Redbooks Paper

Alex Osuna

An Introduction to FlexClone Volumes

Introduction / Overview

The goal of this Redpaper is to help storage system administrators understand the full value FlexClone volumes 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.

This Redpaper describes a powerful new feature that allows N series administrators to instantly create clones of a flexible 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.

© Copyright IBM Corp. 2006. All rights reserved.

FlexClone can be thought of as a transparent writable layer in front of Snapshot, since a FlexClone is initialized with a Snashot copy and updated directly and continuously thereafter (see Figure 1).

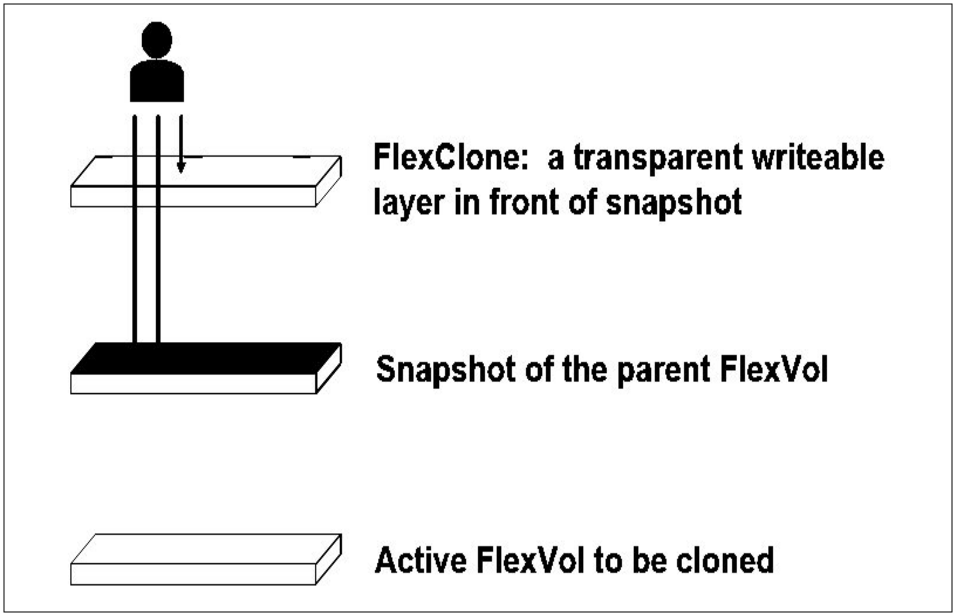
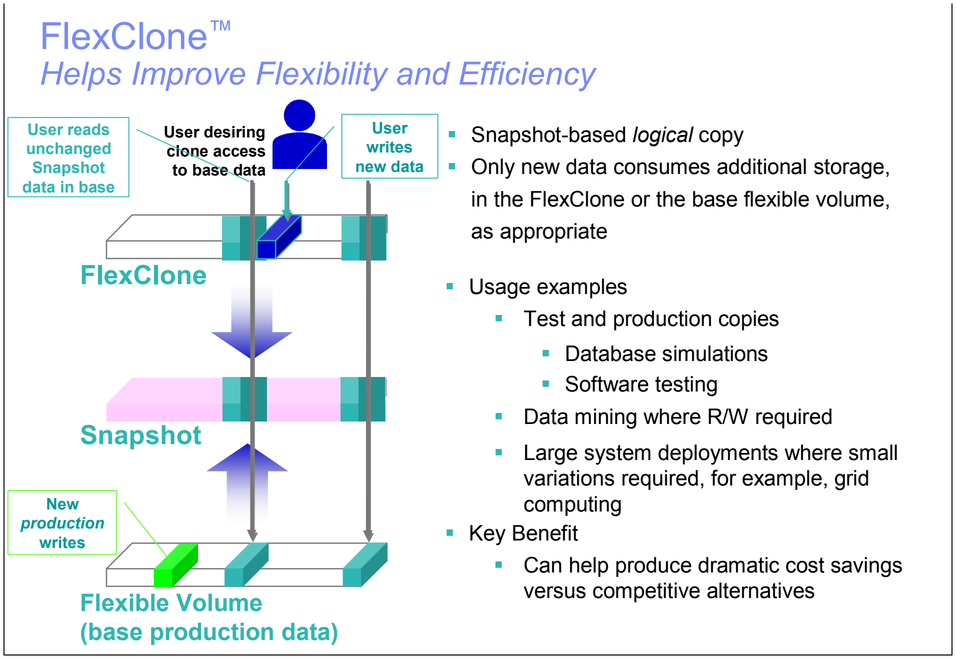

FlexClone: a transparent writeable layer in front of snapshot
Snapshot of the parent FlexVol
Active FlexVol to be cloned

Figure 1 Think of a FlexClone volume as a transparent writable layer in front of a Snapshot

Conceptually, FlexClone volumes are great for any situation where testing or development occurs (see Figure 2 on page 3), 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.



FlexClone™

Helps Improve Flexibility and Efficiency

User reads unchanged Snapshot data in base
User desiring clone access to base data
User writes new data
FlexClone
Snapshot
New *production* writes
Flexible Volume (base production data)

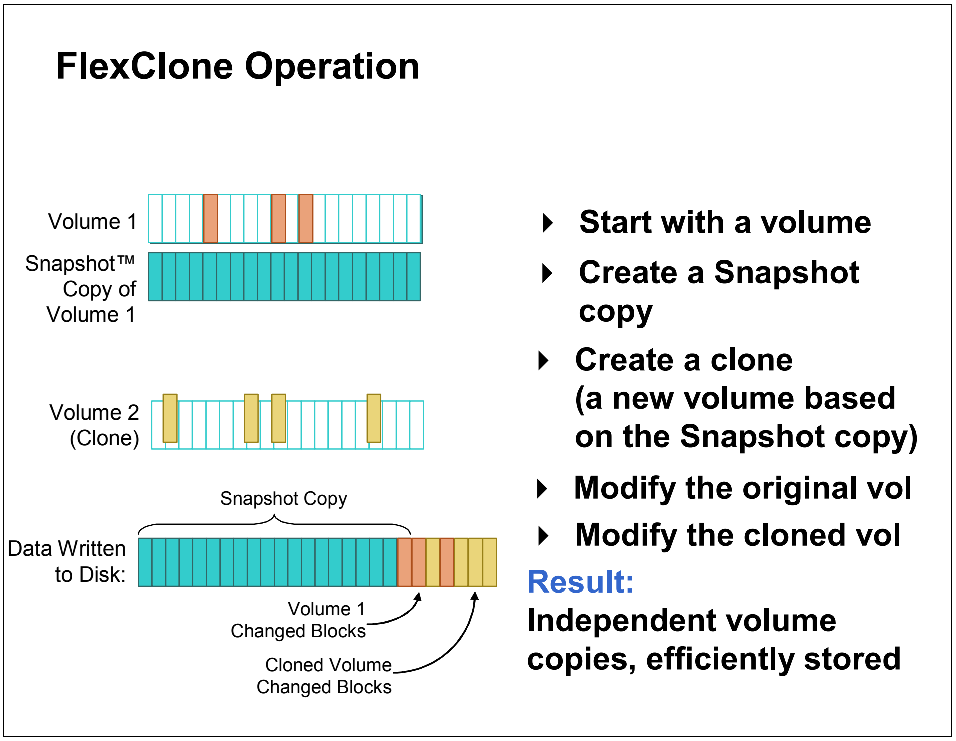
- Snapshot-based *logical* copy
- Only new data consumes additional storage, in the FlexClone or the base flexible volume, as appropriate
- Usage examples
  - Test and production copies
    - Database simulations
    - Software testing
  - Data mining where R/W required
  - Large system deployments where small variations required, for example, grid computing
- Key Benefit
  - Can help produce dramatic cost savings versus competitive alternatives

Figure 2 FlexClone basics

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.

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. N series filers use a Write Anywhere File Layout (WAFL) to manage disk storage. Any new data that gets written to the volume does not 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 (see Figure 3 on page 5). 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 4 on page 6 provides a graphical illustration of how this works.



FlexClone Operation

- Start with a volume
- Create a Snapshot copy
- Create a clone (a new volume based on the Snapshot copy)
- Modify the original vol
- Modify the cloned vol

Result:
Independent volume copies, efficiently stored

Figure 3 FlexClone operation

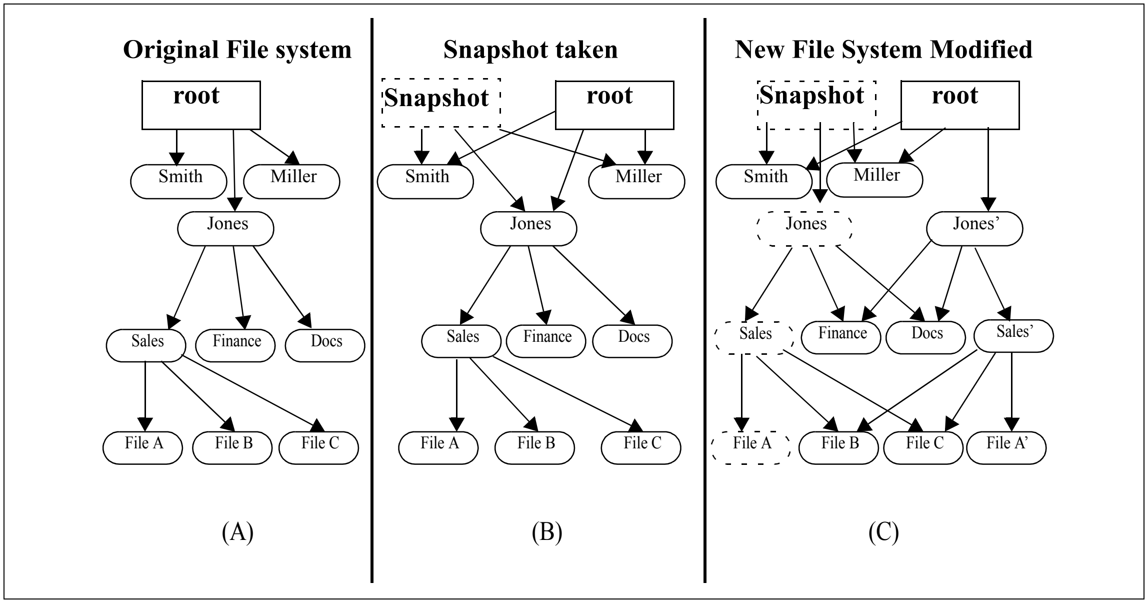


Figure 4 Snapshot copy

WAFL creates a Snapshot copy by duplicating the root. WAFL avoids changing blocks in a Snapshot copy by writing new data to new locations on disk.

You can think of a FlexClone volume as a transparent writable layer in front of the Snapshot copy. Figure 1 on page 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. While 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 or additions to either the parent FlexVol volume or the FlexClone volume.

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, you can manage volumes, Snapshot copies, and FlexClone volumes, including getting their status and seeing the relationships between the parent, Snapshot copy, and clone. 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 (for example, /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 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 1 on page 8 lists a few of the more common examples.

Table 1 Application area

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, and so on. 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 2 on page 9 provides an overview of the commands associated with FlexClone volumes as a reference to help along the way.

Table 2   Function description

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`
	Creates a new read-write clone named Widget, based on an existing flexible volume named Gadget. Reuses an existing Snapshot copy in Gadget as the backing for the clone.	
Status	`vol status [- v] cl_vol_name`	`vol status -v Widget`
	Prints 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.	
Splitting	`vol clone split start cl_vol_name`	`vol clone split start Widget`
	Starts 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.	
	`vol clone split status cl_vol_name`	`vol clone split status Widget`
	Checks the progress of the ongoing clone split for the Widget volume.	
	`vol clone split stop cl_vol_name`	`vol clone split stop Widget`
	Stops 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.	
Space Utilization	`df -A aggr_name`	`df -A aggr1`
	Displays disk usage associated with aggregates. In the example above, limit it to the aggregate named aggr1.	
	`vol container vol_name`	`vol container Widget`
	Displays the name of the aggregate containing the volume named Widget.	

Background for Scenario Inc.

For our example, we will use a fictional company called Scenario Inc. (SI). 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 N Series Storage Systems running the latest ONTAP software, including the FlexClone feature.

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”) (see Figure 5 on page 11, Figure 6 on page 11, and Figure 7 on page 11), 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_ucode=on, convert_ucode=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
```

Figure 5 vol status command

```
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
```

Figure 6 df command

```
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%
```

Figure 7 df command

Alice then creates FlexClone volumes of the two template flexible volumes, giving them names meaningful to her new project (see Figure 8 and Figure 9). 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.
```

Figure 8 Vol clone create

```
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.
```

Figure 9 Vol clone create

During the running of the **`vol clone`** command, Data ONTAP prints an informational message that states `Reverting volume GadgetData to a previous snapshot`. This is the standard message when a Snapshot copy is used to restore a volume to a previous state. Because 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, which appear associated with both the parent and the clone volumes (see Figure 10 on page 13). 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.

```
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
```

Figure 10 snap list

Data ONTAP will print a message about space reservations for FlexClone volumes. Data ONTAP 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.

Alice uses the **vol options** command to check the options for a clone volume (see Figure 11). 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 does not run out of space by watching space usage and setting alerts based on available space in the aggregate (see Figure 12). 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 (see Figure 13).

```
toaster> vol options GadgetTools
nosnap=off, nosnapdir=off, minra=off, no_atime_update=off, nvfail=off,
snapmirrored=off, create_ucode=on, convert_ucode=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
```

Figure 11   vol options

```
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
```

Figure 12   df -g

```
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%
```

Figure 13   df -Ag

Alice creates about 2 GB of new data in the cloned volume. That data shows up as a larger size for GadgetTools in the **df** command output (see Figure 14 on page 15).

```
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
```

Figure 14 df -g command

Alice then deletes some of the original files that were in the volume from the parent because she does not want them to confuse them with her current work.

Notice that the amount of space used in GadgetTools went down from 9 GB to 5 GB (see Figure 15), but the amount accounted for by the associated Snapshot copy space went up from 0 GB to 4 GB. That is because Data ONTAP accounts for the 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 the parent and accounted for in the Snapshot copy space.

```
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
```

Figure 15 df -g

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. There is no danger of running out of space as long as there is available space in the aggregate. We recommend that space associated with Snapshot copies be kept below 100%. That is no problem, because Data ONTAP allows resizing a FlexClone volume, while it is active, just like a regular FlexVol volume. All that needs to be done is to add some space from the containing aggregate (see Figure 16) and adjust the snap reserve to allocate more of the space to the Snapshot copy (see Figure 17).

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

Figure 16 vol size command

```
toaster> snap reserve -V GadgetTools 50
```

Figure 17 snap reserve

Notice that the amount of space associated with GadgetTools is now 40 GB: 20 GB for the volume and another 20 GB for the Snapshot copies (see Figure 18). The amount of space used in the aggregate did not increase by 20 GB (see Figure 19). That is because space reservations are disabled for FlexClone volumes in Data ONTAP.

```
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
```

Figure 18 df -g command

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

Figure 19 df -Ag command

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 “LUNs and FlexClone volumes” on page 33.

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 later realizes that it is not particularly helpful and might just create more confusion. She destroys the clone volume with the following sequence of commands (see Figure 20, Figure 21, and Figure 22).

```
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.
```

Figure 20   First command

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

Figure 21   Second command

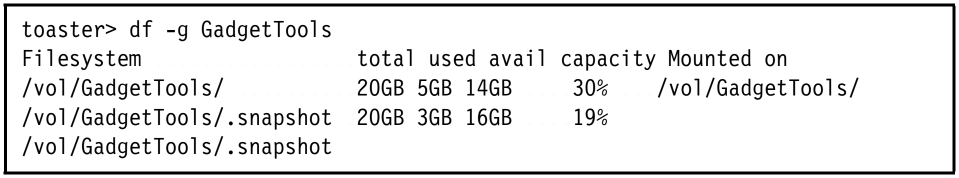
```
toaster> snap delete WidgetTools clone_Whizzer.1
```

Figure 22   Last command

Splitting off a FlexClone volume to create an independent volume

The prototype 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 does not matter if it is the clone or independent metadata that gets updated. You can accomplish this task by doing the following actions.

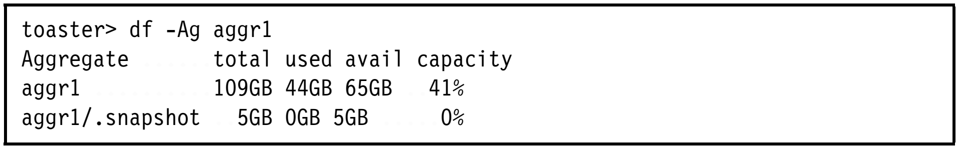
First, make sure there is enough space available in the aggregate to support the new independent volume (see Figure 23).



```
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
```

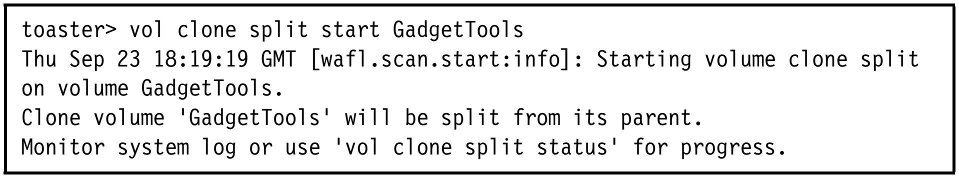
Figure 23 df -g command

The GadgetTools volume and its associated Snapshot copy area take up 40 GB. There is still 65 GB available in the containing aggregate (see Figure 24). Initiate the split and check the progress of the split (see Figure 25).



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

Figure 24 df -Ag



```
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.
```

Figure 25 vol clone split

After the split completes successfully (see Figure 26 on page 19), the space reservations are again enabled, and the space is guaranteed. Notice that the new volume has used approximately 40 GB from the aggregate, and that the Snapshot copy area is currently empty (see Figure 27 on page 19 and Figure 28 on page 19). That is 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 output does not mention that the volume is a clone or has the space guaranteed function disabled (see Figure 29). In fact, the **vol options** command output shows that space is guaranteed for the volume (see Figure 30).

```
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.
```

Figure 26 vol clone split status

```
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
```

Figure 27 df command

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

Figure 28 df -Ag command

```
toaster> vol status GadgetTools
Volume      State  Status   Options
GadgetTools online raid_dp, flex create_ucode=on,
                            convert_ucode=on
      Containing aggregate: 'aggr1'
```

Figure 29 vol status

```
toaster> vol options GadgetTools
nosnap=off, nosnapdir=off, minra=off, no_atime_update=off, nvfail=off,
snapmirrored=off, create_ucode=on, convert_ucode=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
```

Figure 30 vol options command

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.

You could use the **`vol clone split stop`** command to stop the split process. If you 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 is not exactly like starting where the previous split left off, but it is close.

Now that GadgetTools is an independent volume, the backing Snapshot copy in the parent FlexVol volume is no longer needed. Plus, now that most of the development work has stabilized, Alice does not 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 is not needed, because the original data can be reused without modification. Figure 31, Figure 32, Figure 33 on page 21, Figure 34 on page 21, Figure 35 on page 21, Figure 36 on page 21, Figure 37 on page 21, Figure 38 on page 21, and Figure 39 on page 22 demonstrate the cleanup process.

```
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
```

Figure 31 Step 1: snap list

```
toaster> snap delete WidgetTools clone_GadgetTools.1
```

Figure 32 Step 2: snap delete

```
toaster> snap reserve GadgetTools 20
```

Figure 33 Step 3: snap reserve

```
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
```

Figure 34 Step 4: df command

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

Figure 35 Step 5: vol size

```
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
```

Figure 36 Step 6: df command

```
toaster> df -Ag aggr1
Aggregate       total used avail capacity
aggr1           109GB 61GB 47GB   57%
aggr1/.snapshot   5GB 0GB 5GB      7%
```

Figure 37 Step 7: df -Ag command

```
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.
```

Figure 38 Step 8: vol offline

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

Figure 39   Step 9: vol destroy

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

Replacing a volume with a modified clone

Rather than making changes directly in the stable GadgetTools volume, you can clone it and makes your changes in a clone volume named NewTools (see Figure 41). After the changes are validated, you can move NewTools into production in place of the GadgetTools (see Figure 42 and Figure 43 on page 23). Verify the space before the split (see Figure 40).

```
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
```

Figure 40   df command

```
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.
```

Figure 41   vol clone split

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

Figure 42   vol rename

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

Figure 43 vol rename

Up to this point, everything happens transparently on the filer. Now you need to make the new FlexVol volume available to the host in place of the old one (see Figure 45). The details will be different depending on the access method (CIFS, NFS, iSCSI, or FCP); in our example, we are using the **`cifs shares`** command (see Figure 44), but the process would be similar to the following steps. We recommend that you disable access to the volume until the new volume is accessible under the existing access path (see Figure 45, Figure 46, Figure 47 on page 24, and Figure 48 on page 24). The share or export operation completes in seconds, so the volume would not need to stay unavailable for long.

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

Figure 44 The cifs shares command

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

Figure 45 Deleting and adding using the cifs shares command

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

Figure 46 Validating using the cifs shares command

```
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.
```

Figure 47 vol offline

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

Figure 48 vol destroy

Cloning a FlexClone volume

Let us start by cloning the stable volume to create a volume named WorkInProgress (see Figure 49). Changes are made in the clone volume, but do not complete. You discover the failing code and make locks in your progress by making another clone of your working code (named WIP2). Unfortunately, you are now behind schedule, so you ask a coworker to help implement part of the code and you clone the working volume for them to use (named WIP3).

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

Figure 49 cloning process

You issue the **df** command to check the results (see Figure 50).

```
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/WIP2/.snapshot
/vol/WIP3/                    16GB 11GB 4GB         73% /vol/WIP3/
/vol/WIP3/.snapshot            4GB 0GB 3GB           1% /vol/WIP3/.snapshot
```

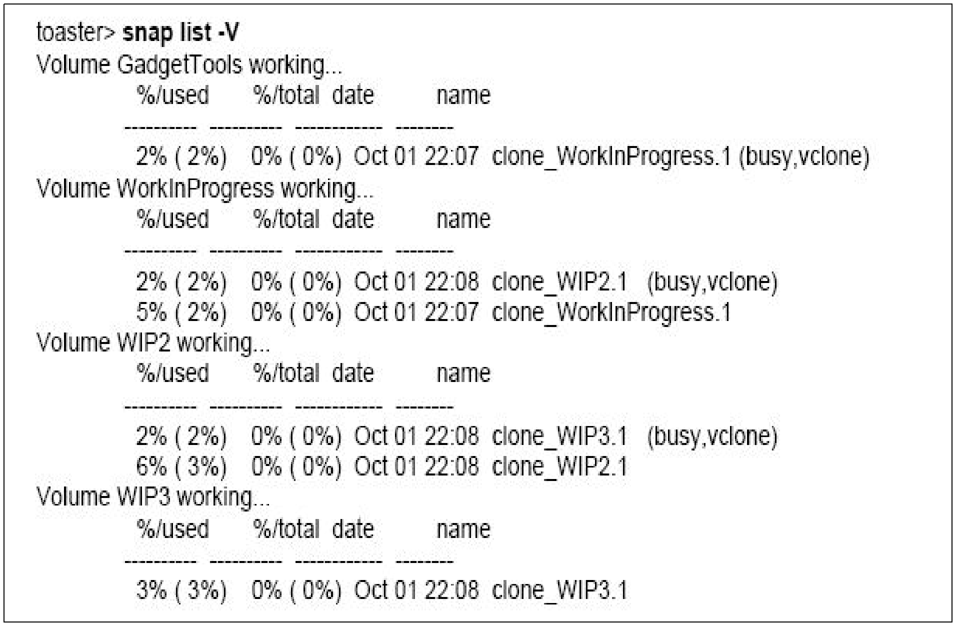
Figure 50 df command

You check the space availability of the aggregate with the **df -Ag** command (see Figure 51).

```
toaster> df -Ag aggr1
Aggregate        total used avail capacity
aggr1            109GB 62GB 47GB   57%
aggr1/.snapshot    5GB 0GB 5GB      4%
```

Figure 51   df -Ag

You issue the **snap list** command to see the new clone volumes and any snap copies (see Figure 52).



```
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
```

Figure 52   snap list

You are ready to test your new code. Your coworker e-mails his part to you, which he has incorporated into the code base. The code passes all the tests and you check the status (see Figure 53). You want to split off your clone (WIP2) and replace the current production volume.

.

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

Figure 53 vol status

You realize that you cannot split the clone because your coworker’s clone (WIP3) is still attached to their clone. This problem could have been discovered by looking at the line that states `Volume has locked snapshots, not splitting` in Figure 54. 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 would not have a backing Snapshot copy. Data ONTAP recognizes this and prevents accidents. You realize that you do not need WIP3 anymore and simply destroy WIP3 before retrying the split (see Figure 55).

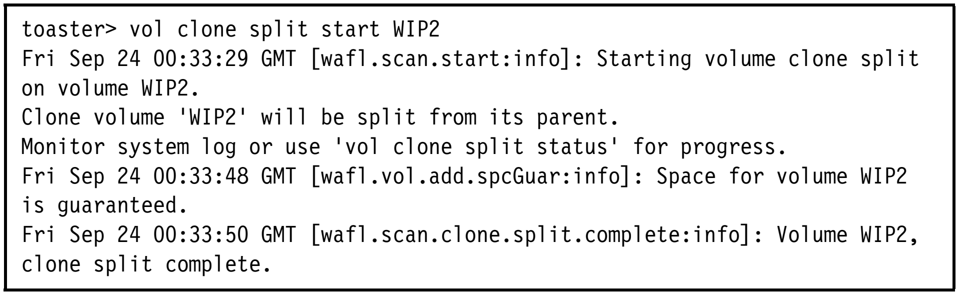
```
toaster> vol clone split start WIP2
Fri Sep 24 00:27:37 GMT [wafl.vv.clone.split:notice]: Volume 'WIP2' has
locked snapshots, not splitting.
vol clone split start: The volume has restricted snapshots
```

Figure 54 vol clone split start

```
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.
```

Figure 55 vol destroy

Follow the same process used in the previous section to replace the production volume with the newly split clone (see Figure 56).

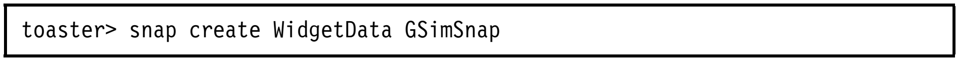


```
toaster> vol clone split start WIP2
Fri Sep 24 00:33:29 GMT [wafl.scan.start:info]: Starting volume clone split
on volume WIP2.
Clone volume 'WIP2' will be split from its parent.
Monitor system log or use 'vol clone split status' for progress.
Fri Sep 24 00:33:48 GMT [wafl.vol.add.spcGuar:info]: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

Figure 56 vol split

Multiple FlexClone volumes from a single parent FlexVol volume

Let us examine a few different scenarios. Each manipulates slightly different data sets as part of the simulation, so you make three FlexClone volumes of the original data (see Figure 58 on page 28), all using the same Snapshot copy (see Figure 57).



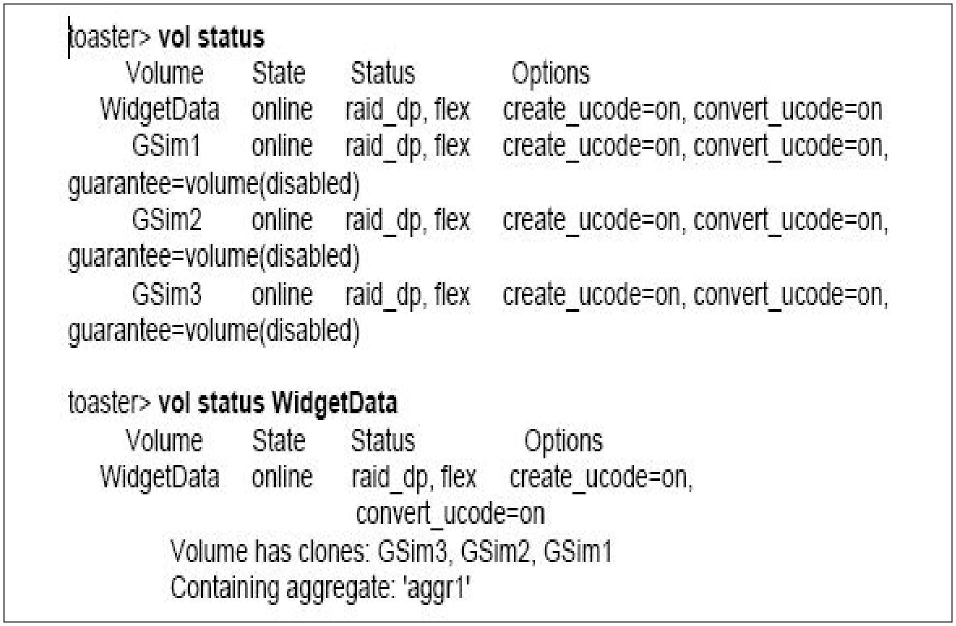
```
toaster> snap create WidgetData GSimSnap
```

Figure 57 snap create

```
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.
```

Figure 58 clone creation

Check the results of the creation with the **vol status** commands (see Figure 59 on page 29).



```
toaster> vol status
       Volume      State      Status           Options
    WidgetData    online    raid_dp, flex    create_ucode=on, convert_ucode=on
        GSim1      online    raid_dp, flex    create_ucode=on, convert_ucode=on,
guarantee=volume(disabled)
        GSim2      online    raid_dp, flex    create_ucode=on, convert_ucode=on,
guarantee=volume(disabled)
        GSim3      online    raid_dp, flex    create_ucode=on, convert_ucode=on,
guarantee=volume(disabled)

toaster> vol status WidgetData
       Volume      State      Status            Options
    WidgetData    online     raid_dp, flex    create_ucode=on,
                              convert_ucode=on
            Volume has clones: GSim3, GSim2, GSim1
            Containing aggregate: 'aggr1'
```

Figure 59 vol status

Modify each clone to be at the desired starting point and mount each clone on a different set of simulation engines, and let them work for a while. Your application is not particularly sensitive about permissions and security. If it was, you 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 made for the clones. Please see “Security, permissions, and SnapLock compliance volumes” on page 43 for a more in-depth discussion of these issues.

When complete, each simulation changed about 5 percent of the data. Altogether, the physical storage required was 115 percent of the original data set. Compare that with 400 percent and the large amount of copy time required if you had made three copies of the data instead. Archive a compressed copy of each final data set to another N series storage system and save the simulation results in case you need to use it at a later date. Then destroy the three FlexClone volumes and release the 15 percent extra production space used during the simulation.

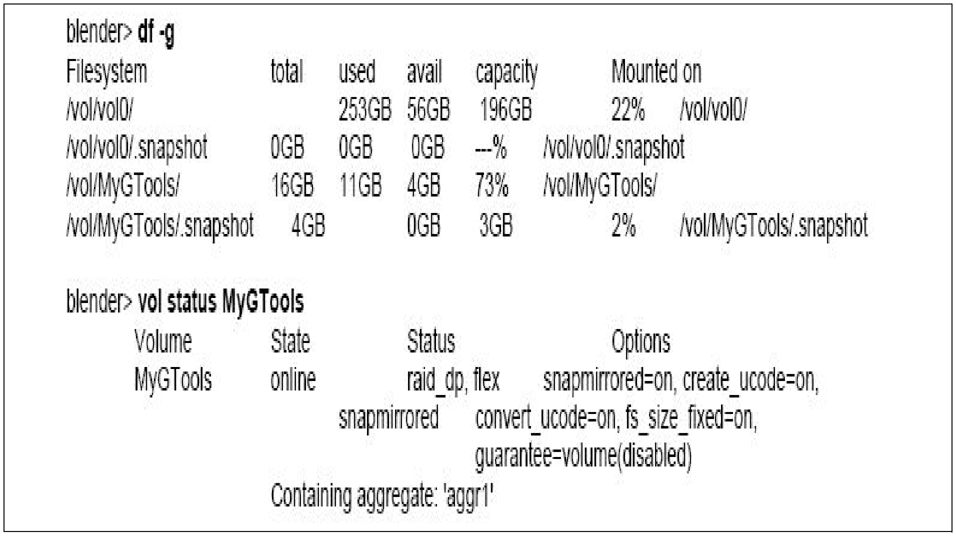
SnapMirror and FlexClone volumes

Let us say that you arrange for a SnapMirror copy of your work to be sent to a N Series storage system (named “blender”) in London. The N Series storage systems have worked together in the past, and 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 your coworker in London, whom we will call 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. You also arrange for a consistent Snapshot copy of your volume that he can use. Here is what Clive does.

Using the Data ONTAP commands above, Clive first creates a FlexVol volume on their N Series in London to receive the SnapMirror image (see Figure 60). Then he 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 the N series (see Figure 61 on page 31).

```
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.
```

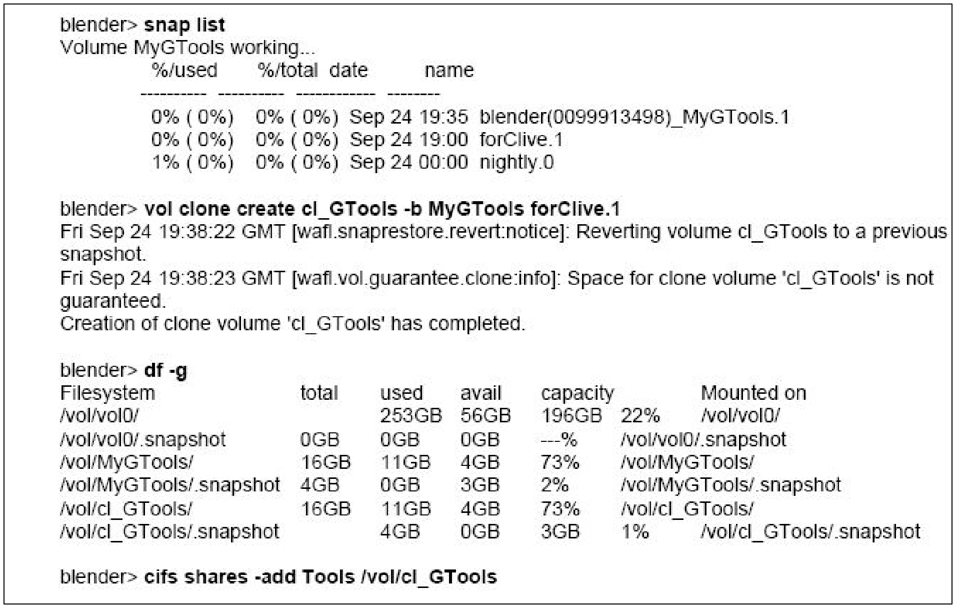
Figure 60 snapmirror to remote site



```
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_ucode=on,
                                    snapmirrored         convert_ucode=on, fs_size_fixed=on,
                                                         guarantee=volume(disabled)
                           Containing aggregate: 'aggr1'
```

Figure 61 df command

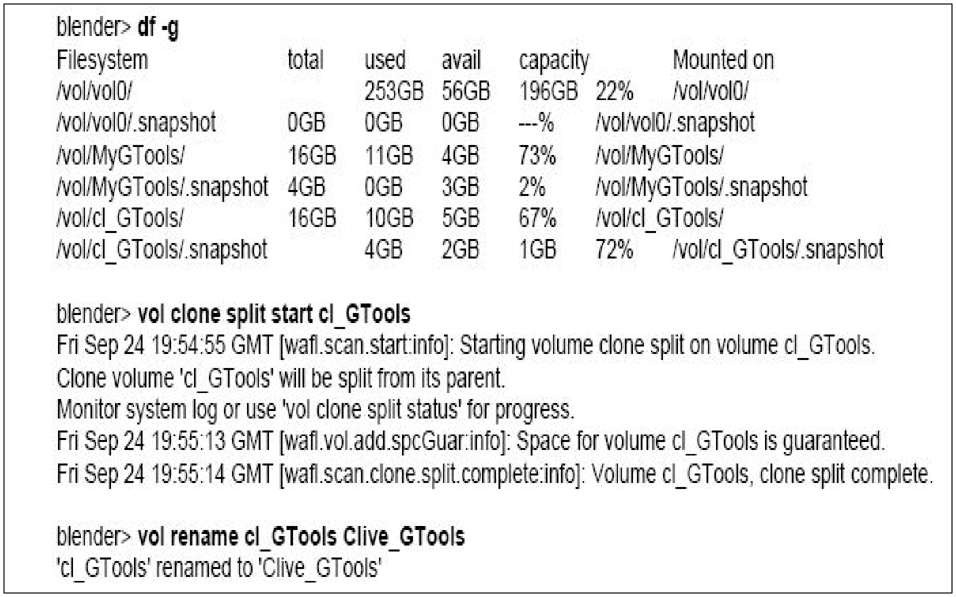
The next step is to create a writable copy of the read-only SnapMirror volume (see Figure 62 on page 31).



```
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
```

Figure 62 Receiving the Snapmirror

Clive uses the existing Snapshot copy you created for them (forClive.1) to back his new clone. You need to be careful not to delete the Snapshot copy your coworker is using while they have a clone dependent on that Snapshot copy. After checking that the new clone volume is available on his system (see Figure 63), Clive makes it accessible to his workstation and proceeds to make the necessary improvements. After he is done with this task, he likes the results so much that 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'
```

Figure 63 Making the volume available

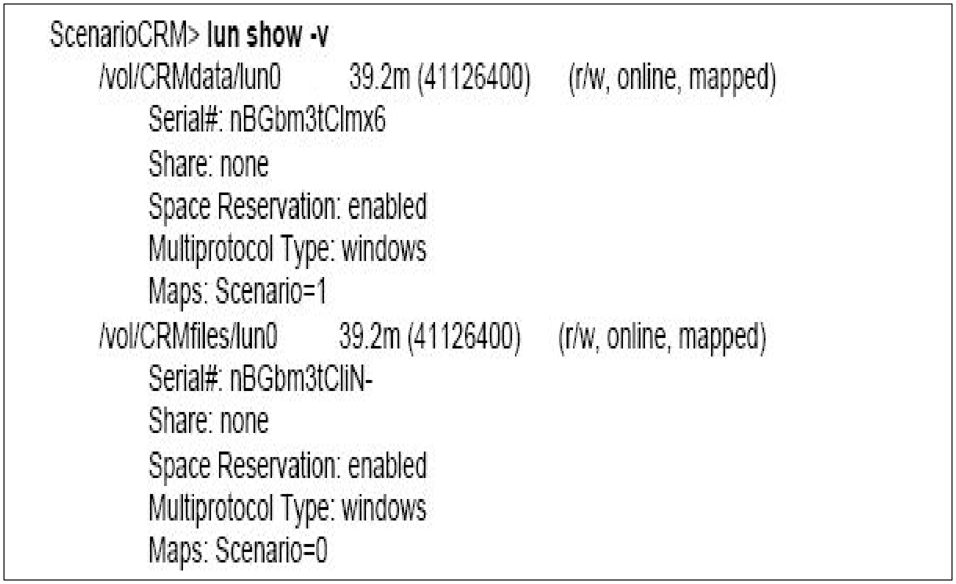
After integrating the changes, your coworker e-mails you the deltas, which turn out to be relatively small. If there were many changes, your coworker could have arranged for a SnapMirror copy of his clone to be sent back to your N Series storage system, where you could have broken off the mirror to replace your 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

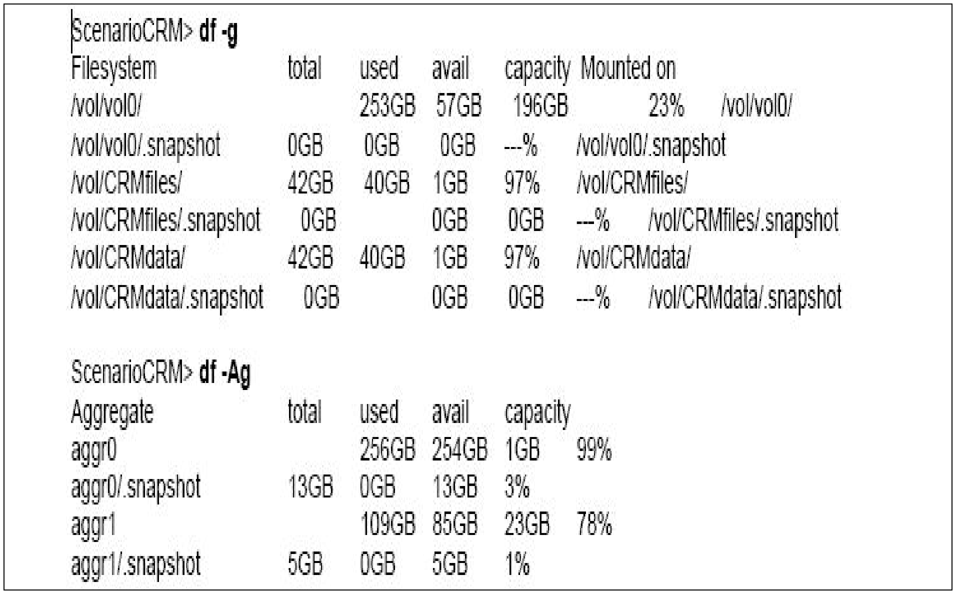
Meanwhile, Dave has been tasked with rolling out an upgrade for SI’s internal CRM system, which runs on top of an Oracle Database. The database is stored on a clustered N Series storage system and delivered to the server on top of iSCSI LUNs. SI’s CRM data is still pretty small, but needs to perform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

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 N Series storage system. The output (see Figure 64) indicates that there are currently two 39.2 GB 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 42 GB in size each. The flexible volumes have automatic Snapshot copies turned off and snap reserve set to 0 percent because the automatic Snapshot copies may not capture the LUN in a consistent state, because it must coordinate the Snapshot copy with the host file system to ensure the data is consistent. Using SnapDrive is one way to integrate the host file system with a N Series 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 (see Figure 65 on page 34) is getting full and has space available.



```
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
```

Figure 64 lun show



```
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
aggr0                           256GB   254GB   1GB     99%
aggr0/.snapshot         13GB    0GB     13GB    3%
aggr1                           109GB   85GB    23GB    78%
aggr1/.snapshot         5GB     0GB     5GB     1%
```

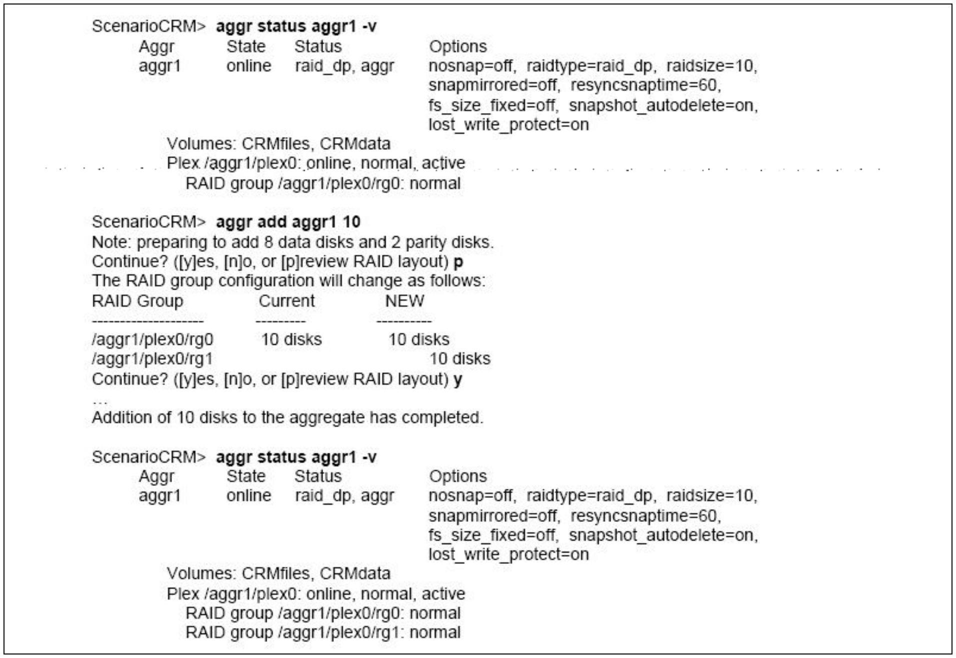
Figure 65 df -g

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 does not 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 (see Figure 66 on page 35) may not have space to hold all the changes generated during the upgrade and testing process. Dave has Data ONTAP on his N Series storage systems and can make all the necessary preparations without interrupting the production environment.

Here’s what Dave does:

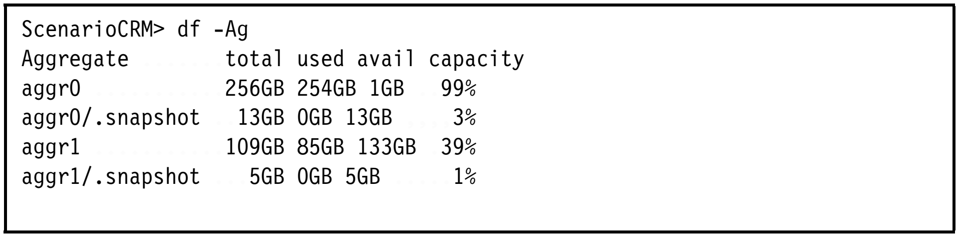
Dave double-checks the raid size defaults on aggr1 (see Figure 66 on page 35 and Figure 67 on page 35), and dynamically grows it by adding another RAID group of 10 disks from available spares. He then checks that the aggregate now contains a second RAID group (rg1) and there is plenty of space for Snapshot copies and FlexClone volumes. Because that is now taken care of, Dave dynamically grows the two flexible volumes containing the CRM LUNs (see Figure 68 on page 36). 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.



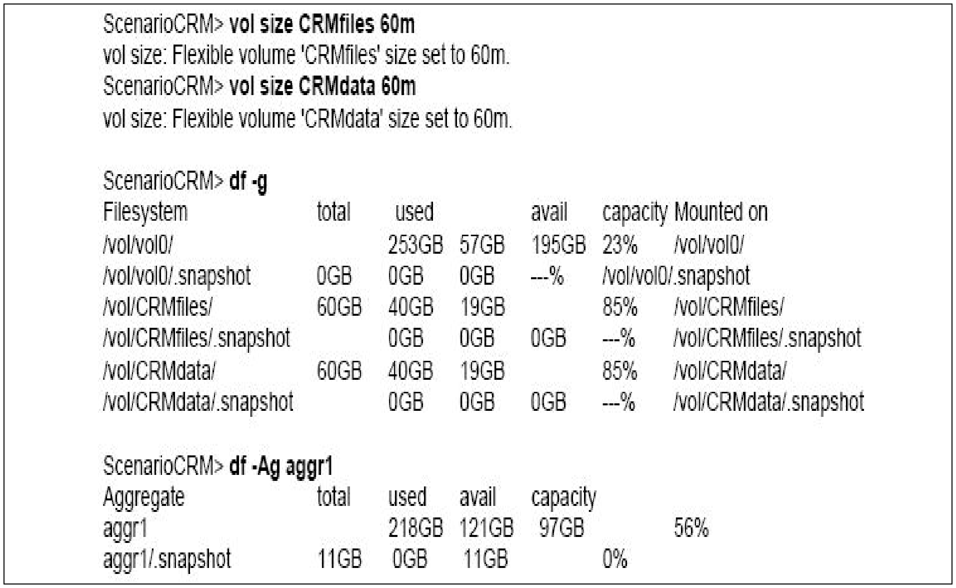
```
ScenarioCRM> aggr status aggr1 -v
        Aggr       State     Status            Options
        aggr1      online    raid_dp, aggr     nosnap=off,  raidtype=raid_dp,  raidsize=10,
                                               snapmirrored=off,  resyncsnaptime=60,
                                               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
```

Figure 66 aggr status



```
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%
```

Figure 67 df -Ag



```
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/CRMfiles/.snapshot
/vol/CRMdata/             60GB     40GB   19GB           85%      /vol/CRMdata/
/vol/CRMdata/.snapshot             0GB    0GB    0GB     ---%     /vol/CRMdata/.snapshot

ScenarioCRM> df -Ag aggr1
Aggregate                 total    used   avail  capacity
aggr1                              218GB  121GB   97GB            56%
aggr1/.snapshot           11GB     0GB    11GB           0%
```

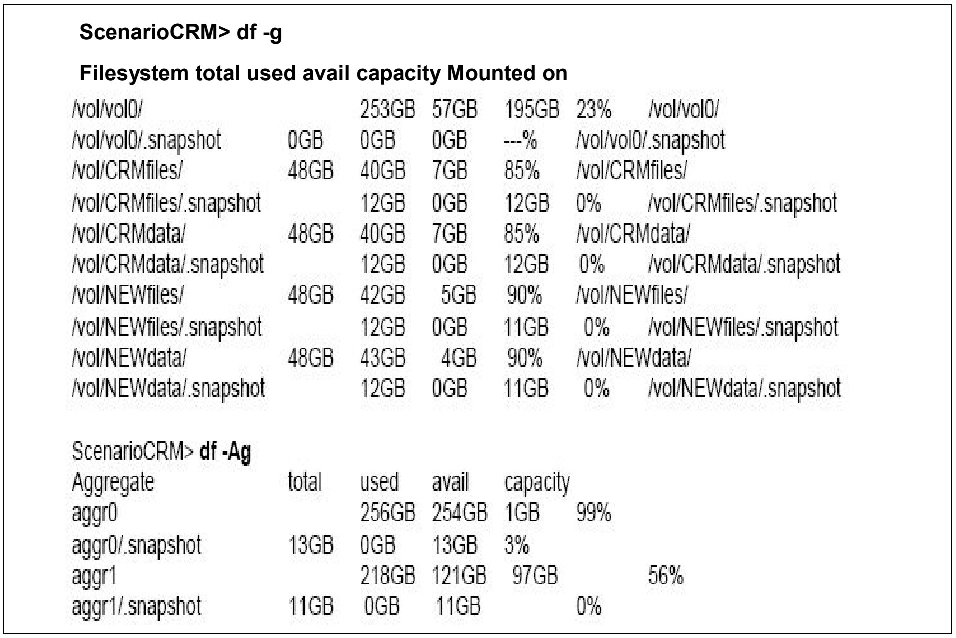
Figure 68   vol size

With his preparations completed, Dave is ready to make the actual FlexClone volumes (see Figure 69 on page 37). 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.
```

Figure 69 vol clone

Dave knows that the cloning processes duplicated all the LUN details along with the volumes (see Figure 70 on page 38). Notice that Data ONTAP is intelligent enough to take the cloned LUNs offline to prevent any mapping errors (see Figure 69). 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 bold below) from the production group (see Figure 71 on page 38 and Figure 72 on page 38). Mappings are additive, so old mappings need to be explicitly removed with the **lun unmap** command (see Figure 73 on page 39). When everything is squared away (see Figure 74 on page 39), Dave brings the cloned LUNs online.



```
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/              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
aggr0                               256GB  254GB  1GB    99%
aggr0/.snapshot             13GB    0GB    13GB   3%
aggr1                               218GB  121GB  97GB          56%
aggr1/.snapshot             11GB    0GB    11GB          0%
```

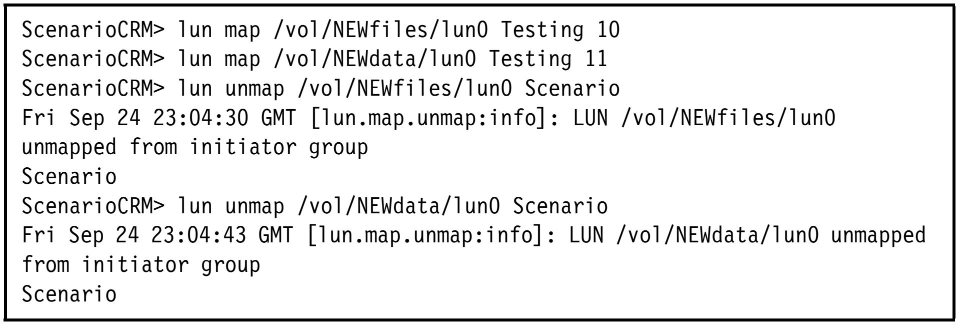
Figure 70   df commands

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

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

Figure 71   igroup command

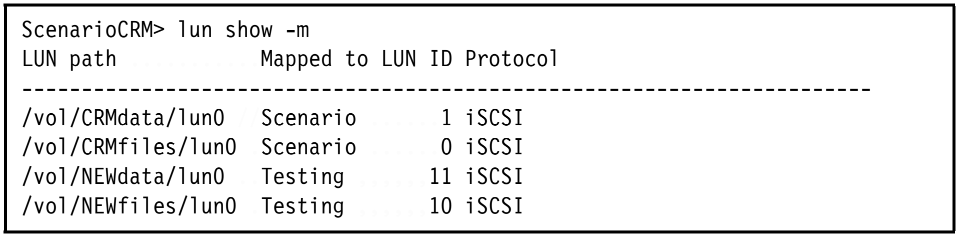
```
ScenarioCRM> lun show -m
LUN path            Mapped to LUN ID Protocol
-----------------------------------------------------------------------
/vol/CRMdata/lun0   Scenario       1 iSCSI
/vol/CRMfiles/lun0  Scenario       0 iSCSI
/vol/NEWdata/lun0   Scenario       1 iSCSI
/vol/NEWfiles/lun0  Scenario       0 iSCSI
```

Figure 72   lun show



```
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
```

Figure 73 lun map



```
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
```

Figure 74 lun show

Scenario summary

This wraps up our scenario 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

This section can be used as a way to understand the full scope of FlexClone benefits, as a refresher of specific FlexClone operations, or as a guided tutorial that N Series administrators can try with their own test storage system.

FlexClone performance

The performance of FlexClone volumes is nearly identical to the performance of flexible volumes. This is due 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 a large amount 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 uses a block reallocation functionality. The `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 (for example, adding disks to an aggregate) that may unbalance block allocations.

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 is 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 retrieve 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 a 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. 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 is 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:

- The FlexClone size minus the amount of space used in the FlexClone volume
- The 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 output will display the smaller value. That is one way for the administrator to track the usage. The over commitment 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 using many tools to notify the administrator when space is becoming in short supply. Data ONTAP 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 is 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, and so on. 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 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 storage system. 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 exported via NFS is identical 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 for which clone or task it is being used.

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 up 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 is 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. 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 is 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 DataFabric Manager application offers an efficient and powerful system for managing a storage infrastructure consisting of N Series storage systems. The FilerView tool, available on every N Series storage system, 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.

FlexClone volumes for LUN cloning

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 (for example, security) prevent sharing that information with a system that needs access to the LUNs. Another case is when the storage system 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.

FlexClone summary

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

This Redpaper thoroughly explored the flexible volume clone feature of Data ONTAP. It explained how FlexClone volumes work, explored practical applications, and discussed performance.

The team that wrote this Redpaper

This Redpaper was produced by a specialist working at the International Technical Support Organization, San Jose Center.

Alex Osuna is a Project Leader at the International Technical Support Organization, San Jose Center. He has over 28 years in the I/T industry and 19 years working exclusively in the storage area. He holds 10 certifications from IBM®, Microsoft® and Red Hat. Before joining the ITSO, he worked as a System Engineer for Tivoli.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

© Copyright International Business Machines Corporation 2006. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

This document created or updated on February 1, 2006.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 026, 5600 Cottle Road
  San Jose, California 95193 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DB2®	Redbooks™
IBM®	Redbooks (logo) ™

The following terms are trademarks of other companies:

Snapshot, SnapDrive, Data ONTAP, WAFL, SnapVault, SnapMirror, SnapManager, FilerView, DataFabric, and NetApp logo are trademarks or registered trademarks of NetApp Corporation or its subsidiaries in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.