



A Thorough Introduction to FlexClone™ Volumes

Miroslav Klivansky, Network Appliance Inc. | October 2004 | TR3347

WHITE PAPER

Network Appliance technology and expertise solve a wide range of data storage challenges for organizations, adding business value and enabling them to create and sustain a competitive advantage.

Overview	3
How <i>FlexClone</i> Volumes Work	4
Practical Applications of <i>FlexClone</i> Technology.....	6
An Extended Example	7
Background for Scenario Inc.....	8
Creating a <i>FlexClone</i> Volume	8
Destroying a <i>FlexClone</i> Volume	12
Splitting off a <i>FlexClone</i> Volume to Create an Independent Volume.....	12
Replacing a Volume with a Modified Clone	15
Cloning a <i>FlexClone</i> Volume	16
Multiple <i>FlexClone</i> Volumes from a Single Parent <i>FlexVol</i> Volume.....	18
SnapMirror and <i>FlexClone</i> Volumes	19
LUNs and <i>FlexClone</i> Volumes	21
Scenario Summary	25
FlexClone Performance.....	27
Best Practices for Success with <i>FlexClone</i> Volumes.....	28
Naming Conventions.....	28
Space Utilization and Reservations	28
Security, Permissions, and SnapLock Compliance Volumes	29
Snapshot Management	30
Cloning and SnapMirror.....	30
Backups of <i>FlexClone</i> Volumes	30
GUI Management and <i>FlexClone</i> Volumes	31
FlexClone Volumes for LUN Cloning	32
Summary	33
References	34
NetApp Technical Reports and White Papers	34
Data ONTAP 7G and Flexible Volumes Information.....	34
Data ONTAP 7G Documentation.....	34
Web Resources	34

Overview

Starting with Data ONTAP™ 7G, filer administrators have access to a powerful new feature that allows them to instantly create clones of a flexible volume (FlexVol volume). A FlexClone volume is a writable point-in-time image of a FlexVol™ volume or another FlexClone volume. FlexClone volumes add a new level of agility and efficiency to storage operations. They take only a few seconds to create and are created without interrupting access to the parent FlexVol volume. FlexClone volumes use space very efficiently, leveraging the Data ONTAP architecture to store only data that changes between the parent and clone. This is a huge potential saving in dollars, space, and energy. In addition to all these benefits, clone volumes have the same high performance as other kinds of volumes.

Conceptually, FlexClone volumes are great for any situation where testing or development occur, any situation where progress is made by locking in incremental improvements, and any situation where there is a desire to distribute data in changeable form without endangering the integrity of the original.

For example, imagine a situation where the IT staff needs to make substantive changes to a production environment. The cost and risk of a mistake are too high to do it on the production volume. Ideally, there would be an instant writable copy of the production system available at minimal cost in terms of storage and service interruptions. By using FlexClone volumes, the IT staff gets just that: an instant point-in-time copy of the production data that is created transparently and uses only enough space to hold the desired changes. They can then try out their upgrades using the FlexClone volumes. At every point that they make solid progress, they clone their working FlexClone volume to lock in the successes. At any point where they get stuck, they just destroy the working clone and go back to the point of their last success. When everything is finally working just the way they like, they can either split off the clone to replace their current production volumes or codify their successful upgrade process to use on the production system during the next maintenance window. The FlexClone feature allows them to make the necessary changes to their infrastructure without worrying about crashing their production systems or making untested changes on the system under tight maintenance window deadlines. The results are less risk, less stress, and higher levels of service for the IT customers.

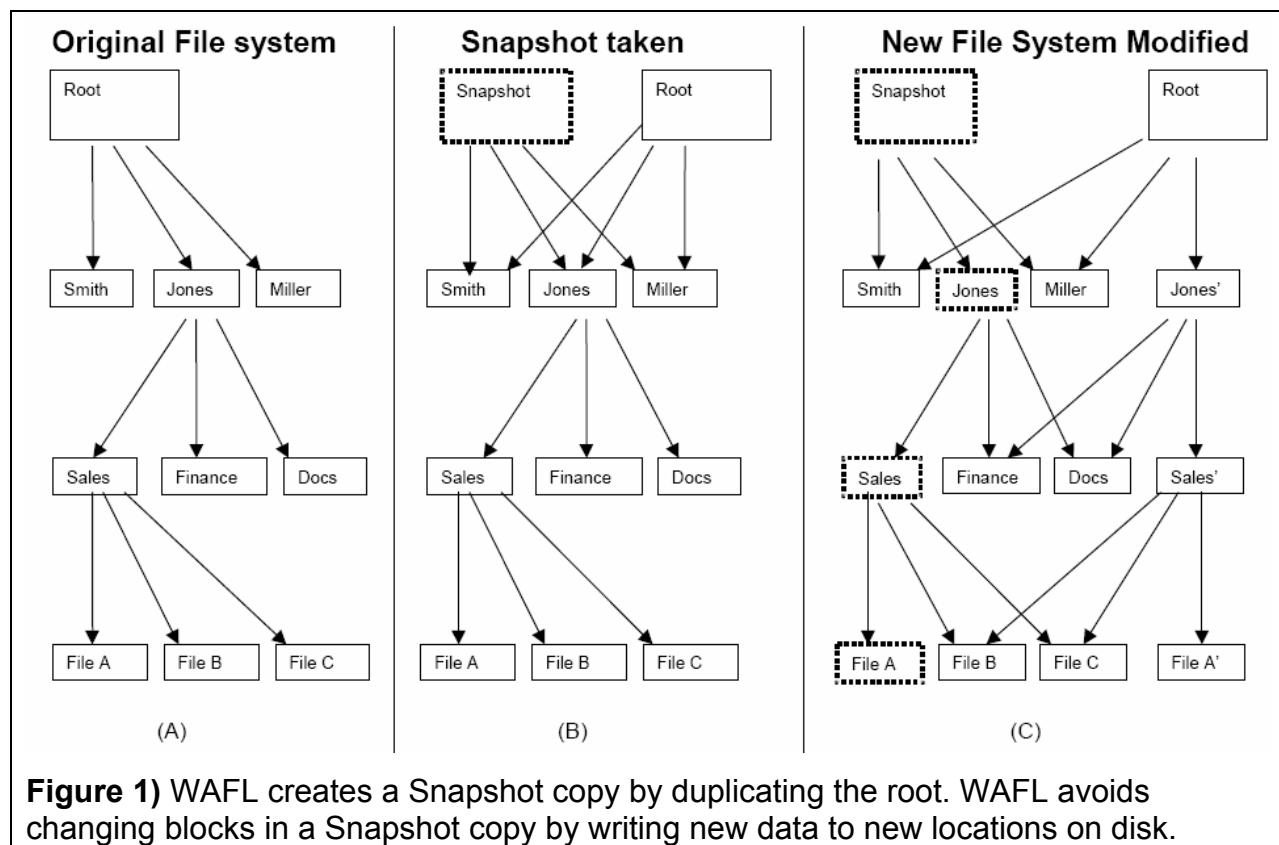
The goal of this report is to help storage system administrators understand the full value FlexClone volumes and Data ONTAP 7G can bring to their operations. In the following sections we will explain how FlexClone volumes work, explore practical applications for FlexClone technology, provide a detailed example scenario, discuss FlexClone performance, detail best practices for success with FlexClone volumes, and conclude with a list of references to learn even more. Table 1 provides some terms and definitions to smooth the way.

Table 1) Data ONTAP 7G terms and definitions.

Aggregate	Collection of physical disks in RAID groups, all shared by the flexible volumes in the aggregate.
Data ONTAP	Software developed by NetApp exclusively for appliances. Optimizes file service by combining patented file-system technology and a microkernel design dedicated to network data access.
FlexVol	A new type of storage volume that can be efficiently spread across many disks and easily resized to meet changing business requirements.
Snapshot™	A feature that creates an online, read-only copy of the entire file system—called a “Snapshot copy”—that protects against accidental deletions or modifications of files without duplicating file contents. Also used by FlexClone volumes for a point-in-time view of the parent.
WAFL®	Write Anywhere File Layout. A patented file system designed exclusively for NetApp appliances to optimize write performance. Developed by Network Appliance. Pronounced the same way as that food best served with butter and maple syrup!

How FlexClone Volumes Work

FlexClone volumes have all the capabilities of a FlexVol volume, including growing, shrinking, and being the source of a Snapshot copy or even another FlexClone volume. The technology that makes this all possible is integral to how Data ONTAP manages storage. NetApp filers use a Write Anywhere File Layout (WAFL) to manage disk storage. Any new data that gets written to the volume doesn't need to go on a specific spot on the disk; it can be written anywhere. WAFL then updates the metadata to integrate the newly written data into the right place in the file system. If the new data is meant to replace older data, and the older data is not part of a Snapshot copy, WAFL will mark the blocks containing the old data as reusable. This can happen asynchronously and does not affect performance. Snapshot copies work by making a copy of the metadata associated with the volume. Data ONTAP preserves pointers to all the disk blocks currently in use at the time the Snapshot copy is created. When a file is changed, the Snapshot copy still points to the disk blocks where the file existed before it was modified, and changes are written to new disk blocks. As data is changed in the parent FlexVol volume, the original data blocks stay associated with the Snapshot copy rather than getting marked for reuse. All the metadata updates are just pointer changes, and the filer takes advantage of locality of reference, NVRAM, and RAID technology to keep everything fast and reliable. Figure 1 provides a graphical illustration of how this works.



You can think of a FlexClone volume as a transparent writable layer in front of the Snapshot copy. Figure 2 provides a memorable illustration of that concept. A FlexClone volume is writable, so it needs some physical space to store the data that is written to the clone. It uses the same mechanism used by Snapshot copies to get available blocks from the containing aggregate. Whereas a Snapshot copy simply links to existing data that was overwritten in the parent, a FlexClone volume stores the data written to it on disk (using WAFL) and then links to the new data as well. The disk space associated with the Snapshot copy and FlexClone is accounted for separately from the data in the parent FlexVol volume. When a FlexClone volume is first created, it needs to know the parent FlexVol volume and also a Snapshot copy of the parent to use as its base. The Snapshot copy can already exist, or it can get created automatically as part of the cloning operation. The FlexClone volume gets a copy of the Snapshot copy metadata and then updates its metadata as the clone volume is created. Creating the FlexClone volume takes just a few moments because the copied metadata is very small compared to the actual data. The parent FlexVol volume can change independently of the FlexClone volume because the Snapshot copy is there to keep track of the changes and prevent the original parent's blocks from being reused while the Snapshot copy exists. The same Snapshot copy is read-only and can be efficiently reused as the base for multiple FlexClone volumes. Space is used very efficiently, since the only new disk space used is either associated with the small amounts of metadata or updates and/or additions to either the parent FlexVol volume or the FlexClone volume. While that may sound like magic to anybody not familiar with NetApp, it works and uses underlying NetApp technology deployed for many years in production enterprise environments. To learn more, please see the [technical reports](#) listed in the References section.

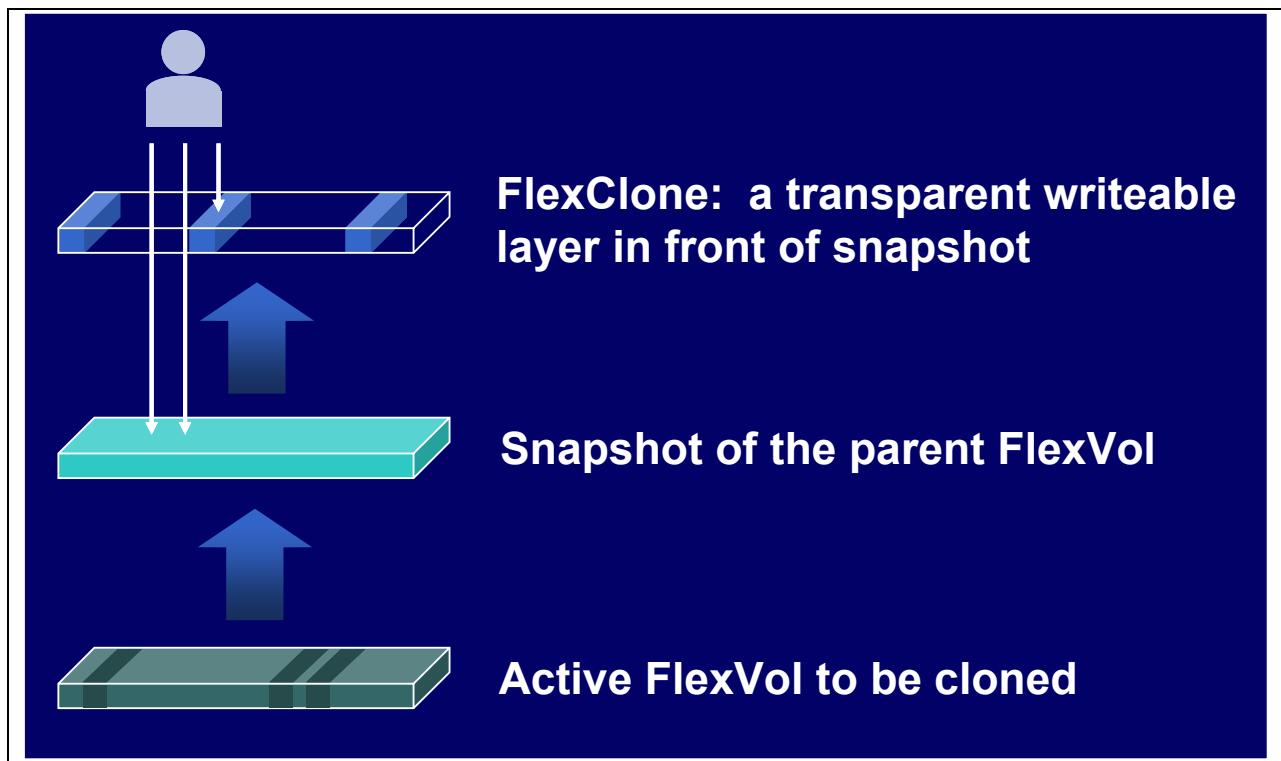


Figure 2) Think of a FlexClone volume as a transparent writable layer in front of a Snapshot copy.

FlexClone volumes appear to the storage administrator just like a FlexVol volume, which is to say that they look like a regular volume and have all of the same properties and capabilities. Using the CLI, FilerView® or DataFabric® Manager, one can manage volumes, Snapshot copies, and FlexClone volumes—including getting their status and seeing the relationships between the parent, Snapshot copy, and clone. In Data ONTAP 7G, the CLI is required to create and split a FlexClone volume. FlexClone volumes are treated just like a FlexVol volume for most operations. The main limitation is that Data ONTAP forbids operations that would destroy the parent FlexVol volume or base Snapshot copy while dependent FlexClone volumes exist. Other caveats are that management information in external files (e.g., /etc) associated with the parent FlexVol volume is not copied, quotas for the clone volume get reset rather than added to the parent FlexVol volume, and LUNs in the cloned volume are automatically marked offline until they are uniquely mapped to a host system. Lastly, splitting the FlexClone volume from the parent volume to create a fully independent volume requires adequate free space in the aggregate to copy shared blocks.

Practical Applications of FlexClone Technology

FlexClone technology enables multiple, instant data set clones with no storage overhead. It provides dramatic improvements for application test and development environments and is tightly integrated with the patented NetApp file-system technology and a microkernel design in a way that renders competitive methods archaic. FlexClone volumes are ideal for managing production data sets. They allow effortless error containment for bug fixing and development. They simplify platform upgrades for ERP and CRM applications. Instant FlexClone volumes provide data for multiple simulations against large data sets for ECAD, MCAD, and Seismic applications—all without unnecessary duplication or waste of physical space. The ability to split FlexClone volumes from their parent lets administrators easily create new permanent, independent volumes for forking project data.

FlexClone volumes have their limits, but the real range of applications is limited only by imagination. Table 2 lists a few of the more common examples.

Table 2) Example application areas for Data ONTAP 7G FlexClone volumes.

Application Area	Benefit
Application Testing	Make the necessary changes to infrastructure without worrying about crashing production systems. Avoid making untested changes on the system under tight maintenance window deadlines. Less risk, less stress, and higher service-level agreements.
Data Mining	Data mining operations and software can be implemented more flexibly because both reads and writes are allowed.
Parallel Processing	Multiple FlexClone volumes of a single milestone/production data set can be used by parallel processing applications across multiple servers to get results more quickly.

Online Backup	Immediately resume read-write workload on discovering corruption in the production data set by mounting the clone instead. Use database features such as DB2 write-suspend or Oracle® hot backup mode to transparently prepare the database volumes for cloning by delaying write activity to the database. This is necessary because databases need to maintain a point of consistency.
System Deployment	Maintain a template environment and use FlexClone volumes to build and deploy either identical or variation environments. Create a test template that is cloned as needed for predictable testing. Faster and more efficient migration using the Data ONTAP SnapMirror® feature in combination with FlexClone volumes.
IT Operations	Maintain multiple copies of production systems: live, development, test, reporting, etc. Refresh working FlexClone volumes regularly to work on data as close to live production systems as practical.

An Extended Example

The best way to get a sense of how FlexClone volumes work and how they may be of benefit is through an in-depth example. Our example is based on a fictional production environment that can help highlight how FlexClone volumes can add real business value!

Table 3 provides an overview of the commands associated with FlexClone volumes as a reference to help along the way.

Table 3) Data ONTAP 7G commands related to FlexClone volumes.

Function	Syntax	Example
Creation	vol clone create cl_vol_name [-s {volume file none}] -b f_p_vol_name [parent_snap]	vol clone create Widget -s file -b Gadget GadgetSnap
	<i>Create a new read-write clone named Widget, based on an existing flexible volume named Gadget. Reuse an existing Snapshot copy in Gadget as the backing for the clone.</i>	
Status	vol status [-v] cl_vol_name	vol status -v Widget
	<i>Print verbose status information for the volume named Widget. The volume can be a FlexVol volume, FlexClone volume, or traditional volume. Prints less information without the “-v” flag.</i>	
Splitting	vol clone split start cl_vol_name	vol clone split start Widget
	<i>Start the process of splitting the FlexClone volume named Widget from its parent FlexVol volume. The split will make a copy of all blocks associated with the FlexClone volume that currently reside in the parent FlexVol volume. The process can take a while to complete and proceeds in the background. Afterward the volume will be an independent FlexVol volume with space guarantees reenabled.</i>	
	vol clone split status cl_vol_name	vol clone split status Widget

	<i>Check on the progress of the clone split going on for the Widget volume.</i>	
	vol clone split stop cl_vol_name	vol clone split stop Widget
<i>Stop the clone split process. Any blocks that have already been copied will continue to be associated with the clone volume. Another split start command will resume the split with any blocks that are still shared.</i>		
Space Utilization	df -A aggr_name	
	<i>Display disk usage associated with aggregates. In the example above, limit it to the aggregate named aggr1.</i>	
	vol container vol_name	vol container Widget
	<i>Display the name of the aggregate containing the volume named Widget.</i>	

For a complete reference to the Data ONTAP 7G CLI, please see the [Data ONTAP 7G Command Reference](#).

Background for Scenario Inc.

For our example we'll use a fictional company called Scenario Inc. (or often just SI for those in the know). SI is a small and innovative company that specializes in scenario planning for large multinational corporations and governments.

SI's work calls for analyzing lots of data and synthesizing it into likely scenarios. SI heavily leverages technology to gain competitive advantage for its clients. SI's data storage needs include databases, complex simulations running on multiple compute nodes, and data in flat files for custom applications that SI prototyped for its clients. Having built its reputation on timeliness and accuracy, SI needs to rapidly produce results.

SI's extensive collection of data is the basis for current and new products and cannot be compromised. SI wants to keep costs down by eliminating unnecessary copies, yet wants the flexibility and peace of mind of not experimenting with its valuable data. SI has recently migrated to NetApp filers running the latest DOT7.0 software, including the FlexClone feature. SI is in great shape to deliver value for its clients!

Creating a FlexClone Volume

SI just landed a new contract with Gadget Corp. and wants to impress Gadget with quick results. Luckily SI has done similar work with Widget Corp. and can reuse part of the Widget work as a template. Alice gets the assignment and decides that she will start the development efforts by cloning the tools and data sets SI created for Widget's scenarios.

Using the **vol status** and **df** commands from the CLI of her file (named "toaster"), Alice refreshes her memory of the Widget volumes. She sees that they are part of the aggregate named "aggr1," which still has space available.

```
toaster> vol status -v WidgetData
      Volume      State      Status      Options

```

```

WidgetData    online      raid_dp, flex      nosnap=off, nosnapdir=off, minra=off,
                no_atime_update=off, nvfail=off, snapmirrored=off,
                create_icode=on, convert_icode=on, maxdirsize=1310,
                fs_size_fixed=off, guarantee=volume, svo_enable=off,
                svo_checksum=off, svo_allow_rman=off, svo_reject_errors=off,
                fractional_reserve=100

Containing aggregate: 'aggr1'
Plex /aggr1/plex0: online, normal, active
RAID group /aggr1/plex0/rg0: normal

```

```

toaster> df -g
Filesystem      total    used   avail   capacity  Mounted on
/vol/vol0/      253GB   57GB   196GB   23%       /vol/vol0/
/vol/vol0/.snapshot  0GB    0GB    0GB    ---%       /vol/vol0/.snapshot
/vol/WidgetData/ 16GB    1GB    14GB    9%       /vol/WidgetData/
/vol/WidgetData/.snapshot  4GB    0GB    3GB    2%       /vol/WidgetData/.snapshot
/vol/WidgetTools/ 16GB    7GB    8GB    44%       /vol/WidgetTools/
/vol/WidgetTools/.snapshot  4GB    0GB    3GB    1%       /vol/WidgetTools/.snapshot

toaster> df -Ag
Aggregate      total    used   avail   capacity
aggr0          256GB   254GB   1GB    99%
aggr0/.snapshot 13GB    0GB    13GB    0%
aggr1          109GB   41GB    67GB   38%
aggr1/.snapshot 5GB    0GB    5GB    0%

```

Alice then creates FlexClone volumes of the two template flexible volumes, giving them names meaningful to her new project. She does not specify the name of a backing Snapshot copy, so Data ONTAP creates one automatically to use for the new FlexClone volumes.

```

toaster> vol clone create GadgetTools -b WidgetTools
Wed Sep 22 22:21:32 GMT [wafl.snaprestore.revert:notice]:
Reverting volume GadgetTools to a previous snapshot.
Wed Sep 22 22:21:32 GMT [wafl.vol.guarantee.clone:info]:
Space for clone volume 'GadgetTools' is not guaranteed.
Creation of clone volume 'GadgetTools' has completed.

```

```

toaster> vol clone create GadgetData -b WidgetData
Wed Sep 22 22:21:52 GMT [wafl.snaprestore.revert:notice]:
Reverting volume GadgetData to a previous snapshot.
Wed Sep 22 22:21:53 GMT [wafl.vol.guarantee.clone:info]:
Space for clone volume 'GadgetData' is not guaranteed.
Creation of clone volume 'GadgetData' has completed.

```

```

toaster> snap list
Volume WidgetData working...
  %/used  %/total  date      name
  -----  -----  -----
  2% ( 2%)  0% ( 0%) Sep 22 22:21  clone_GadgetData.1 (busy,vclone)
  5% ( 3%)  0% ( 0%) Sep 22 20:00  hourly.0
Volume WidgetTools working...
  %/used  %/total  date      name
  -----  -----  -----

```

```

2% ( 2%) 0% ( 0%) Sep 22 22:21 clone_GadgetTools.1 (busy,vclone)
Volume GadgetTools working...
  %/used   %/total  date      name
-----
2% ( 2%) 0% ( 0%) Sep 22 22:21 clone_GadgetTools.1
Volume GadgetData working...
  %/used   %/total  date      name
-----
3% ( 3%) 0% ( 0%) Sep 22 22:21 clone_GadgetData.1

```

During the **vol clone** command, Data ONTAP prints an informational message saying “Reverting volume *GadgetData* to a previous snapshot.” For those not familiar with Data ONTAP, this is the standard message when a Snapshot copy is used to restore a volume to a previous state. Since FlexClone volumes leverage Snapshot technology to get a point-in-time image of the parent FlexVol volume, the same mechanism and message are used. The volume mentioned in the message is the new FlexClone volume. Although the word “revert” implies that it is going back to a previous version, it is not actually “reverted,” since it has just come into existence.

The **snap list** command shows that there are now two new volumes, “GadgetTools” and “GadgetData,” plus two new Snapshot copies, “clone_GadgetTools.1” and “clone_GadgetData.1” that appear associated with both the parent and the clone volumes. In the clone volume they appear as regular Snapshot copies. In the parent FlexVol volume, there is the added “(busy,vclone)” notation to indicate that the Snapshot copy is being used as the backing Snapshot copy for a FlexClone volume.

Alice remembers her NetApp University training and notes the message printed by Data ONTAP about space reservations for FlexClone volumes. Data ONTAP 7G does not guarantee space reservations for clone volumes and mentions that guarantees are disabled. The clone volume inherits the same space reservation mode as the parent, and these reservations are enforced when the clone volume is split from the parent. However, due to the complex relationship between parent and clone space usage, space reservations for the clone are explicitly not enforced by Data ONTAP. It is up to the system administrator to make sure there is sufficient space in the aggregate to accommodate all requested changes to the parent and the clone. Data ONTAP makes this much simpler than traditional storage by using aggregates and flexible volumes, which can be dynamically resized without disrupting access.

```

toaster> vol options GadgetTools
nosnap=off, nosnapdir=off, minra=off, no_atime_update=off, nvfail=off,
snapmirrored=off, create_icode=on, convert_icode=on, maxdirsize=1310,
fs_size_fixed=off, guarantee=volume(disabled), svo_enable=off,
svo_checksum=off, svo_allow_rman=off, svo_reject_errors=off,
fractional_reserve=100

```

```

toaster> df -g
Filesystem      total    used   avail capacity Mounted on
/vol/vol0/      253GB   57GB   196GB  23%    /vol/vol0/
/vol/vol0/.snapshot  0GB    0GB    0GB   ---%   /vol/vol0/.snapshot
/vol/WidgetData/ 16GB    1GB    14GB   9%    /vol/WidgetData/
/vol/WidgetData/.snapshot  4GB    0GB    3GB   2%    /vol/WidgetData/.snapshot
/vol/WidgetTools/ 16GB    7GB    8GB   44%    /vol/WidgetTools/
/vol/WidgetTools/.snapshot  4GB    0GB    3GB   1%    /vol/WidgetTools/.snapshot
/vol/GadgetTools/ 16GB    7GB    8GB   44%    /vol/GadgetTools/
/vol/GadgetTools/.snapshot  4GB    0GB    3GB   1%    /vol/GadgetTools/.snapshot

```

```

/vol/GadgetData/          16GB   1GB   14GB      9%   /vol/GadgetData/
/vol/GadgetData/.snapshot  4GB    0GB    3GB      1%   /vol/GadgetData/.snapshot

toaster> df -Ag
Aggregate      total    used   avail capacity
aggro          256GB   254GB   1GB    99%
aggro/.snapshot 13GB    0GB    13GB    0%
aggri          109GB   41GB   67GB   38%
aggri/.snapshot 5GB    0GB    5GB    0%

```

Alice uses the **vol options** command to check the options for a clone volume. She notes that space guarantees for the volume are marked as disabled, so she will need to be a bit more careful that the volume doesn't run out of space by watching space usage and setting alerts based on available space in the aggregate. She also notes that both the parent and clone volumes are considered to be the same size, and no additional space has been used in the containing aggregate.

She's ready to get to work! Alice creates about 2GB of new data in the cloned volume. That data shows up as a larger size for GadgetTools in the **df** command:

```

toaster> df -g GadgetTools
Filesystem      total    used   avail capacity Mounted on
/vol/GadgetTools/   16GB   9GB    6GB    61%   /vol/GadgetTools/
/vol/GadgetTools/.snapshot  4GB    0GB    3GB    1%   /vol/GadgetTools/.snapshot

```

Alice then deletes some of the original files that were in the volume from the parent because she doesn't want them to confuse her current work:

```

toaster> df -g GadgetTools
Filesystem      total    used   avail capacity Mounted on
/vol/GadgetTools/   16GB   5GB    6GB    61%   /vol/GadgetTools/
/vol/GadgetTools/.snapshot  4GB    4GB    0GB    96%   /vol/GadgetTools/.snapshot

```

Notice that the amount of space used in GadgetTools went down from 9GB to 5GB, but the amount accounted for by the associated Snapshot copy space went up from 0GB to 4GB. That's because Data ONTAP accounts for blocks changed between the parent FlexVol volume and the FlexClone volume by associating them with the Snapshot copy space. The deleted data might still be needed by the parent FlexVol volume, so the blocks associated with the deleted files are linked to by the parent and accounted for in the Snapshot copy space.

Alice notices that the Snapshot copy has used up 96% of the snap reserve. The snap reserve is not a hard limit and can go above 100%. The space used comes from the available space in the aggregate, just like the space for new data written to the FlexClone volume. (To learn more about Snapshot copy disk consumption, please see the [Data ONTAP 7G Online Backup and Recovery Guide](#).) There is no danger of running out of space as long as there is available space in the aggregate. Alice likes to keep her filer tidy and would prefer to keep the space associated with Snapshot copies below 100%. That's no problem, because Data ONTAP 7G allows her to resize a FlexClone volume on the fly—just like a regular FlexVol volume. All she has to do is add some space from the containing aggregate and adjust the snap reserve to allocate more of the space to the Snapshot copy:

```

toaster> vol size GadgetTools +20g
vol size: Flexible volume 'GadgetTools' size set to 40g.

```

```

toaster> snap reserve -V GadgetTools 50

```

```

toaster> df -g GadgetTools
Filesystem      total  used  avail  capacity  Mounted on
/vol/GadgetTools/    20GB  5GB  14GB  30%       /vol/GadgetTools/
/vol/GadgetTools/.snapshot    20GB  3GB  16GB  19%       /vol/GadgetTools/.snapshot

toaster> df -Ag aggr1
Aggregate      total  used  avail  capacity
aggr1        109GB  44GB  65GB  41%
aggr1/.snapshot    5GB   0GB   5GB   0%

```

Notice that the amount of space associated with GadgetTools is now 40GB—20GB for the volume and another 20GB for the Snapshot copies. The amount of space used in the aggregate did not increase by 20GB. That's because space reservations are disabled for FlexClone volumes in Data ONTAP 7G. Additional space will be used from the aggregate up to the limit shown as “avail” space, but it will not be reserved ahead of time. It is possible that some changes to the clone volume will fail because the aggregate has run out of space. The way around that is to add disks to the aggregate, which can be done on-the-fly. An example is shown later in the section [LUNs and FlexClone Volumes](#).

Destroying a FlexClone Volume

As Alice continues to work, she realizes that she may be able to reuse some work from the GeeWhiz project. She makes a clone of the volume in question, but some time along the way realizes that it's not particularly helpful and might just create more confusion. She destroys the clone volume with the following sequence of commands:

```

toaster> vol offline Whizzer
Thu Sep 23 01:28:50 GMT [cifs.terminationNotice:warning]:
CIFS: shut down completed: CIFS is disabled for volume Whizzer.
Volume 'Whizzer' is now offline.

toaster> vol destroy Whizzer
Are you sure you want to destroy this volume? yes
Volume 'Whizzer' destroyed.

toaster> snap delete WidgetTools clone_Whizzer.1

```

With that complete, Alice focuses back on her previous direction and continues to make good progress with the project.

Splitting off a FlexClone Volume to Create an Independent Volume

The prototyping work for Widget has gone well. After tinkering with the tool code, SI has figured out how to build tools that do not require any transformation in the data. SI decides to split off the tools clone and destroy the data sets clone. FlexClone volumes can be used indefinitely, but there are a number of good reasons for a storage administrator to split off a FlexClone volume to create a fully independent FlexVol volume. They may wish to replace the current parent FlexVol volume with the modified FlexClone volume, need to free the blocks pinned down by the FlexClone volume base

Snapshot copy, or wish to have Data ONTAP enforce space reservations for the volume for more predictable administration. Performance is *not* one of the reasons. Since the clone volume uses the same aggregate as the parent, they both get to use the exact same disks. Both take advantage of WAFL and NVRAM for fast writes, and since changes can be written to anywhere on disk, it doesn't matter if it is the clone or independent metadata that gets updated. Here is how it's done...

First Alice needs to make sure there is enough space available in the aggregate to support the new independent volume:

```
toaster> df -g GadgetTools
Filesystem      total    used   avail  capacity  Mounted on
/vol/GadgetTools/  20GB    5GB   14GB   30%      /vol/GadgetTools/
/vol/GadgetTools/.snapshot  20GB    3GB   16GB   16%      19%
/vol/GadgetTools/.snapshot

toaster> df -Ag aggr1
Aggregate      total    used   avail  capacity
aggr1          109GB   44GB   65GB   41%
aggr1/.snapshot  5GB    0GB    5GB    0%
```

The GadgetTools volume and its associated Snapshot copy area take up 40GB. There is still 65GB available in the containing aggregate, so the coast is clear. Alice initiates the split and checks progress a little bit into the split:

```
toaster> vol clone split start GadgetTools
Thu Sep 23 18:19:19 GMT [wafl.scan.start:info]: Starting volume clone split on volume GadgetTools.
Clone volume 'GadgetTools' will be split from its parent.
Monitor system log or use 'vol clone split status' for progress.

toaster> vol clone split status
Volume 'GadgetTools', 1263616 of 1440768 inodes processed (88%)
4676608 blocks scanned. 2401280 blocks updated.
Thu Sep 23 18:19:30 GMT [wafl.vol.add.spcGuar:info]: Space for volume GadgetTools is guaranteed.
Thu Sep 23 18:19:31 GMT [wafl.scan.clone.split.complete:info]: Volume GadgetTools, clone split complete.
```

After the split completes successfully, the space reservations are again enabled, and the space is guaranteed. Alice takes a closer look. Notice that the new volume has used approximately 40GB from the aggregate, and that the Snapshot copy area is currently empty. That's because all the data is now associated with the volume, and there is no Snapshot copy data in the brand-new volume. In addition, the **vol status** command does not mention the volume being a clone or having space guarantees disabled. In fact, the **vol options** command shows that space is guaranteed for the volume.

```
toaster> df -g GadgetTools
Filesystem      total    used   avail  capacity  Mounted on
/vol/GadgetTools/  20GB    6GB   13GB   31%      /vol/GadgetTools/
/vol/GadgetTools/.snapshot  20GB    0GB   20GB   0%      /vol/GadgetTools/.snapshot

toaster> df -Ag aggr1
Aggregate      total    used   avail  capacity
aggr1          109GB   81GB   27GB   75%
aggr1/.snapshot  5GB    0GB    5GB    0%
```

toaster> vol status GadgetTools				
Volume	State	Status		Options

```
GadgetTools  online  raid_dp, flex      create_icode=on,
                                         convert_icode=on
                                         Containing aggregate: 'aggr1'
```

```
toaster> vol options GadgetTools
nosnap=off, nosnapdir=off, minra=off, no_atime_update=off, nvfail=off,
snapmirrored=off, create_icode=on, convert_icode=on, maxdirsize=1310,
fs_size_fixed=off, guarantee=volume, svo_enable=off, svo_checksum=off,
svo_allow_rman=off, svo_reject_errors=off, fractional_reserve=100
```

The clone split operation may take some time and proceeds in the background. Alice uses the **vol clone split status** command to view the command's progress. Both the clone and parent flexible volumes remain available during this process of splitting them apart. Upon completion, the Snapshot copy on which the clone was based will be unlocked in the parent FlexVol volume. Any Snapshot copies in the clone are automatically removed by Data ONTAP at the end of processing. The backing Snapshot copy in the parent FlexVol volume is not removed, since it may be needed by other FlexClone volumes or just desired as a regular Snapshot copy.

Alice could have used the **vol clone split stop** command to stop the split process. If she did, all of the blocks that have already been duplicated by the **vol clone split start** command would remain split apart. If the split were ever started again, the remaining shared and new blocks would be separated. It's not exactly like starting where the previous split left off, but it is close.

Now that GadgetTools is an independent volume, Alice wants to tidy things up a bit. The backing Snapshot copy in the parent FlexVol volume is no longer needed. Plus now that most of the development work has stabilized, Alice doesn't need the volume to be quite as large or to reserve as large a fraction of space for the Snapshot copies. Lastly, the *GadgetData* clone isn't needed because original data can be reused without modification. Here's what she does:

```
toaster> snap list WidgetTools
Volume WidgetTools working...
      %/used    %/total   date      name
-----  -----
  1% ( 1%)  0% ( 0%) Sep 23 20:00 hourly.0
  1% ( 1%)  0% ( 0%) Sep 23 00:00 nightly.0
  2% ( 1%)  1% ( 0%) Sep 22 22:21 clone_GadgetTools.1

toaster> snap delete WidgetTools clone_GadgetTools.1

toaster> snap reserve GadgetTools 20

toaster> df -g GadgetTools
Filesystem          total    used   avail  capacity  Mounted on
/vol/GadgetTools/    32GB    6GB   25GB   19%       /vol/GadgetTools/
/vol/GadgetTools/.snapshot  8GB    0GB   7GB    0%       /vol/GadgetTools/.snapshot

toaster> vol size GadgetTools 20g
vol size: Flexible volume 'GadgetTools' size set to 20g

toaster> df -g GadgetTools
Filesystem          total    used   avail  capacity  Mounted on
/vol/GadgetTools/    16GB    6GB   9GB    39%       /vol/GadgetTools/
/vol/GadgetTools/.snapshot  4GB    0GB   3GB    1%       /vol/GadgetTools/.snapshot

toaster> df -Ag aggr1
```

```

Aggregate          total    used    avail  capacity
  aggr1           109GB   61GB   47GB   57%
  aggr1/.snapshot  5GB    0GB    5GB    7%

```

toaster> **vol offline GadgetData**

Thu Sep 23 23:02:51 GMT [cifs.terminationNotice:warning]:
CIFS: shut down completed: CIFS is disabled for volume GadgetData.
Volume 'GadgetData' is now offline.

toaster> **vol destroy GadgetData**

Are you sure you want to destroy this volume? yes
Volume 'GadgetData' destroyed.

Notice that the **vol size** command above *reduced* the size of the volume down to 20GB. The ability to reduce the size of a FlexVol volume is a new feature of Data ONTAP 7G.

Replacing a Volume with a Modified Clone

After a couple of days of using the new volume, Alice gets a great idea that she would like to try. Rather than doing it directly in the stable GadgetTools volume, she clones it and makes her changes in a clone volume she names NewTools. The idea works, and after some further testing Alice decides to move the new work into production in place of the current stable version:

```

toaster> df -g
Filesystem      total    used    avail  capacity      Mounted on
/vol/GadgetTools/  16GB    6GB    9GB    39%      /vol/GadgetTools/
/vol/GadgetTools/.snapshot  4GB    0GB    3GB    2%      /vol/GadgetTools/.snapshot
/vol/NewTools/   16GB   11GB    4GB    73%      /vol/NewTools/
/vol/NewTools/.snapshot  4GB    1GB    2GB    49%      /vol/NewTools/.snapshot

```

toaster> **vol clone split start NewTools**

Thu Sep 23 23:29:30 GMT [wafl.scan.start:info]: Starting volume clone split on volume NewTools.
Clone volume 'NewTools' will be split from its parent.
Monitor system log or use 'vol clone split status' for progress.
Thu Sep 23 23:29:37 GMT [wafl.vol.add.spcGuar:info]: Space for volume NewTools is guaranteed.
Thu Sep 23 23:29:38 GMT [wafl.scan.clone.split.complete:info]: Volume NewTools, clone split complete.

toaster> **vol rename GadgetTools OLD**
'GadgetTools' renamed to 'OLD'

toaster> **vol rename NewTools GadgetTools**
'NewTools' renamed to 'GadgetTools'

```

toaster> cifs shares
Name      Mount Point      Description
---      -----
Tools      /vol/OLD
NewTools   /vol/GadgetTools

```

Up to this point everything happens transparently on the filer. Now Alice needs to make the new FlexVol volume available to the host in place of the old one. The details would be different depending on the access method (CIFS, NFS, iSCSI, or FCP), but the process would be similar to the steps below. To be most safe, Alice would disable access to the volume until the new volume is accessible

under the existing access path. The share or export operation completes in seconds, so the volume would not need to stay unavailable for long.

```
toaster> cifs shares -delete Tools
toaster> cifs shares -add Tools /vol/GadgetTools
toaster> cifs shares -delete NewTools

toaster> cifs shares
Name          Mount Point          Description
---          -----
Tools          /vol/GadgetTools
```

```
toaster> vol offline OLD
Thu Sep 23 23:44:45 GMT [cifs.terminationNotice:warning]:
CIFS: shut down completed: CIFS is disabled for volume OLD.
Volume 'OLD' is now offline.
```

```
toaster> vol destroy OLD
Are you sure you want to destroy this volume? yes
Volume 'OLD' destroyed.
```

Now Alice has tools that are even better, and she was able to safely test her ideas with the help of cloning. If she had wanted to be even safer, Alice could have kept the old volume around until she was sure that everything worked.

Cloning a FlexClone Volume

Alice continues working on the scenario tools for Gadget. Since she is happy with the progress she's made so far, she starts by cloning the stable volume to create a volume named "WorkInProgress." She makes her changes in the clone volume, but at some point gets stuck on a tricky part. She figures it out and locks in her progress by making another clone of her working code (named "WIP2"). Unfortunately she's now a bit behind schedule, so she asks Bob to help implement part of the code and clones her working volume for Bob to use (Bob uses "WIP3").

```
toaster> vol clone create WorkInProgress -b GadgetTools
...
toaster> vol clone create WIP2 -b WorkInProgress
...
toaster> vol clone create WIP3 -b WIP2
...

toaster> df -g
Filesystem          total    used   avail capacity  Mounted on
/vol/GadgetTools/   16GB   11GB   4GB    73%   /vol/GadgetTools/
/vol/GadgetTools/.snapshot  4GB    0GB    3GB    2%    
/vol/GadgetTools/.snapshot
/vol/WorkInProgress/ 16GB   11GB   4GB    73%   /vol/WorkInProgress/
/vol/WorkInProgress/.snapshot  4GB    0GB    3GB    2%    
/vol/WorkInProgress/.snapshot
/vol/WIP2/          16GB   11GB   4GB    3%    /vol/WIP2/
/vol/WIP2/.snapshot  4GB    0GB    3GB    2%    
/vol/WIP3/          16GB   11GB   4GB    73%   /vol/WIP3/
/vol/WIP3/.snapshot  4GB    0GB    3GB    1%    /vol/WIP3/.snapshot

toaster> df -Ag aggr1
Aggregate          total    used   avail capacity
aggr1              109GB   62GB   47GB   57%
```

```

aggr1/.snapshot          5GB    0GB    5GB    4%
toaster> snap list -V
Volume GadgetTools working...
  %/used  %/total  date      name
  -----
  2% ( 2%) 0% ( 0%) Oct 01 22:07 clone_WorkInProgress.1 (busy,vclone)
Volume WorkInProgress working...
  %/used  %/total  date      name
  -----
  2% ( 2%) 0% ( 0%) Oct 01 22:08 clone_WIP2.1 (busy,vclone)
  5% ( 2%) 0% ( 0%) Oct 01 22:07 clone_WorkInProgress.1
Volume WIP2 working...
  %/used  %/total  date      name
  -----
  2% ( 2%) 0% ( 0%) Oct 01 22:08 clone_WIP3.1 (busy,vclone)
  6% ( 3%) 0% ( 0%) Oct 01 22:08 clone_WIP2.1
Volume WIP3 working...
  %/used  %/total  date      name
  -----
  3% ( 3%) 0% ( 0%) Oct 01 22:08 clone_WIP3.1

```

A bit later both she and Bob have made great progress and are ready to test their new code. Bob e-mails his parts to Alice, who incorporates them in her code base. The code passes all the tests, and Alice wants to split off her clone ("WIP2") and replace the current production volume:

```

toaster> vol status WIP2
  Volume   State   Status          Options
    WIP2    online  raid_dp, flex  create_icode=on, convert_icode=on,
                                         guarantee=volume(disabled)
                                         Clone, backed by volume 'WorkInProgress', snapshot 'clone_WIP2.1'
                                         Volume has clones: WIP3
                                         Containing aggregate: 'aggr1'

```

```

toaster> vol clone split start WIP2
Fri Sep 24 00:27:37 GMT [waf1.vv.clone.split:notice]: Volume 'WIP2' has locked snapshots, not splitting.
vol clone split start: The volume has restricted snapshots

```

Alice tries to do this, but realizes that she can't because the clone Bob used ("WIP3") is still attached to her clone. She could have seen that by noticing the line that says "Volume has clones." WIP3 depends on a Snapshot copy currently contained in WIP2. Splitting off the clone would delete all the Snapshot copies in the clone, which would mean that WIP3 wouldn't have a backing Snapshot copy. Data ONTAP recognizes this and prevents accidents. Alice realizes that she doesn't need WIP3 anymore and simply destroys WIP3 before retrying the split:

```

toaster> vol offline WIP3
Fri Sep 24 00:33:16 GMT [cifs.terminationNotice:warning]:
CIFS: shut down completed: CIFS is disabled for volume WIP3.
Volume 'WIP3' is now offline.

```

```

toaster> vol destroy WIP3
Are you sure you want to destroy this volume? yes
Volume 'WIP3' destroyed.

```

```

toaster> vol clone split start WIP2

```

```

Fri Sep 24 00:33:29 GMT [wafl.scan.start:info]: Starting volume clone split on volume WIP2.
Clone volume 'WIP2' will be split from its parent.
Monitor system log or use 'vol clone split status' for progress.
Fri Sep 24 00:33:48 GMT [wafl.vol.add.spcGuar:info]: Space for volume WIP2 is guaranteed.
Fri Sep 24 00:33:50 GMT [wafl.scan.clone.split.complete:info]: Volume WIP2, clone split complete.

```

Alice then follows the same process used in the previous section to replace the production volume with the newly split clone.

Multiple FlexClone Volumes from a Single Parent FlexVol Volume

Now that the tools are in good shape, Alice needs to examine a few different scenarios. Each manipulates slightly different data sets as part of the simulation, so she makes three FlexClone volumes of the original data, all using the same backing Snapshot copy:

```
toaster> snap create WidgetData GSimSnap
```

```
toaster> vol clone create GSim1 -b WidgetData GSimSnap
```

```

Fri Sep 24 00:46:31 GMT [wafl.snaprestore.revert:notice]: Reverting volume GSim1 to a previous snapshot.
Fri Sep 24 00:46:32 GMT [wafl.vol.guarantee.clone:info]: Space for clone volume 'GSim1' is not guaranteed.
Creation of clone volume 'GSim1' has completed.

```

```
toaster> vol clone create GSim2 -b WidgetData GSimSnap
```

```

Fri Sep 24 00:46:55 GMT [wafl.snaprestore.revert:notice]: Reverting volume GSim2 to a previous snapshot.
Fri Sep 24 00:46:55 GMT [wafl.vol.guarantee.clone:info]: Space for clone volume 'GSim2' is not guaranteed.
Creation of clone volume 'GSim2' has completed.

```

```
toaster> vol clone create GSim3 -b WidgetData GSimSnap
```

```

Fri Sep 24 00:47:02 GMT [wafl.snaprestore.revert:notice]: Reverting volume GSim3 to a previous snapshot.
Fri Sep 24 00:47:03 GMT [wafl.vol.guarantee.clone:info]: Space for clone volume 'GSim3' is not guaranteed.
Creation of clone volume 'GSim3' has completed.

```

```
toaster> vol status
```

Volume	State	Status	Options
WidgetData	online	raid_dp, flex	create_icode=on, convert_icode=on
GSim1	online	raid_dp, flex	create_icode=on, convert_icode=on,
			guarantee=volume(disabled)
GSim2	online	raid_dp, flex	create_icode=on, convert_icode=on,
			guarantee=volume(disabled)
GSim3	online	raid_dp, flex	create_icode=on, convert_icode=on,
			guarantee=volume(disabled)

```
toaster> vol status WidgetData
```

Volume	State	Status	Options
WidgetData	online	raid_dp, flex	create_icode=on,
			convert_icode=on

Volume has clones: GSim3, GSim2, GSim1

Containing aggregate: 'aggr1'

Alice modifies each clone to be at the desired starting point, mounts each clone on a different set of simulation engines, and lets them work for a while. Her application is not particularly sensitive about permissions and security. If it was, Alice might first mount each FlexClone volume on an administrative host, change file permission or ownership to match the authorized users, and then remount them to the

application servers she made the clones for. Please see the [Security, Permissions, and SnapLock Compliance Volumes](#) section for a more in-depth discussion of the issues.

When complete, each simulation changed about 5% of the data. All together the physical storage required was 115% of the original data set. Compare that with 400% and lots of copy time required if Alice had made three copies of the data instead. Alice archives a compressed copy of each final data set to a NearStore® appliance and saves the simulation results to present to Gadget at their next meeting. Then she destroys the three FlexClone volumes and releases the 15% extra production space used during the simulation.

SnapMirror and FlexClone Volumes

Alice needs a hand with a special part of the Widget project looking at various trends around the European Union. She ropes in Clive from the London office to provide his expertise. Alice arranges for a SnapMirror copy of her work to be sent to Clive's filer (named "blender") in London. They have worked together in the past, and their filers can access each other via SnapMirror because they have the correct entries in their respective `/etc/snapmirror.conf` files. The SnapMirror copy is read-only, so Clive will clone the volume to obtain an instantly writable version to work with. Because the SnapMirror destination is read-only, Clive will not be able to create a Snapshot copy to use as a FlexClone volume backing store on his end. Alice also arranges for a consistent Snapshot copy of her volume that Clive can use. Here is what Clive does over in London:

```
blender> vol create MyGTools aggr1 20m
Creation of volume 'MyGTools' with size 20m on containing aggregate
'aggr1' has completed.

blender> vol restrict MyGTools
Fri Sep 24 19:33:31 GMT [cifs.terminationNotice:warning]
CIFS: shut down completed: CIFS is disabled for volume MyGTools.
Volume 'MyGTools' is now restricted.

blender> snapmirror initialize -S 192.168.0.111:GadgetTools -w MyGTools
Transfer started.
Monitor progress with 'snapmirror status' or the snapmirror log.
Fri Sep 24 19:36:13 GMT [wafl.vol.guarantee.replica:info]: Space for replica volume 'MyGTools' is not
guaranteed.
Transfer successful.

blender> df -g
Filesystem          total    used   avail   capacity      Mounted on
/vol/vol0/          253GB  56GB  196GB   22%      /vol/vol0/
/vol/vol0/.snapshot  0GB    0GB   0GB    ---%      /vol/vol0/.snapshot
/vol/MyGTools/      16GB   11GB  4GB    73%      /vol/MyGTools/
/vol/MyGTools/.snapshot  4GB    0GB   3GB    2%       /vol/MyGTools/.snapshot

blender> vol status MyGTools
      Volume      State      Status      Options
      MyGTools    online    raid_dp, flex  snapmirrored=on, create_icode=on,
                           snapmirrored  convert_icode=on, fs_size_fixed=on,
                           guarantee=volume(disabled)
                           Containing aggregate: 'aggr1'
```

Using the Data ONTAP commands above, Clive first creates a FlexVol volume on his filer in London to receive the SnapMirror image. He then restricts the volume as SnapMirror requires. When the destination volume is ready, he initiates the SnapMirror synchronization with the **snapmirror initialize** command. Finally, he checks that the new volume is available and working on his filer. The next step is to create a writable copy of the read-only SnapMirror volume:

```
blender> snap list
Volume MyGTools working...
  %/used  %/total  date      name
  -----  -----  -----
  0% ( 0%)  0% ( 0%)  Sep 24 19:35  blender(0099913498)_MyGTools.1
  0% ( 0%)  0% ( 0%)  Sep 24 19:00  forClive.1
  1% ( 0%)  0% ( 0%)  Sep 24 00:00  nightly.0

blender> vol clone create cl_GTools -b MyGTools forClive.1
Fri Sep 24 19:38:22 GMT [wafl.snaprestore.revert:notice]: Reverting volume cl_GTools to a previous
snapshot.
Fri Sep 24 19:38:23 GMT [wafl.vol.guarantee.clone:info]: Space for clone volume 'cl_GTools' is not
guaranteed.
Creation of clone volume 'cl_GTools' has completed.

blender> df -g
Filesystem      total    used   avail  capacity  Mounted on
/vol/vol0/        253GB   56GB  196GB   22%      /vol/vol0/
/vol/vol0/.snapshot  0GB    0GB   0GB    ---%      /vol/vol0/.snapshot
/vol/MyGTools/     16GB   11GB   4GB    73%      /vol/MyGTools/
/vol/MyGTools/.snapshot  4GB    0GB   3GB    2%       /vol/MyGTools/.snapshot
/vol/cl_GTools/     16GB   11GB   4GB    73%      /vol/cl_GTools/
/vol/cl_GTools/.snapshot  4GB    0GB   3GB    1%       /vol/cl_GTools/.snapshot

blender> cifs shares -add Tools /vol/cl_GTools
```

Clive uses the existing Snapshot copy Alice created for him ("forClive.1") to back his new clone. Alice needs to be careful not to delete the Snapshot copy Clive is using while he has a clone dependent on that Snapshot copy. After checking that the new clone volume is available on his system, Clive makes it accessible to his workstation and proceeds to make the necessary improvements. After he is done, he likes the results so much he splits off the clone to use as part of his toolbox:

```
blender> df -g
Filesystem      total    used   avail  capacity  Mounted on
/vol/vol0/        253GB   56GB  196GB   22%      /vol/vol0/
/vol/vol0/.snapshot  0GB    0GB   0GB    ---%      /vol/vol0/.snapshot
/vol/MyGTools/     16GB   11GB   4GB    73%      /vol/MyGTools/
/vol/MyGTools/.snapshot  4GB    0GB   3GB    2%       /vol/MyGTools/.snapshot
/vol/cl_GTools/     16GB   10GB   5GB    67%      /vol/cl_GTools/
/vol/cl_GTools/.snapshot  4GB    2GB   1GB    72%      /vol/cl_GTools/.snapshot
```

```
blender> vol clone split start cl_GTools
Fri Sep 24 19:54:55 GMT [wafl.scan.start:info]: Starting volume clone split on volume cl_GTools.
Clone volume 'cl_GTools' will be split from its parent.
Monitor system log or use 'vol clone split status' for progress.
Fri Sep 24 19:55:13 GMT [wafl.vol.add.spcGuar:info]: Space for volume cl_GTools is guaranteed.
Fri Sep 24 19:55:14 GMT [wafl.scan.clone.split.complete:info]: Volume cl_GTools, clone split complete.
```

```
blender> vol rename cl_GTools Clive_GTools
'cl_GTools' renamed to 'Clive_GTools'
```

After integrating his changes, Clive e-mails Alice the deltas, which turn out to be relatively small. If there were lots of changes, Clive could have arranged for a SnapMirror copy of his clone to be sent back to Alice's filer, where she could have broken off the mirror to replace her original data set with the improved one.

SnapMirror can also be used within the same filer to create a FlexClone volume on a different aggregate than the parent FlexVol volume. This may be useful for space management, migration, or whatever reason an administrator needs to move or duplicate a FlexVol volume on another aggregate.

LUNs and FlexClone Volumes

Meanwhile, Dave has been tasked with rolling out an upgrade for SI's internal CRM system, which runs on top of an Oracle Database. The database is stored on a clustered NetApp filer and delivered to the server on top of iSCSI LUNs. SI's CRM data is still pretty small, but needs to perform quickly. Here's what the system starts out with:

```
ScenarioCRM> lun show -v
/vol/CRMdata/lun0      39.2m (41126400)  (r/w, online, mapped)
  Serial#: nBGbm3tClmx6
  Share: none
  Space Reservation: enabled
  Multiprotocol Type: windows
  Maps: Scenario=1
/vol/CRMfiles/lun0      39.2m (41126400)  (r/w, online, mapped)
  Serial#: nBGbm3tCliN-
  Share: none
  Space Reservation: enabled
  Multiprotocol Type: windows
  Maps: Scenario=0

ScenarioCRM> df -g
Filesystem      total  used  avail capacity  Mounted on
/vol/vol0/          253GB  57GB  196GB   23%    /vol/vol0/
/vol/vol0/.snapshot  0GB   0GB   0GB   ---%    /vol/vol0/.snapshot
/vol/CRMfiles/      42GB  40GB  1GB   97%    /vol/CRMfiles/
/vol/CRMfiles/.snapshot  0GB   0GB   0GB   ---%    /vol/CRMfiles/.snapshot
/vol/CRMdata/       42GB  40GB  1GB   97%    /vol/CRMdata/
/vol/CRMdata/.snapshot  0GB   0GB   0GB   ---%    /vol/CRMdata/.snapshot

ScenarioCRM> df -Ag
Aggregate      total  used  avail capacity
aggro          256GB  254GB  1GB   99%
aggro/.snapshot 13GB   0GB   13GB  3%
aggr1          109GB  85GB  23GB  78%
aggr1/.snapshot 5GB   0GB   5GB   1%
```

Dave chooses to build the CRM LUNs on top of flexible volumes to take advantage of the performance provided by the disk spindles in the aggregate within the NetApp storage system. The output indicates that there are currently two (2) 39.2GB iSCSI LUNS mapped to an initiator group named "Scenario". The initiator group Scenario represents the LUN mapping for the LUNs on the storage system to the

host running Oracle. The LUNs reside in two flexible volumes of about 42GB in size each. The flexible volumes have automatic Snapshot copies turned off and snap reserve set to 0% because the automatic Snapshot copies may not capture the LUN in a consistent state due to the requirement to coordinate the Snapshot copy with the host file system to ensure the data is consistent. Using SnapDrive™ is an example of a way to integrate the host file system with a NetApp storage Snapshot copy. Also, Dave wants all the space in the volume dedicated for the LUNs and any cloning or mirroring that may be required. In this example note that the aggregate containing the LUN volumes is getting full and has space available.

The FlexVol cloning feature makes rolling out upgrades easier. Dave can put the CRM database briefly into a delayed write mode (for example, Oracle hot backup or DB2 write suspend), and make two FlexClone volumes: one volume with the CRM database and the other volume holding the CRM binaries. Since the FlexClone volumes are created in a few seconds, the database doesn't stay in hot backup mode for long. However, since the FlexClone volumes require access to Snapshot copies in the parent FlexVol volume, Dave needs to do some additional preparation to make everything move smoothly during the maintenance window. He also realizes that aggr1 may not have space to hold all the changes generated during the upgrade and testing process. Luckily for Dave, he's running Data ONTAP 7G on his NetApp storage systems and can make all the necessary preparations without interrupting the production environment. Here's what Dave does:

```
ScenarioCRM> aggr status aggr1 -v
      Aggr      State    Status      Options
      aggr1    online   raid_dp, aggr  nosnap=off, raidtype=raid_dp, raidsize=10,
                                         snapmirrored=off, resyncsnaptime=60,
                                         fs_size_fixed=off, snapshot_autodelete=on,
                                         lost_write_protect=on
      Volumes: CRMfiles, CRMdata
      Plex /aggr1/plex0: online, normal, active
      RAID group /aggr1/plex0/rg0: normal
```

```
ScenarioCRM> aggr add aggr1 10
Note: preparing to add 8 data disks and 2 parity disks.
Continue? ([y]es, [n]o, or [p]review RAID layout) p
The RAID group configuration will change as follows:
      RAID Group      Current      NEW
      -----
      /aggr1/plex0/rg0  10 disks    10 disks
      /aggr1/plex0/rg1                10 disks
Continue? ([y]es, [n]o, or [p]review RAID layout) y
...
Addition of 10 disks to the aggregate has completed.
```

```
ScenarioCRM> aggr status aggr1 -v
      Aggr      State    Status      Options
      aggr1    online   raid_dp, aggr  nosnap=off, raidtype=raid_dp, raidsize=10,
                                         snapmirrored=off, resyncsnaptime=60,
                                         fs_size_fixed=off, snapshot_autodelete=on,
                                         lost_write_protect=on
      Volumes: CRMfiles, CRMdata
      Plex /aggr1/plex0: online, normal, active
      RAID group /aggr1/plex0/rg0: normal
      RAID group /aggr1/plex0/rg1: normal
```

```
ScenarioCRM> df -Ag
Aggregate      total      used      avail      capacity
```

aggr0		256GB	254GB	1GB	99%
aggr0/.snapshot	13GB	0GB	13GB	3%	
aggr1		109GB	85GB	133GB	39%
aggr1/.snapshot	5GB	0GB	5GB	1%	

Dave double-checks the raidsize defaults on aggr1 and dynamically grows it by adding another RAID group of 10 disks from available spares. He then checks that the aggregate now contains a second RAID group (rg1) and there is plenty of space for Snapshot copies and FlexClone volumes. Since that's now taken care of, Dave dynamically grows the two flexible volumes containing the CRM LUNs. Since Dave is just adjusting the size of the volumes on the filer, all this happens transparently to the host system, which is using the LUNs. Data ONTAP also allows the LUNs to be resized, but that requires integration with the host operating system, using additional steps or software such as SnapDrive.

ScenarioCRM> **vol size CRMfiles 60m**

vol size: Flexible volume 'CRMfiles' size set to 60m.

ScenarioCRM> **vol size CRMdata 60m**

vol size: Flexible volume 'CRMdata' size set to 60m.

ScenarioCRM> **df -g**

Filesystem	total	used	avail	capacity	Mounted on
/vol/vol0/		253GB	57GB	195GB	23% /vol/vol0/
/vol/vol0/.snapshot	0GB	0GB	0GB	---	/vol/vol0/.snapshot
/vol/CRMfiles/	60GB	40GB	19GB	85%	/vol/CRMfiles/
/vol/CRMfiles/.snapshot		0GB	0GB	0GB	---
/vol/CRMdata/	60GB	40GB	19GB	85%	/vol/CRMdata/
/vol/CRMdata/.snapshot		0GB	0GB	0GB	---

ScenarioCRM> **df -Ag aggr1**

Aggregate	total	used	avail	capacity	
aggr1		218GB	121GB	97GB	56%
aggr1/.snapshot	11GB	0GB	11GB	0%	

Preparations completed, Dave is ready to make the actual FlexClone volumes. During the next maintenance window, he can put the database in hot backup mode and make the FlexClone volumes in less than a minute:

ScenarioCRM> **vol clone create NEWfiles -b CRMfiles**

Fri Sep 24 22:47:54 GMT [wafl.snaprestore.revert:notice]: Reverting volume NEWfiles to a previous snapshot.

Fri Sep 24 22:47:55 GMT [wafl.vol.guarantee.clone:info]: Space for clone volume 'NEWfiles' is not guaranteed.

Creation of clone volume 'NEWfiles' has completed.

Fri Sep 24 22:47:56 GMT [lun.newLocation.offline:warning]:

LUN /vol/NEWfiles/lun0 has been taken offline to prevent map conflicts after a copy or move operation.

ScenarioCRM> **vol clone create NEWdata -b CRMdata**

Fri Sep 24 22:48:20 GMT [wafl.snaprestore.revert:notice]: Reverting volume NEWdata to a previous snapshot.

Fri Sep 24 22:48:21 GMT [wafl.vol.guarantee.clone:info]: Space for clone volume 'NEWdata' is not guaranteed.

Creation of clone volume 'NEWdata' has completed.

Fri Sep 24 22:48:21 GMT [lun.newLocation.offline:warning]:

LUN /vol/NEWdata/lun0 has been taken offline to prevent map conflicts after a copy or move operation.

ScenarioCRM> **df -g**

Filesystem	total	used	avail	capacity	Mounted on
/vol/vol0/		253GB	57GB	195GB	23% /vol/vol0/
/vol/vol0/.snapshot	0GB	0GB	0GB	---	/vol/vol0/.snapshot
/vol/CRMfiles/	60GB	40GB	19GB	85%	/vol/CRMfiles/
/vol/CRMfiles/.snapshot		0GB	0GB	0GB	---
/vol/CRMdata/	60GB	40GB	19GB	85%	/vol/CRMdata/
/vol/CRMdata/.snapshot		0GB	0GB	0GB	---

```

/vol/vol0/          253GB  57GB   195GB  23%   /vol/vol0/
/vol/vol0/.snapshot 0GB    0GB    0GB    ---%   /vol/vol0/.snapshot
/vol/CRMfiles/     48GB   40GB   7GB    85%   /vol/CRMfiles/
/vol/CRMfiles/.snapshot 12GB   0GB    12GB   0%    /vol/CRMfiles/.snapshot
/vol/CRMdata/      48GB   40GB   7GB    85%   /vol/CRMdata/
/vol/CRMdata/.snapshot 12GB   0GB    12GB   0%    /vol/CRMdata/.snapshot
/vol/NEWfiles/     48GB   42GB   5GB    90%   /vol/NEWfiles/
/vol/NEWfiles/.snapshot 12GB   0GB    11GB   0%    /vol/NEWfiles/.snapshot
/vol/NEWdata/      48GB   43GB   4GB    90%   /vol/NEWdata/
/vol/NEWdata/.snapshot 12GB   0GB    11GB   0%    /vol/NEWdata/.snapshot

ScenarioCRM> df -Ag
Aggregate          total  used   avail  capacity
aggro              256GB 254GB 1GB    99%
aggro/.snapshot    13GB  0GB   13GB   3%
aggr1              218GB 121GB 97GB   56%
aggr1/.snapshot    11GB  0GB   11GB   0%

```

Dave knows that the cloning processes duplicated all the LUN details along with the volumes. Notice that Data ONTAP is intelligent enough to take the cloned LUNs offline to prevent any mapping errors. If he tries to bring the LUNs online before making the changes, Data ONTAP will also issue a warning and prevent the operation until the conflict is eliminated. Dave needs to make the necessary mapping adjustments to take advantage of the cloned LUNs without disturbing the production work. He creates a new initiator group for the testing, maps the cloned LUNs to the new group, and removes the duplicate mappings (seen in *red italics* below) from the production group. Mappings are additive, so old mappings need to be explicitly removed with the `lun unmap` command. When everything is squared away, he brings the cloned LUNs online:

```

ScenarioCRM> igrup show
Scenario (iSCSI) (ostype: windows):
iqn.1991-05.com.microsoft:miroslav-lxp

```

```
ScenarioCRM> igrup create -i -t windows Testing iqn.1991-05.com.microsoft:miroslav-lxp
```

```

ScenarioCRM> lun show -m
LUN path      Mapped to  LUN ID  Protocol
-----
```

/vol/CRMdata/lun0	Scenario	1	iSCSI
/vol/CRMfiles/lun0	Scenario	0	iSCSI
/vol/NEWdata/lun0	<i>Scenario</i>	<i>1</i>	iSCSI
/vol/NEWfiles/lun0	<i>Scenario</i>	<i>0</i>	iSCSI

```

ScenarioCRM> lun map /vol/NEWfiles/lun0 Testing 10
ScenarioCRM> lun map /vol/NEWdata/lun0 Testing 11
ScenarioCRM> lun unmap /vol/NEWfiles/lun0 Scenario
Fri Sep 24 23:04:30 GMT [lun.map.unmap:info]: LUN /vol/NEWfiles/lun0 unmapped from initiator group
Scenario
ScenarioCRM> lun unmap /vol/NEWdata/lun0 Scenario
Fri Sep 24 23:04:43 GMT [lun.map.unmap:info]: LUN /vol/NEWdata/lun0 unmapped from initiator group
Scenario

```

```

ScenarioCRM> lun show -m
LUN path      Mapped to  LUN ID  Protocol
-----
```

/vol/CRMdata/lun0	Scenario	1	iSCSI
/vol/CRMfiles/lun0	Scenario	0	iSCSI

```

/vol/NEWdata/lun0      Testing      11      iSCSI
/vol/NEWfiles/lun0     Testing      10      iSCSI

ScenarioCRM> lun online /vol/NEWfiles/lun0
ScenarioCRM> lun online /vol/NEWdata/lun0
ScenarioCRM> lun show
/vol/CRMdata/lun0      39.2m (41126400)  (r/w, online, mapped)
/vol/CRMfiles/lun0     39.2m (41126400)  (r/w, online, mapped)
/vol/NEWdata/lun0      39.2m (41126400)  (r/w, online, mapped)
/vol/NEWfiles/lun0     39.2m (41126400)  (r/w, online, mapped)

```

He maps the cloned LUNS to a testing server via iSCSI and performs the upgrade using the cloned copies. After extensive testing and a couple of minor tweaks, Dave is ready to move the upgraded system into production. He's confident that everything will work because he has been testing with the same database contents as were being used in production just the other day. (In fact, if he were paranoid, he'd make another clone of the database for the final test.)

Dave destroys the cloned database LUN volume because he no longer needs it for testing. He then splits off the upgraded CRM binaries volume and uses it to replace the current production volume during a scheduled maintenance window. He knows he can keep the maintenance windows short because everything is already tested, and all he needs to do is swap volume names between the upgraded volume and the current production volume, plus change the LUN mappings to reflect the volume changes.

Another methodology that may work better for some organizations is to use the FlexClone volumes to develop and thoroughly test a set of patches/deltas that can be applied to the production volume during a maintenance window. The details of how that may be accomplished are very application-dependent and are left as an exercise for the reader.

Scenario Summary

This wraps up our story about Scenario Inc. and Alice, Bob, Clive, and Dave. It provided an in-depth demonstration of the benefits FlexClone volumes bring to real-world situations. Through the extended example, we hope you learned how to:

- >Create FlexClone volumes, including making multiple FlexClone volumes from a single parent, and FlexClone volumes of an existing FlexClone volume
- Split off a FlexClone volume from the parent to create a fully independent flexible volume
- Combine the benefits of FlexClone technology with SnapMirror to create writable FlexClone volumes across arbitrary distances
- Clone LUNs using FlexClone volumes for additional space management benefits relative to traditional LUN cloning
- Use Data ONTAP commands needed to manage FlexClone volumes and related tasks
- Manage volume and aggregate size to make sure FlexClone operations always succeed

The preceding section can be used as a way of understanding the full landscape of FlexClone benefits, as a refresher of specific FlexClone operations, or as a guided tutorial that filer administrators can try with their own test filer.

FlexClone Performance

The performance of FlexClone volumes is nearly identical to the performance of flexible volumes. This is thanks to the way cloning is tightly integrated with WAFL and the filer architecture. Unlike other implementations of cloning technology, FlexClone volumes are implemented as a simple extension to existing core mechanisms.

The impact of cloning operations on other system activity should also be relatively light and transitory. The FlexClone create operation is nearly identical to creating a Snapshot copy. Some CPU, memory, and disk resources are used during the operation, which usually completes in seconds. The clone metadata is held in memory like a regular volume, so the impact on filer memory consumption is identical to having another volume available. After the clone creation completes, all ongoing accesses to the clone are nearly identical to accessing a regular volume.

Splitting the clone to create a fully independent volume also uses resources. While the split is occurring, free blocks in the aggregate are used to copy blocks shared between the parent and the clone. This incurs disk I/O operations and can potentially compete with other disk operations in the aggregate. The copy operation also uses some CPU and memory resources, which may impact the performance of a fully loaded filer. Data ONTAP addresses these potential issues by completing the split operation in the background and sets priorities in a way that does not significantly impact foreground operations. It is also possible to manually stop and restart the split operation if some critical job requires the full resources of the filer.

The final area to consider is the impact on disk usage from frequent operations where FlexClone volumes are split off and used to replace the parent FlexVol volume. The split volume is allocated free blocks in the aggregate, taking contiguous chunks as they are available. If there is lots of free space in the aggregate, the blocks allocated to the split volume should be mostly contiguous. If the split is used to replace the original volume, the blocks associated with the destroyed original volume will become available and create a potentially large free area within the aggregate. That free area should also be mostly contiguous. In cases where many simultaneous volume operations reduce contiguous regions for the volumes, Data ONTAP 7G implements brand new block reallocation functionality. The new **reallocate** command makes defragmentation and sequential reallocation even more flexible and effective. It reduces any impact of frequent clone split and replace operations, as well as optimizes performance after other disk operations (e.g., adding disks to an aggregate) that may unbalance block allocations. For additional information, please see the [Data ONTAP Command Reference and Storage Management Guide](#).

Best Practices for Success with FlexClone Volumes

The extended example in the previous sections provided a guided tour of FlexClone functionality and practices. This section distills some of the best practices around FlexClone volumes and integrates them with related applications.

Naming Conventions

Since a FlexClone volume can be treated just like a FlexVol volume indefinitely, it makes sense to name the volume based on its use. In the previous example, the cloned volumes were used for the “Gadget” project, so the FlexClone volumes were named “*GadgetTools*” and “*GadgetData*.” Since there were only a few volumes and the plan was to split off the clone at some point, this naming scheme made the most sense. For situations where there are many cloned volumes on the system and the FlexClone volumes are expected to continue indefinitely, use names that call out the clone relationship. One simple naming convention is to prepend the characters “*cl_*” in front of the volume name. For example, use “*cl_GadgetTools*” for situations where it makes sense to explicitly mark the volume as a clone.

The same logic applies for naming the backing Snapshot copy. Data ONTAP names the automatic clone Snapshot copy based on the name of the clone volume. For example, “**vol clone create GadgetTools -b WidgetTools**” would create a Snapshot copy named “*clone_GadgetTools.1*” in the “*WidgetTools*” volume and mark the Snapshot copy as being busy backing a clone. That way, it’s clear that the Snapshot copy supports a clone volume and which volume it is backing. If a Snapshot copy for a clone is created manually, use a name that will highlight the relationship between the Snapshot copy and the clone volumes depending on it. An administrator can easily rederive all this information using Data ONTAP commands, but good naming conventions make it clear from the start and reduce mistakes.

Space Utilization and Reservations

WAFL is designed to provide good layout of data on disks. The architecture takes advantage of temporal locality to create good spatial locality for blocks. FlexClone volumes benefit from this WAFL feature, just like all other volumes, and administrators do not need to worry about the physical disk layout of clone volumes.

Nonetheless, it makes sense to pay attention to the amount of physical space available for clone volume operations. Space guarantees are disabled for FlexClone volumes in Data ONTAP 7G. That means it is possible for changes in the clone volume to fail and generate “no space” errors. Any blocks that are different between the parent and the clone backing Snapshot copy are accounted for in the Snapshot copy area associated with the parent. That space is guaranteed and managed using the **vol size** and **snap reserve** commands. It is possible for the Snapshot copy area to become overcommitted and more than 100% full, but it’s better for the administrator to explicitly make the necessary space available. Blocks that are written to the clone volume are allocated from free space in the aggregate. The amount of space allocated to the FlexClone volume is also limited by the volume size (just like any other volume). That means that the available space in the clone is the MINIMUM of:

FlexClone size minus the amount of space used in the FlexClone volume

Free space in the containing aggregate

Even though the clone volume is considered to be only partially full according to the volume size and amount used, if there is not enough space available in the aggregate, then the “Avail” column of the **df** command will display the smaller value. That is one way for the administrator to track the usage. The overcommitment may also become apparent when attempting to split the clone volume from the parent. Enough space is required in the aggregate to duplicate any shared blocks, and Data ONTAP will not allow the operation to proceed until enough space is available.

The best practice is to be aware of how space reservations work and monitor available space when working with FlexClone volumes. Alerts can be set in many tools for notifying the administrator when space is becoming exhausted. Data ONTAP 7G makes it simple to resize volumes and aggregates, so the administrator has more control over available space than ever before. Running out of space is a minor inconvenience, since it’s possible to dynamically grow volumes, Snapshot copy areas, and even aggregates.

Security, Permissions, and SnapLock Compliance Volumes

Organizations and applications are increasingly more aware of security and access permissions. This may be motivated by the desire to prevent access to sensitive records, protect production data from accidental corruption, secure mission-critical systems from cyberthreats, etc. When a FlexVol volume is cloned, the FlexClone volume contains data with the exact same ownership and permissions as the parent. Any users and applications that could access the FlexClone volume would also be able to access the parent. As a best practice, it is better to use separate user accounts for development/testing and production. That means the administrator needs some way to allow access to the FlexClone volume but not the parent.

One method is to first mount/map each FlexClone volume on an administrative host, change file permissions and/or ownership to match the authorized development/test users, and then remount the FlexClone volumes to the appropriate servers. This can all be done without affecting the production application and may be considered a necessary step to ensure security. The only downside is that the process is manual and may take some time to complete for large FlexClone volumes.

Another method is to use access management options associated with making the FlexClone volume available from the filer. The **cifs access** command sets or modifies the share-level access control list (“ACL”) of a share. It may also be used to set Windows® machine account access to the share when Kerberos is used. The **exportfs** command can be used with NFS to export directories and files, determine the current exports, check the access cache for an export, and determine the actual storage path of an export. When a FlexVol volume is cloned, the FlexClone volume is exported via NFS identically to the parent. Access to the parent via CIFS, iSCSI, or FCP is not automatically cloned, and additional steps are required to make the FlexClone volume available via those protocols. The administrator would need to modify access permissions using either **exportfs** or **cifs access** to ensure that appropriate security policies are enforced for accessing the FlexClone volumes and the parent volumes.

It is possible to create a FlexClone volume based on a SnapLock™ compliance FlexVol volume. We recommend *avoiding* this practice unless the business needs for doing so are compelling. The reason is that once created, the FlexClone volume cannot be deleted until the retention periods for all files contained in the original SnapLock compliance volume have expired. While no additional storage is used, the FlexClone volume counts against the maximum number of volumes available on the filer.

Snapshot Management

One of the main best practice considerations for Snapshot copy management is naming conventions. Data ONTAP will create automatic Snapshot copies for FlexClone volumes with meaningful names. It will also indicate through various commands when a Snapshot copy is associated with a clone volume. For manually created Snapshot copies, it is a good idea to use a Snapshot copy name that makes it clear that the Snapshot copy is being used for a clone and which clone or task it is being used for. Since Snapshot copies cannot be renamed, the administrator will need to stay aware of any previously existing Snapshot copies that are also being used to back clone volumes. Lastly, be aware that Data ONTAP locks any Snapshot copies used to back clone volumes until the clone is either split off or destroyed. Any disk blocks associated with the Snapshot copy volume will remain locked and cannot be reused until the Snapshot copy is deleted.

Data ONTAP does not automatically delete backing Snapshot copies in the parent when the FlexClone volume is split or destroyed. There are a number of good reasons for this, but it basically comes down to the fact that it's hard to guess when the Snapshot copy may be needed for future work. As a best practice, it is up to the administrator to review existing Snapshot copies after FlexClone volumes are removed and choose which ones to delete. Deleting a Snapshot copy is easily done with the **snap delete** command, and the **snap list** command displays existing Snapshot copies.

Cloning and SnapMirror

Today, global enterprises need to protect and quickly recover data in the event of natural or human-caused disasters, operator errors, or technology and application failures. They also need an efficient way to distribute data to remote locations. Without an effective data protection and distribution strategy, operations can be brought to a standstill, resulting in millions of dollars of lost revenue. SnapMirror software makes this all possible in a powerful, yet easy-to-use and administer package. NetApp SnapMirror software delivers the disaster recovery and data distribution solution that today's global enterprises need. By replicating data at high speeds over a LAN or a WAN, SnapMirror software provides the highest possible data availability and fastest recovery for mission-critical applications.

Combined with FlexClone technology, it is now possible to make even greater use of SnapMirror volumes, whether used to leverage existing investment—turning disaster recovery investments into greater productivity—or new initiatives such as distributed development around the world and around the clock. Best practices for the two technologies are still evolving, and the example above demonstrated some of the necessary steps. Each organization's needs are unique. We invite readers to imagine how their environment might benefit from SnapMirror and FlexClone volumes.

Backups of FlexClone Volumes

FlexClone volumes can be treated like regular volumes for backup applications. The same best practices for backing up volumes would also apply to FlexClone volumes. Software features such as SnapVault™ and SnapManager® will work with FlexClone volumes, since it's possible to create Snapshot copies of a clone. Other backup software will also see clone volumes just like any other volume.

The main consideration is that while the FlexClone volumes use space only for blocks changed in the aggregate, a full backup would end up making a copy of all the data seen in the volume. That might result in greater usage of backup media than anticipated if the expectation is that only changed blocks are backed up. To back up only the changed blocks, the backup application needs to be aware of

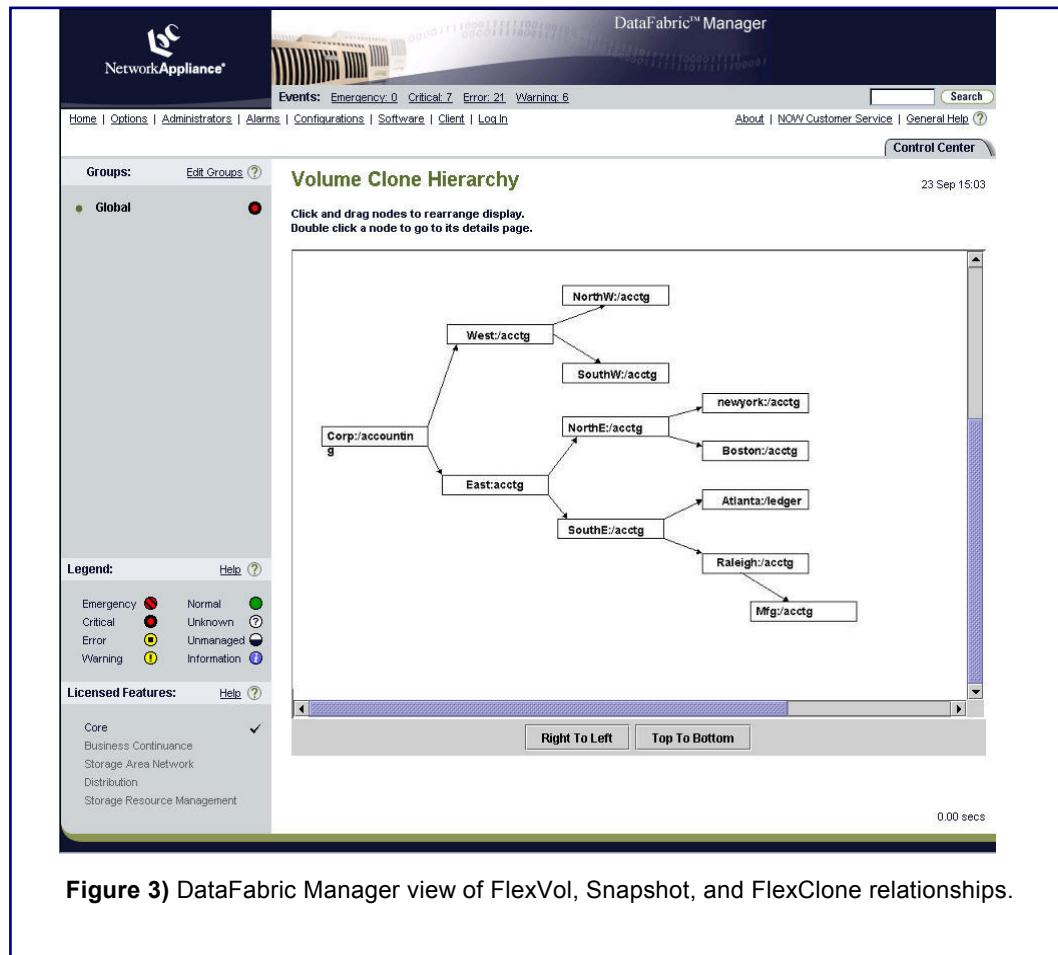
FlexClone. It is wise to confirm with the backup application vendor that the application is aware of FlexClone volumes and capable of backing up only the changed blocks.

Lastly, performance of the containing aggregate may be affected during the backup process. All the blocks associated with the volume could be read, competing with any other applications for the filer and disk resources. As for backups of regular volumes, the administrator should expect some performance impact and schedule the backup appropriately.

GUI Management and FlexClone Volumes

The NetApp DataFabric Manager application offers an efficient and powerful system for managing a storage infrastructure consisting of NetApp storage systems and NetCache® appliances. The FilerView tool, available on every NetApp storage appliance, enables IT administrators to use Web browsers to access a consistent, easy-to-use graphical user interface (GUI) for everyday administration tasks.

Neither FilerView nor DataFabric Manager currently offers options for creating and splitting FlexClone volumes. However, once FlexClone volumes are created, they can be managed like any other volume via FilerView. The related Snapshot copies and LUNs can also be managed. In addition, DataFabric Manager provides intuitive visualization tools for examining the relationships between volumes, Snapshot copies, and FlexClone volumes. Figure 3 is an example.



FlexClone Volumes for LUN Cloning

Data ONTAP 7G provides two mechanisms for making LUN clones. The original method, available prior to Data ONTAP 7G, uses the **lun create -b** and **lun clone** commands to turn a writable LUN Snapshot copy into a cloned LUN. This method fully duplicates all the blocks in the original LUN during the LUN cloning operation. If the original LUN is 100GB in size, the LUN clone also requires 100GB of storage. Data ONTAP 7G changes the syntax of the LUN cloning commands to more closely match the new vol cloning syntax. Table 3 shows the new and old syntax.

Table 4) Comparison of LUN cloning syntax for Data ONTAP 7G and prior versions.

Data ONTAP 7G	Before Data ONTAP 7G
lun clone create clone_LUNpath [-o noreserve] -b parent_LUNpath parent_snap	lun create -b snapshot_LUN_path [-o noreserve] LUN_path (note: creates backing store snapshot LUN)
lun clone split start LUN_path	lun clone start LUN_path
lun clone split status LUN_path	lun clone show LUN_path
lun clone split stop LUN_path	lun clone stop LUN_path

The older method is still available, along with the new option of using the FlexClone method for creating a writable clone of the FlexVol volume containing the LUNs to be cloned. The FlexClone method only uses physical storage for those blocks that changed in the clone relative to the base Snapshot copy. If the original LUN is 100GB, and only 1GB changed in the cloned LUN, then only 1GB of additional space is used. This is a considerable saving in terms of space and duplication time. Of course, if the FlexClone volume is at some point split off from the parent, all the shared blocks will be duplicated. There are many applications that will never require that final split and can get away with less physical space than traditional LUN clones require.

With two LUN cloning methods available, a natural question is which method to use for a given situation. Let's take a closer look at some additional considerations.

The main consideration for creating LUN clones is that the LUN is in a consistent state at the moment the base Snapshot copy is created. One standard best practice is to disable automatic Snapshot copies on volumes containing LUNs. If the LUN is used to store a database or file system, some additional tool at the host level is needed to put the LUN in a consistent state. Many such tools exist, and most can temporarily do so in a way that does not disrupt host operations. Once the LUN is in a consistent state, the steps described in the extended example can be used to clone the LUN and make it accessible. During the process, it is important to make sure that LUN mapping for the clone is updated to not conflict with the original LUNs. Data ONTAP helps prevent errors by automatically taking any cloned LUNs offline until the administrator brings them online manually.

LUN clones, though they share blocks, have an extra level of indirection on reads (assuming they have not been split). This is not the case with LUNs on FlexClone volumes. There is no additional redirection on reads. Snapshot copies created in the volume after a LUN clone has been created will also "lock" the base Snapshot copy from which the LUN clone was created. Even after the clone is split, a user will be unable to delete the base Snapshot copy unless all Snapshot copies created since the clone was created—but before it was split—have been deleted. FlexClone volumes do not have this behavior, since Snapshot copies are created in the FlexClone volume and are not shared with the parent FlexVol volume after clone creation.

The storage administrator should provision storage such that all applications store their data in individual flexible volumes to take greatest advantage of FlexClone capabilities. That improves the granularity available for cloning. However, if the goal is to clone all the data (possibly in separate LUNs) in a single operation, and all LUNs are stored in one flexible volume, then using the FlexClone feature makes the most sense.

A FlexClone volume can create writable volumes even from a read-only volume SnapMirror destination. Therefore, it is now possible use LUNs in flexible volumes on secondary or disaster recovery sites for purposes such as backup or verification. This method requires LUN clones to be created within the same flexible volume.

To boil all the above down to a simple recommendation is difficult. In most cases, it would be better to create a FlexClone volume in order to duplicate a LUN. The case where it may not make sense is when a volume contains data unrelated to the LUN and additional considerations (e.g., security) prevent sharing that information with a system that needs access to the LUNs. Another case is when the filer already contains close to the maximum number of volumes. Since a FlexClone volume counts as a volume, it may not be possible to create additional FlexClone volumes at a time when additional LUN clones can be created.

To learn more about LUNs, please see the excellent [Block Management with Data ONTAP 7G](#) white paper, the [Data ONTAP 7G Storage Management Guide](#), and the [Command Reference](#).

Summary

Starting with Data ONTAP 7G, storage administrators have access to greater flexibility and performance. Flexible volumes, aggregates, and RAID-DP™ provide unparalleled levels of storage virtualization, enabling IT staff to economically manage and protect enterprise data without compromise. FlexClone volumes are one of the many powerful features that make this possible, providing instantaneous writable volume copies that use only as much storage as necessary to hold new data.

FlexClone volumes enable and simplify many operations. Application testing benefits from less risk, less stress, and higher service levels by using FlexClone volumes to try out changes on clone volumes and upgrade under tight maintenance windows by simply swapping tested FlexClone volumes for the originals. Data mining and parallel processing benefit by using multiple writable FlexClone volumes from a single data set, all without using more physical storage than needed to hold the updates. FlexClone volumes can be used as online backup and disaster recovery volumes—immediately resuming read-write operation if a problem occurs. System deployment becomes much easier by cloning template volumes for testing and rollout. IT operations benefit from multiple copies of a

production system that can be used for testing and development and refreshed as needed to more closely mirror the live data.

This paper thoroughly explored the flexible volume clone feature of Data ONTAP 7G. It explained how FlexClone volumes work, explored practical applications, walked through an extended example, discussed performance, and detailed a number of related best practices. Our hope is that storage system administrators now have a solid understanding of the full value FlexClone volumes and Data ONTAP 7G can bring their operations. Please take a moment to imagine how they can help you!

References

NetApp Technical Reports and White Papers

A Storage Networking Appliance	www.netapp.com/tech_library/3001.html
File System Design for an NFS File Server Appliance	www.netapp.com/tech_library/3002.html
Synchronous SnapMirror Design and Implementation	www.netapp.com/tech_library/3326.html
Enabling Rapid Recovery with SnapVault	www.netapp.com/tech_library/3252.html
iSCSI: Accelerating the Transition to Network Storage	www.netapp.com/tech_library/3241.html
Writable LUNs and LUN Cloning for Oracle	www.netapp.com/tech_library/3266.html

Data ONTAP 7G and Flexible Volumes Information

Data ONTAP 7G Overview	
Introduction	
Configuration and Capacity Planning	
Data Protection	
LUN Management	
Performance Considerations	
Security Considerations	
Data ONTAP 7G SnapMirror Design and Implementation Guide	
Block Management with Data ONTAP 7G	
Data ONTAP 7G and OLTP Performance	

Data ONTAP 7G Documentation

Storage Management Guide	
Command Reference	
System Administration Guide	
Online Backup and Recovery Guide	

Web Resources

Network Appliance, Inc.	www.netapp.com
Network Appliance on the Web (NOW Portal)	http://now.netapp.com

[FlexClone FAQ](#)[Knowledge Base](#)

Legal Information

Copyright

Copyright © 1994–2004 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

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

Software derived from copyrighted material of Network Appliance, Inc. is subject to the following license and disclaimer: Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

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

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

Contact information

Network Appliance, Inc.

495 East Java Drive

Sunnyvale, CA 94089

Telephone: (408) 822-6000

Fax: (408) 822-4501

Support telephone: (888) 4-NETAPP

Documentation comment: doccomments@netapp.com