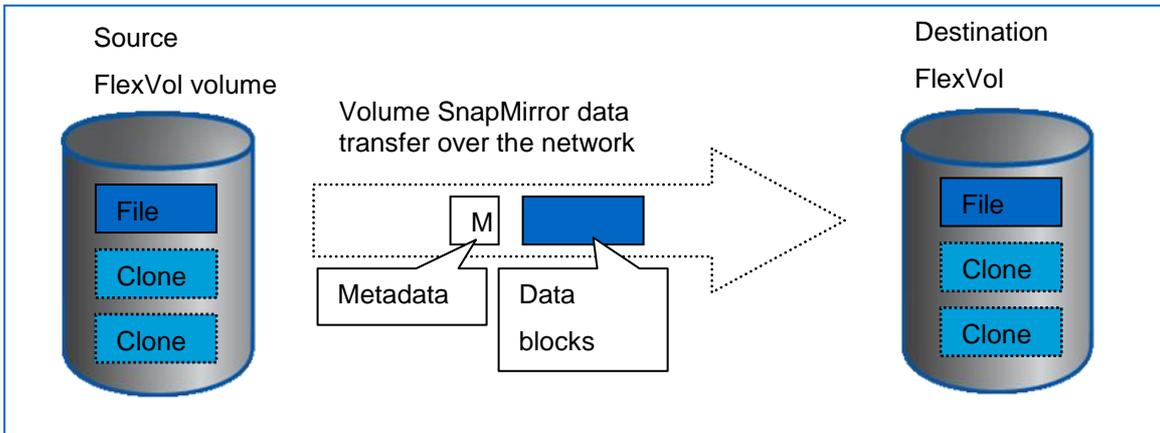
One important issue to consider when planning for environments with a large number of FlexClone files or LUNs is the performance requirements of the host-side application and the performance expected from the storage system. The performance section of this document has a discussion about the performance characteristics of FlexClone at file and LUN levels.

# 5 INTEROPERABILITY WITH OTHER NETAPP PRODUCTS

## 5.1 VOLUME SNAPMIRROR

Volume SnapMirror® is an efficient data replication product that transfers only the modified data disk blocks to the destination after the initial base transfer. Volume SnapMirror transfers data in 4KB disk blocks to the destination. If a FlexVol volume that is the source of a volume SnapMirror copy contains FlexClone files or LUNs, then volume SnapMirror transfers only the physical block and a small amount of metadata as shown in Figure 3. On the destination also only one copy of the physical block along with its metadata is stored, and the block is shared among the source and its clones. So, the destination volume will look exactly like the source volume, and all the clone files or LUNs on the destination will share the same physical blocks, shown in Figure 3.

Figure 3) Volume SnapMirror transfer of FlexClone instance of file or LUN.



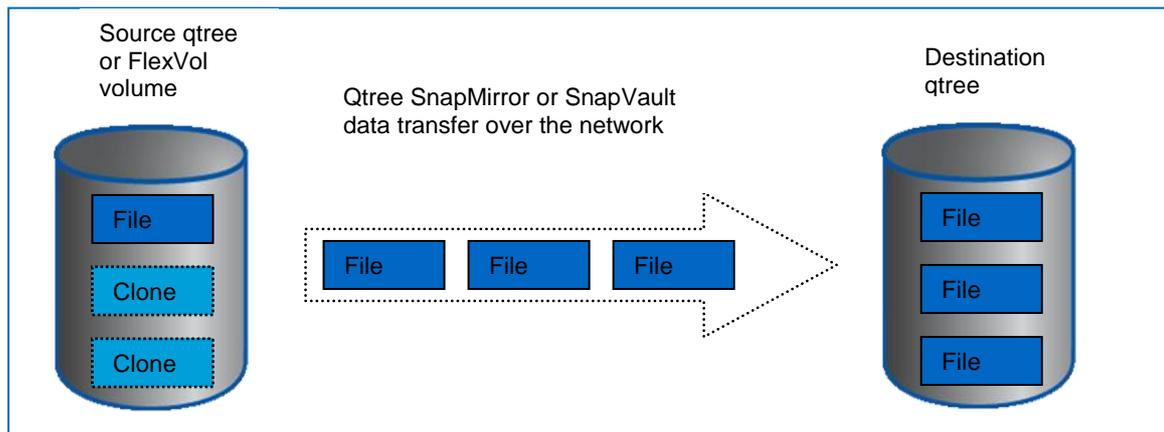
Volume SnapMirror is efficient and conserves network bandwidth as well as the storage space on the destination when the source has FlexClone instances of files or LUNs.

## 5.2 QTREE SNAPMIRROR AND SNAPVAULT (7-MODE ONLY)

Qtree SnapMirror enables replication of the source volume or qtree to a destination qtree. SnapVault® is a product for protecting data against loss and preserving old versions of data. SnapVault replicates data in primary system paths to qtrees on a SnapVault secondary storage system. SnapVault uses the qtree replication engine to perform replication.

Qtree SnapMirror and SnapVault all work at the logical file level. So, these features are not aware of the fact that the logical files, which are clones, share the same physical blocks. Hence all the clone files and LUNs are transferred to the destination as different physical files and LUNs and are stored on the destination as different files and LUNs. So, no block sharing exists on the destination as all clones are stored as separate files and they occupy space for their own data blocks, as shown in Figure 4.

Figure 4) SnapMirror transfer of FlexClone instance of file or LUN.



The destination size needs to be equal to the sum of the sizes of all logical files and LUNs present on the source. One way to get an estimate of the required size of the destination volume is to run the `df` command with the volume name as argument to figure out the used and saved space. The size of destination must be equal to or greater than the sum of used and saved space. Running deduplication on the destination after the SnapMirror transfer is complete will reduce the amount of used space on the destination to be almost the same as the source.

## 5.3 NDMP AND DUMP

Dump and NDMP also work at the logical level, just like qtree SnapMirror. So, when you back up a FlexVol volume or qtree, each clone will be backed up as an individual file. When restoring back, they are restored as individual files.

If dump is triggered while a clone operation is in progress, the dump Snapshot copy contains a partially cloned file. The file appears to dump as a partial file, and dump is capable of managing such files.

## 5.4 SYNCHRONOUS SNAPMIRROR (7-MODE ONLY)

A FlexVol volume that contains a FlexClone instance of a file or LUN that was created using the clone command should not be used as the source of a synchronous SnapMirror relationship. Data ONTAP does not prevent you from establishing a synchronous SnapMirror relationship using a FlexVol volume that has a FlexClone instance of a file or a LUN as the source. However, this is not a supported configuration and so should not be attempted.

## 5.5 DATA MOTION FOR VOLUMES (7-MODE AND CLUSTER-MODE)

FlexClone for files and LUNs cannot be created during the cutover phase of the `vol move` operation.

## 5.6 SNAPSHOT

All regular Snapshot operations are supported on a FlexVol volume that has FlexClone instances of files or LUNs in it. If a Snapshot copy is created while a clone operation in progress, the partially cloned file will get locked in the Snapshot copy. Cloning process will complete, and a full FlexClone instance of the file will be available for use after the cloning process is complete.

The partial file shows up in the Snapshot copy because the Snapshot copy was created while the clone operation was in progress. Having this partially cloned file in the Snapshot copy has no impact on either the source file or the clone files that are present in the active file system.

In Data ONTAP 8.1, FlexClone can be used to create a clone of a file or LUN that is present in a Snapshot copy using the “-s” option.

## 5.7 VOLUME OPERATIONS

### VOL COPY

The source FlexVol volume for a `vol copy` operation can have FlexClone files or LUNs in it. Once the vol copy operation completes, the destination FlexVol volume will also have the attribute “sis” attached to it and will show up in the output of the `vol status` command. The file or LUN and its clones on the destination FlexVol volume will share the data blocks just like they did on the source FlexVol volume.

### VOL AUTOSIZE

A FlexClone volume of a file or LUN will occupy some space on the physical storage for storing the metadata. If the FlexVol volume that has `vol autosize` configured on it and has a creation of a FlexClone instance of a file or LUN in progress on it and the FlexVol volume runs out of space in the middle of creating the metadata required for the clone, the autosize operation will not be activated. So, the size of the FlexVol volume will not be increased by autosize in this case even though the FlexVol volume is full. The clone operation will fail.

When enabling `vol autosize` on a FlexVol volume that has a FlexClone instance of a file or LUN in it, the maximum size limit for autosize operation must be less than the maximum allowed volume size for that platform. If the limit is more than the maximum allowed size, autosize will not be enabled, and an error will be generated.

### VOL CLONE

A FlexClone volume can be created using a FlexVol volume that has a FlexClone instance of a file or LUN in it as its parent. The clone volume will contain both the file or LUN and its clone. The source and clone files or LUNs that are present in the clone volume will continue to share blocks just like they share blocks in the parent volume. In fact, the file or LUN and its clones that are present in the parent volume and the file or LUN and its clones that are present in the clone volume all share the same

underlying physical data blocks, thus minimizing physical disk space usage. If the clone volume is split from its parent, then the file or LUN and its clones will not share blocks in the child FlexVol volume that was split and so exist as separate files or LUNs. So, the amount of used space in the child FlexVol volume that was split will be greater than what the used space was before the split operation.

A FlexClone instance of a file or LUN cannot be created on a FlexClone volume that is currently being split from its parent. Once the split operation is completed, then the FlexClone instance of a file or LUN can be created on the FlexVol volume that was split.

## 5.8 MULTISTORE (7-MODE)

MultiStore® is a feature of Data ONTAP that enables you to partition the storage and network resources of a single storage system so that it appears as multiple storage systems on the network. Each storage system created as a result of the partitioning is called a vFiler® unit. For details, see the MultiStore management guide on the NOW site at <http://now.netapp.com>.

In Data ONTAP 7.3.3 and later, the clone command will work in any vFiler context. For more details on using the clone command in a vFiler context, refer to the “About FlexClone Files and FlexClone LUNs” section of the Data ONTAP storage management guide on the NOW site at <http://now.netapp.com>. When the clone creation request is issued from:

- A non-default vFiler unit, both the source and destination should be owned by the specified vFiler unit.
- Default vFiler unit, source and destination can be in different vFiler units.

In Data ONTAP 7.3.2 and earlier, the clone command does not work in a vFiler context. So, it can only be used in the default vFiler instance, which is vfiler0.

## 5.9 DEDUPLICATION

A FlexClone instance of a file or a LUN can be created in a FlexVol volume that has deduplication enabled on it. Deduplication shares the physical blocks among different logical files or LUNs. So when creating a FlexClone instance of a file or LUN on a volume that had deduplication already run on it, you might reach the maximum block sharing limit of 32,767 before you create 32,767 FlexClone files or LUNs, as the physical block might already be shared during the deduplication process.

A FlexClone operation cannot be performed on FlexVol volume that has a `sis undo` operation currently running on it.

## 5.10 REBOOT

If the storage appliance is rebooted while a creation of a FlexClone instance of a file or LUN is in progress, the FlexClone operation is automatically restarted after the storage system boots up.

## 5.11 CLUSTER FAILOVER

In case of an HA pair, if a takeover happens and there is a creation of a FlexClone instance of a file or LUN running on the system that is being taken over, the clone operation is terminated prior to the takeover and then automatically restarted after the takeover is complete. Similarly, any clone operations in progress are terminated and automatically restarted after a giveback.

## 5.12 FLEXSHARE (7-MODE ONLY)

FlexShare® software is a powerful quality-of-service tool for Data ONTAP storage systems. It lets you assign individual priorities to multiple application workloads consolidated on a single system. The workload generated by creating or deleting a FlexClone instance of a file or LUN is treated as system workload by FlexShare. FlexShare can be used to set a priority to this workload generated by the

FlexClone volume, and so its impact on the storage system can be varied according to the priority set for system operations in FlexShare.

### 5.13 SNAPLOCK

FlexClone at the file and LUN levels is not supported on a SnapLock® volume.

## 6 PERFORMANCE

FlexClone at file and LUN levels is designed to be very efficient as it leverages the internal structure and flexibility of WAFL. In most cases, the storage system performance will not be affected due to the FlexClone operation. However, under heavy load and for certain types of workloads that are write intensive on the clone or its source and are highly random in nature, FlexClone at file and LUN levels might cause some performance impact on the storage system performance for writes to source or its clone. This is because the storage system has to adjust the metadata for the shared blocks being overwritten.

## 7 EXAMPLE USE CASE (7-MODE)

For the purpose of illustrating the ease and efficiency of using FlexClone at file and LUN levels, this section walks through a hypothetical use case that involves the complete lifecycle of FlexClone at the file level.

Let's say there is a company that specializes in developing image-processing software for huge size images such as satellite and space images. Its researchers are continuously working on creating newer and better image-processing algorithms for image enhancements that are focused on specific parts of the images that are bad or not clear enough. Since each researcher has a separate algorithm to test on the images, each of them needs a private copy of these huge image files. Since the enhancement algorithms are applied to specific and small parts of the images that are not clear, the changes in the underlying data for these different images are small compared to the image size. Having so many copies of these huge image files eats up a lot of space, and since most of the image files are similar, storage space is wasted for storing redundant data.

This is where the storage efficiency of FlexClone at file level is put to best use. For illustration purposes, let's assume that that the file is approximately 10GB in size.

Also, let's assume for illustration purposes that the same volume is mounted to a Linux® client using NFS and also is mapped to a Windows® client using CIFS. This is possible because of the multiprotocol support offered by NetApp storage systems.

Since there is a 10GB file in the volume, here is what the df command reports on the storage appliance:

```
netapp01> vol status testvol
```

Volume	State	Status	Options
testvol	online	raid_dp, flex	

Containing aggregate: 'testaggr'

```
netapp01>
```

```
netapp01> df testvol
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/vol/testvol/	83886080	<b>10509416</b>	73376664	13%	
/vol/testvol/.snapshot	20971520	0	20971520	0%	

The df command shows that at volume has approximately 10GB of used space, which is consistent with the fact that we have a 10GB file.

Since the volume is mounted on the Linux client using NFS, here is the output of the ls, df, and du commands from the client:

```
netapph1%ls -lt
total 10506376
-rw-rw-rw- 1 root root 10737418244 Oct  6 15:53 test_file
netapph1%
netapph1%df .
Filesystem            1K-blocks      Used Available Use% Mounted on
netapp01:/vol/testvol
                        83886080 10509728 73376352  13% /tmp/FlexClone
netapph1%.
netapph1%du -B 1K .
4      ./snapshot
10506384      .
```

The preceding commands show that the volume has one file of about 10GB. Note that df and du commands show approximately the same amount of used space.

Figure 5 is a screen capture from the Windows box that has the volume mapped using CIFS.

Figure 5) Volume mapped to the client using CIFS.

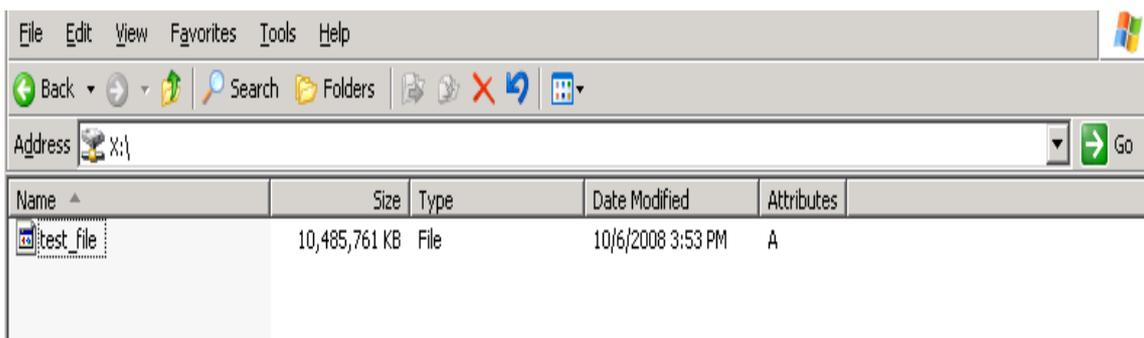
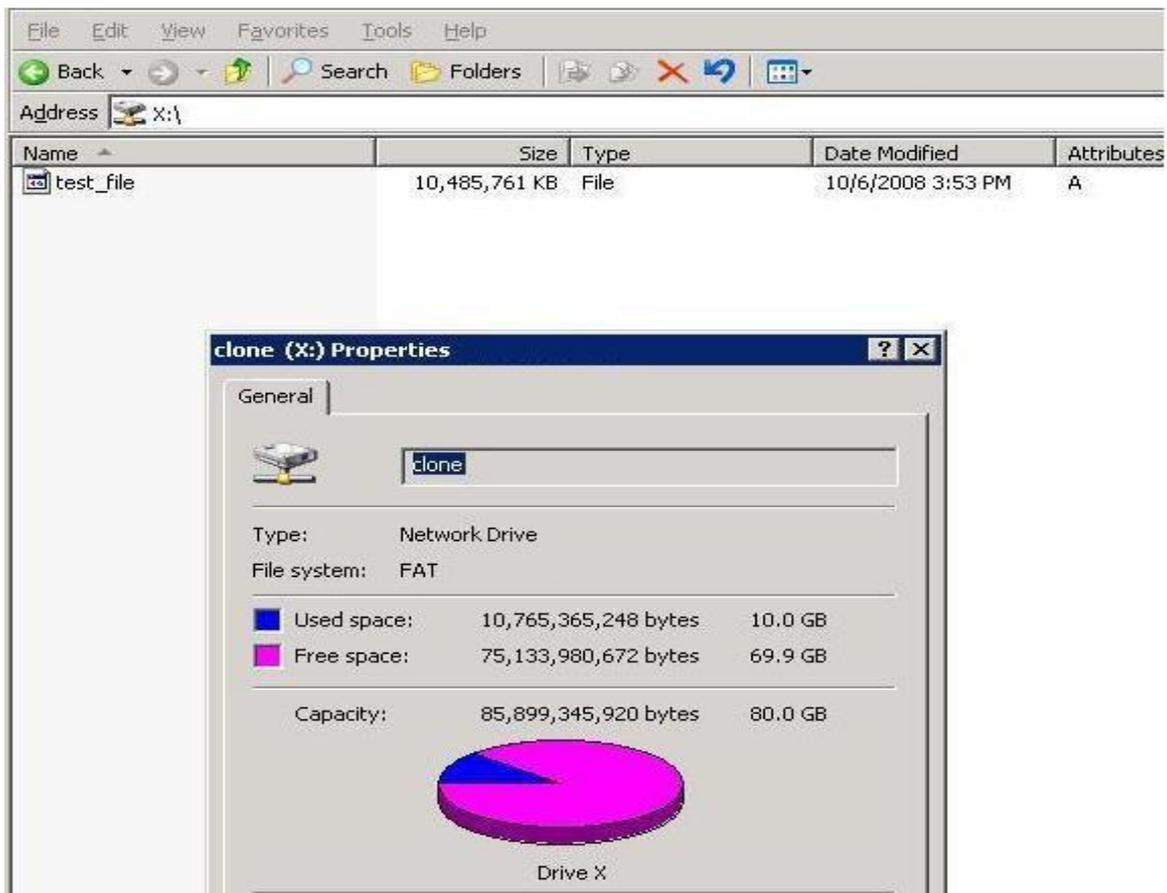


Figure 6 shows the properties of the share.

Figure 6) Properties of the CIFS share.



As you can see, the file size and used space show up as about 10GB, which is what is expected.

Now, suppose two researchers want to work on the same data in this file but would like to experiment with two different algorithms. So, instead of creating two copies of the file and using a total of 20GB in storage space, the storage admin decides to use the new FlexClone enhancements to create a clone of the file.

Here is what the admin does to create the clone:

Start the cloning process using the clone start command:

```
netapp01>
netapp01> clone start /vol/testvol/test_file /vol/testvol/clone_test_file
Clone operation started successfully. ID: 538.
```

As you see, the clone start command outputs the ID of the clone operation as soon as the cloning process begins, indicating the successful completion of the clone. So the storage system now has a 10GB file and one clone of the file. Here is what the df command shows on the storage system:

```

netapp01> df testvol

Filesystem              kbytes      used      avail capacity  Mounted on
/vol/testvol/           83886080  10538852  73347228     13%
/vol/testvol/
/vol/testvol/.snapshot  20971520           0  20971520     0%
/vol/testvol/.snapshot

netapp01>

```

If you notice the difference between the used space in the output of the df command and the used space from the df command we issued earlier prior to starting the clone process, there is only a slightly higher amount of used space now even though we have two 10GB logical files. This is because the file and the clone share physical data blocks. Only extra space occupied by the clone is for storing its metadata.

The clone that was just created must show up to the NFS and CIFS clients as a regular file. Let's go to the clients and figure out what they see.

On the Linux host, the ls command lists two files, each of size approximately 10GB:

```

netapph1%ls -lt
total 21012752
-rw-rw-rw- 1 root root 10737418244 Oct  6  2008 clone_test_file
-rw-rw-rw- 1 root root 10737418244 Oct  6 15:53 test_file

```

However, the df command on the client shows that the used space is only slightly more than the last time we issued a df before cloning:

```

netapph1%df
Filesystem              1K-blocks      Used Available Use% Mounted on
netapp01:/vol/testvol
                        83886080  10538848  73347232  13% /tmp/FlexClone

```

The used space is slightly more than 10GB even though we have two files of size 10GB each. Again, this is because the file and the clone share physical data blocks on the disk.

Here is the output of the du command on the Linux host:

```

netapph1%du -B 1K .
4          ./snapshot
21012760 .

```

As you can see, du reports approximately 20GB of used space. This is because there are two files of size 10GB each, even though on the storage system, they share data blocks.

The difference in used space as reported by du and df is an indication that the volume might have a FlexClone instance of a file or LUN in it.

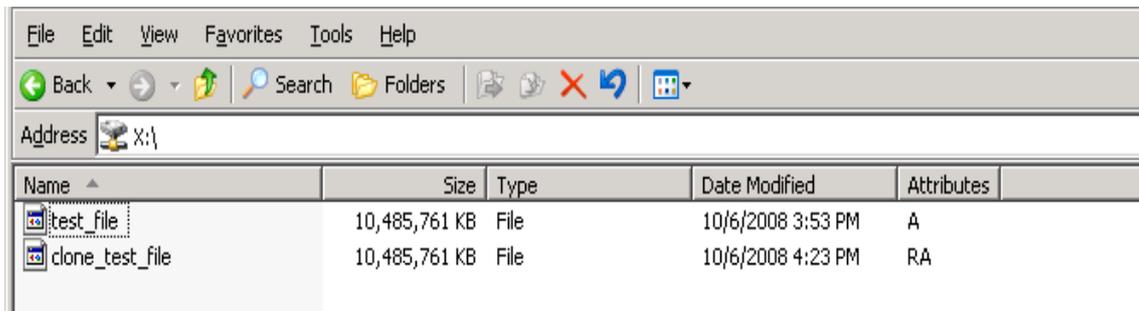
The difference between the used space as reported by df and du will give a rough estimate of the size of the shared data. So in this example the amount of shared data is:

$$21,012,760\text{KB} - 10,538,848\text{KB} = 10,538,848\text{KB}$$

So, we have approximately 10GB of shared data, which is true because we have a clone of a 10GB file, and the clone and its source share the same data blocks.

Now, let's see what the Windows client with the mapped CIFS share shows (Figure 7).

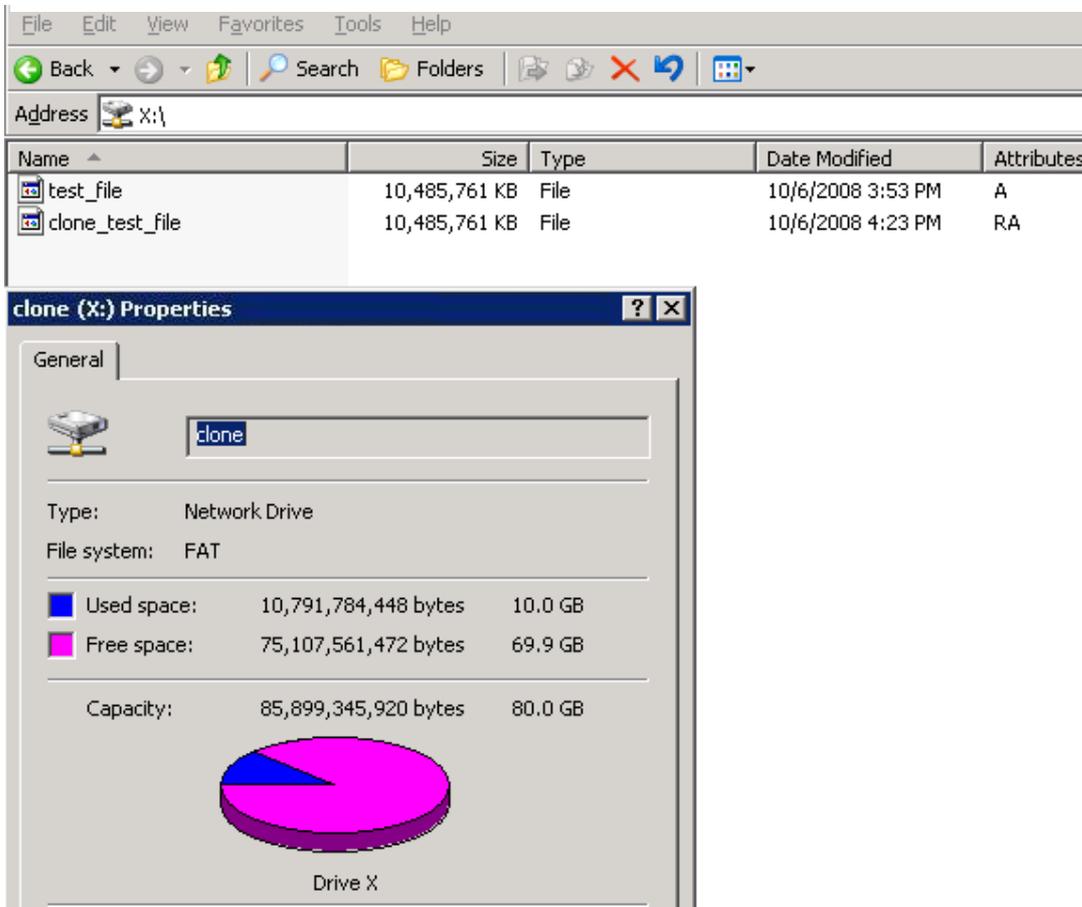
Figure 7) Files as seen from the client.



We see two files of size 10GB, just as we saw on the Linux host.

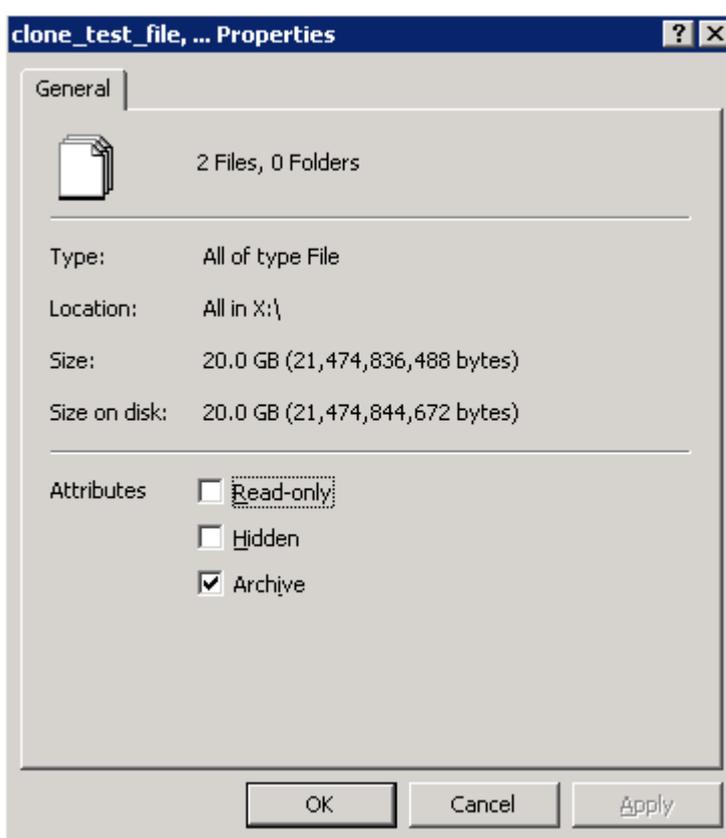
Figure 8 shows the properties of the share.

Figure 8) Properties of the share.



We see that the total used space on the CIFS share is 10GB even though we have two files each of size 10GB in the share. Again, this is because the file and the clone share physical data blocks on the disk. The amount of shared data can be calculated by subtracting the used space as shown in share properties from the sum of the sizes of the files (Figure 9).

Figure 9) Properties of all the files.



So, size of the shared data = 20GB – 10GB = 10GB, which is expected.

Now, both the researchers can start working on their own individual files. Both the file and clone show up in the UNIX® or Windows client as regular files. All file operations can be performed on both the clone and its source, including write, read, append, or delete. As data in either the source or the clone gets overwritten, more space on the disk is occupied, since the source file and the clone file no longer share the new data blocks that are written.

The “sis” attribute now appears in the output of the vol status command since the FlexVol volume now has a FlexClone instance of a file:

```
netapp01> vol status testvol

      Volume State      Status      Options
testvol online         raid_dp, flex
                        sis
Containing aggregate: 'testaggr'
```

Now, suppose another researcher comes up with a new algorithm and needs a copy of the file to work with. So, we create another clone as follows:

```
netapp01> clone start /vol/testvol/test_file
/vol/testvol/clone_test_file_2
Clone operation started successfully. ID: 539.
```

Here is the output of the df command on the storage system:

```
netapp01> df testvol
Filesystem          kbytes      used      avail capacity  Mounted on
/vol/testvol/      83886080  10539472  73346608  13%  /vol/testvol/
/vol/testvol/.snapshot 20971520          0  20971520    0%
/vol/testvol/.snapshot
```

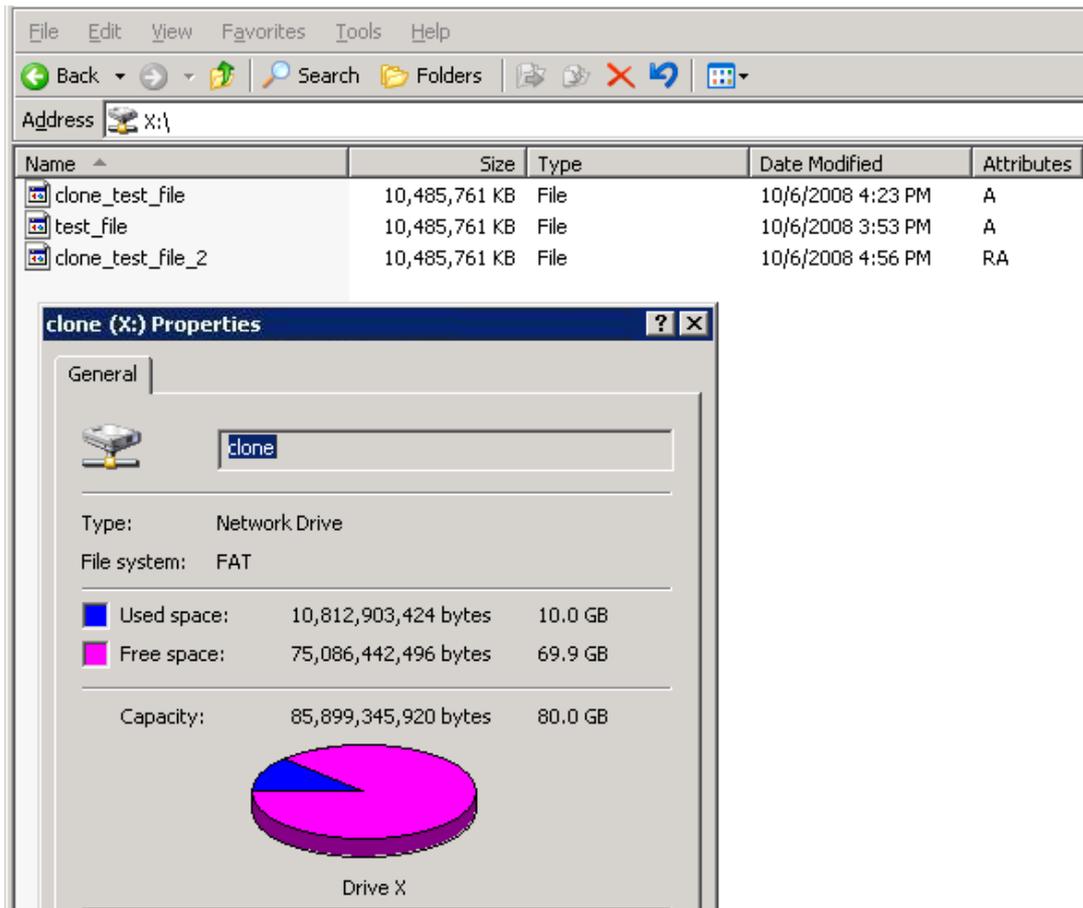
Notice that the used space is only slightly more than previous used space before creating the second clone. Again, this is because the file and the clone share physical data blocks. Only extra space occupied by the clone is for storing its metadata.

Now the Linux and Windows clients see three files. Here is the output from the Linux client:

```
netapph1%df .
Filesystem          1K-blocks      Used Available Use% Mounted on
netapp01:/vol/testvol
                        83886080  10559456  73326624  13% /tmp/FlexClone
netapph1%ls -lt
total 31519128
-rw-rw-rw-  1 root root 10737418244 Oct  6  2008 clone_test_file_2
-rw-rw-rw-  1 root root 10737418244 Oct  6 16:23 clone_test_file
-rw-rw-rw-  1 root root 10737418244 Oct  6 15:53 test_file
.
netapph1%du -B 1K .
4          ./snapshot
31519136
```

Figure 10 is the screen capture from the Windows client.

Figure 10) Properties of the share.



Notice that both the Linux and Windows client show a total of three files of size 10GB each but show a used space of approximately 10GB. This is because the source and clone files share data blocks.

Now that the files are ready for testing by the researchers for use, they start running their algorithms on the files. Suppose that the initial source file is the first to test the algorithm on. Let's assume that the algorithm overwrites about 1GB of data in the file.

As data in the source or clone is overwritten, the blocks that are overwritten will start to occupy space on the disk. So, after testing is complete on the first file, the admin will notice that disk utilization has gone up by approximately 1GB.

Now that the testing on the first file is complete, the admin can go ahead and delete the file:

```
netapph1% rm test_file
```

As discussed earlier, deleting the source file or any of its clones will not affect the other clone files in any way, and they continue to share blocks on disk.

Deleting the 10GB file will only return about 1GB of space back. That's because the 1GB was the data overwritten in the file in the course of testing. Even though the file was of size 10GB, this 1GB is the data blocks that were not shared. So, once the file is deleted, we get the 1GB of space back. When a file or clone that has shared blocks is deleted, the shared blocks continue to be used by the remaining clone files or LUNs. If there are other clone files that share the data blocks, deleting a file or its clone will only return to free space the space used for the metadata of that file or clone and the space used by the data blocks of that file or clone that are not shared. So, not much space is returned to free space until the source file or LUN and all clone files or LUNs of that source are deleted.

Even though the initial source file is deleted, all the clone files still exist and show up to the NFS or CIFS clients as normal files. They also continue to share blocks on disk.

The preceding use case example discusses in detail the complete lifecycle of a clone file and also highlights the advantages of FlexClone at file level.

As discussed in the previous sections of this document, the process to create a FlexClone instance of a LUN is the same as creating a FlexClone instance of a file. Here is the process to create a FlexClone instance of a LUN:

1. Make sure the LUN exists:

```
netapp01> lun show /vol/testvol/testlun
      /vol/testvol/testlun      10.0g (10742215680) (r/w, online,
mapped)
netapp01> lun show -v /vol/testvol/testlun
      /vol/testvol/testlun      10.0g (10742215680) (r/w, online,
mapped)
      Serial#: HnSrBoMaK8ZF
      Share: none
      Space Reservation: enabled
      Multiprotocol Type: windows
      Maps: igroup1=1
```

2. Create a FlexClone instance of the LUN:

```
netapp01> clone start /vol/testvol/testlun /vol/testvol/clone_testlun
Clone operation started successfully. ID: 9.
```

## 8 FLEXCLONE IN VIRTUALIZED ENVIRONMENTS

Virtualized environments provide a perfect use case for FlexClone. A virtual machine (VM) is made up of one or many binary files that are the disks of the virtual machine, and some set of associated files or metadata, which define the configuration of the VM. These virtual disk files can range in size from a few gigabytes to tens of gigabytes, depending on their use.

In a virtualized environment new virtual machines can be deployed using an existing VM as a template and cloning a new VM from that template. During this process a copy is made of the template virtual disk file, and new configuration files or metadata are generated and associated with the new virtual disk to create a new VM. The copy of the virtual disk file is the most time-consuming part of the process and is performed by the physical server hosting the virtual machine.

NetApp FlexClone provides the ability to offload the work of copying the virtual disks from the servers to the storage appliance.

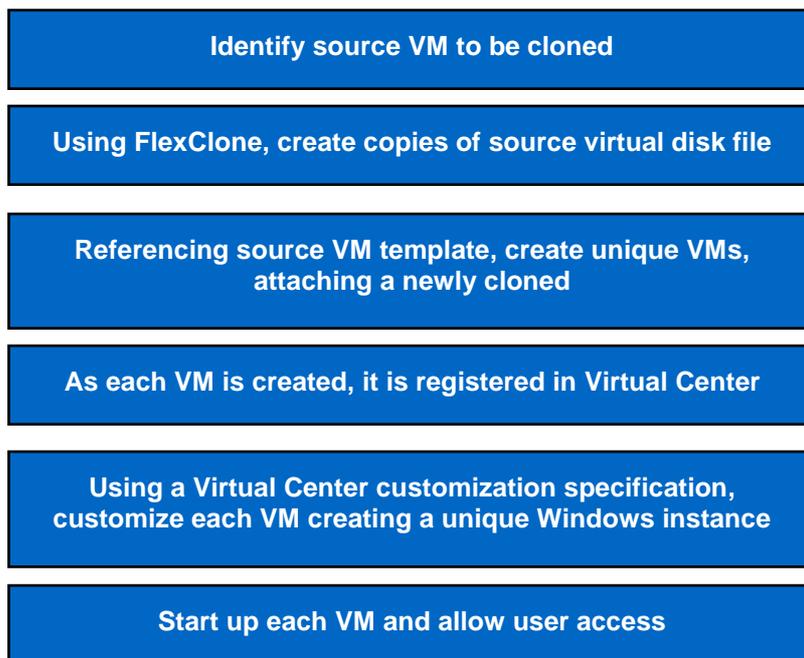
In virtual desktop infrastructure (VDI) environments the ability to quickly clone large numbers of VMs becomes even more attractive. In these environments there are typically hundreds or thousands of VMs, as compared to the number of virtual servers you might see in a virtual server environment. Also, in terms of VDI, there are cases where the cloning or deployment and redeployment of desktops provide greater flexibility in a VDI solution if the storage is capable of creating clones very quickly. For example, keeping up with operating system updates by patching a single copy of a VM and then redeploying that VM by cloning it thousands of times is an attractive solution. However, it is not possible to redeploy thousands of VMs in a reasonable frame of time using host-based copy methods. If the storage device does not enable a method of quick redeployment, then this type of process can be impossible to implement. Cloning hundreds or thousands of VMs takes time, for example, six to 12 minutes per VM or five to 10 per hour. That means to redeploy one thousand VMs could take over

four days, and during this time the servers suffer the additional workload of traditional copy processes.

FlexClone at the file level eliminates the I/O necessary to clone the VMs, enabling the cloning of mass numbers of virtual machines in minutes. However, there is more to creating clones than simply making copies of the virtual disks. For example, in a VMware® environment, VMs need to be registered in the Virtual Center Server and then customized to make them unique entities that can be joined to a Microsoft® Active Directory® domain.

Figure 11 is an example workflow of the VM cloning process. In this workflow NetApp FlexClone is used to replace the copy operation that would normally be performed by the host when creating multiple clones of VMs running Microsoft Windows as the guest operating system.

Figure 11) Workflow of the VM cloning process



NetApp has provided a utility to automate the entire process of cloning, registration, customization, and deployment of virtual machines. This utility is called the Rapid Cloning Utility (RCU). The RCU makes use of the native toolsets in the VMware Virtual Infrastructure SDK and the NetApp ONTAPI SDK, including FlexClone, and VMware Virtual Center features such as Guest Customization to efficiently clone and customize virtual machines. To customize VMs the RCU uses existing Guest Customization specifications stored within VMware Virtual Center. Monitoring of cloning activities can be done in the Virtual Infrastructure client, where actions are displayed as they are performed on the VMs. Information about the cloning process is also collected in the Data ONTAP messages file.

The Rapid Cloning Utility is discussed in NetApp TR-3705.

## 9 CONCLUSION

FlexClone technology provides a space- and time-efficient way to create and manage multiple copies of data on the storage system. The recent enhancements to FlexClone technology with increased granularity and Snapshot copy independence provide a powerful tool for storage administrators to rapidly deploy multiple copies of datasets while saving storage space.

## 10 REFERENCES

### 10.1 NETAPP TECHNICAL REPORTS AND WHITE PAPERS

- TR-3001: A Storage Networking Appliance  
[www.netapp.com/us/library/technical-reports/tr-3001.html](http://www.netapp.com/us/library/technical-reports/tr-3001.html)
- TR-3002: File System Design for an NFS File Server Appliance  
[www.netapp.com/us/library/white-papers/wp\\_3002.html](http://www.netapp.com/us/library/white-papers/wp_3002.html)
- TR-3347: A Thorough Introduction to Flexclone Volumes  
[www.netapp.com/us/library/technical-reports/tr-3347.html](http://www.netapp.com/us/library/technical-reports/tr-3347.html)
- TR-3505: Netapp Deduplication for FAS and V-Series Deployment and Implementation Guide  
[www.netapp.com/us/library/technical-reports/tr-3505.html](http://www.netapp.com/us/library/technical-reports/tr-3505.html)
- TR-3705: Netapp and Vmware View (Vdi) Solution Guide  
[www.netapp.com/us/library/technical-reports/tr-3705.html](http://www.netapp.com/us/library/technical-reports/tr-3705.html)

### 10.2 DATA ONTAP PRODUCT DOCUMENTATION

- System Administration Guide  
[http://now.netapp.com/NOW/knowledge/docs/ontap/ontap\\_index.shtml](http://now.netapp.com/NOW/knowledge/docs/ontap/ontap_index.shtml)
- Data Protection Online Backup and Recovery Guide  
[http://now.netapp.com/NOW/knowledge/docs/ontap/ontap\\_index.shtml](http://now.netapp.com/NOW/knowledge/docs/ontap/ontap_index.shtml)
- MultiStore Management Guide  
[http://now.netapp.com/NOW/knowledge/docs/ontap/ontap\\_index.shtml](http://now.netapp.com/NOW/knowledge/docs/ontap/ontap_index.shtml)
- Storage Management Guide  
[http://now.netapp.com/NOW/knowledge/docs/ontap/ontap\\_index.shtml](http://now.netapp.com/NOW/knowledge/docs/ontap/ontap_index.shtml)

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®



© Copyright 2014 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, xxx, and xxx are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. <<Insert third-party trademark notices here.>> All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.