

Creating a FLash ARchive (FLAR) image

Written by Greg King
Sunday, 30 September 2012 12:00

DISCLAIMER! This document is nothing more than the musings of the author as he attempts to perform the stated tasks. Conclusions and approaches may very well be incorrect, inefficient, or otherwise outside of professionally accepted best practices. Use this document at your own risk! In this document, screen outputs will be presented in **green**. Where keyboard input is required, the prompt will be in bolded red. **#** means you should be at the super user prompt, **\$** means you should be at an unprivileged user prompt. Do not include these prompts in your input! The command to be typed will be shown in **blue**.

ls -al

means you type ls -al at the super user prompt. The **FLash ARchive** (commonly referred to as a *flar image*) process is very similar to using Norton's Ghost on the Microsoft Windows platform except that you do not need to boot into a special environment in order to create a flar.

There are a few options related to creating differential flars, or creating a flar against a file system other than what is currently running, but those are outside of the scope of this document. We will only focus on the mandatory arguments, and refer you to flar man page (**\$ man flarcreate**) for more detailed information.

The mandatory command syntax for creating a flar image is as follows:
flar create -n testme /export/testme.flar

create : should be obvious, we wish to create a flar image

-n : names the archive, in this case it names it testme

the final argument is the output path and file name.

In the above example, we are creating a flar image located at /export/testme.flar with a flar name of testme

There are three other options we will discuss

-D exclude datasets

-S Skip the disk space check and do not write archive size data to the archive.

-c compress

It is a GREAT idea to include the architecture type in your flar name / filename. Obviously, a

Creating a FLash ARchive (FLAR) image

Written by Greg King

Sunday, 30 September 2012 12:00

flar created on a x86 system can not be used on a sparc based system. Likewise there are architectural differences in sparc systems themselves (sun4v , sun4u, etc) which are not compatible. You do not need to put the .flar extension on your image, but it makes sense to do so. Also, ensure your target directory exists before attempting to create the image. Notice that we are putting the date and some basic information in our name/filename. This will prove valuable when we use this image to jumpstart and / or network install a system.

Before creating a flar, delete unnecessary files such as in /var/tmp.

Since this is my first flar creation, I will create a zfs dataset to hold my flar images. You can put them anywhere you wish, I prefer this directory for now.

zfs create rpool/flar

and then set it for NFS sharing

```
# zfs set sharenfs='rw,root=. ' rpool/flar
```

```
# zfs set mountpoint=/flar rpool/flar
```

and then I will create my flar image.

You can see how long a command takes to run by preceding the command with ptime !

I create flar images as kind of a milestone and like to include the date, architecture and milestone in my names. I use the time stamp of YYYYMMDD so that they are sorted by age in an ls -al output.

```
# ptime flar create -n 20120928_sun4u_WS_baseline /flar/20120928_sun4u_WS_baseline.f  
lar
```

```
Full Flash  
Checking integrity...  
Integrity OK.  
Running precreation scripts...  
Precreation scripts done.  
Determining the size of the archive...  
The archive will be approximately 6.24GB.  
Creating the archive...  
Archive creation complete.  
Running postcreation scripts...  
Postcreation scripts done.
```

```
Running pre-exit scripts...  
Pre-exit scripts done.
```

```
real 20:14.419
```

Creating a FLash ARchive (FLAR) image

Written by Greg King

Sunday, 30 September 2012 12:00

user 8:55.484

sys 2:47.938 **USING -D**

With ZFS you can not exclude files or directories inside of a dataset from flar create. You can exclude a dataset from a flar image using the -D option as illustrated below. We will exclude the swap, dump and flar directories from our image.

```
# ptime flar create -n 20120928_sun4u_WS_baseline -D rpool/dump -D rpool/swap -D  
rpool/flar /flar/20120928_sun4u_WS_baseline_withd.flar
```

Full Flash

Checking integrity...

Integrity OK.

Running precreation scripts...

Precreation scripts done.

Determining the size of the archive...

The archive will be approximately 6.24GB.

Creating the archive...

Archive creation complete.

Running postcreation scripts...

Postcreation scripts done.

Running pre-exit scripts...

Pre-exit scripts done.

real 18:24.858

user 8:55.363

sys 2:49.246 **USING -c**

To compress the image, we will use -c (note that it is lower case). This option uses the 'compress' (see the compress man page for more information) function on the image.

```
# ptime flar create -c -n 20120928_sun4u_WS_baseline -D rpool/dump -D rpool/swap -D  
rpool/flar /flar/20120928_sun4u_WS_baseline_withdc.flar
```

Full Flash

Checking integrity...

Integrity OK.

Running precreation scripts...

Precreation scripts done.

Determining the size of the archive...

The archive will be approximately 4.24GB.

Creating the archive...

Archive creation complete.

Running postcreation scripts...

Postcreation scripts done.

Running pre-exit scripts...

Pre-exit scripts done.

real 46:37.169

user 38:08.140

Creating a FLash ARchive (FLAR) image

Written by Greg King

Sunday, 30 September 2012 12:00

sys 4:51.283 We saved 2GB on our image but it took twice as long to create.

USING -S

Our final test will skip the file size check and just build the image.

```
# ptime flar create -c -S -n 20120928_sun4u_WS_baseline -D rpool/dump -D rpool/swap -D  
rpool/flare /flare/20120928_sun4u_WS_baseline_withdcs.flare
```

```
Full Flash  
Checking integrity...  
Integrity OK.  
Running precreation scripts...  
Precreation scripts done.  
Creating the archive...  
Archive creation complete.  
Running postcreation scripts...  
Postcreation scripts done.  
Running pre-exit scripts...  
Pre-exit scripts done.
```

```
real 21:30.869
```

```
user 16:20.222
```

```
sys 2:40.168 Nice! Using -S cut our creation time by more than 50%
```

CONCLUSION

Shown below are the file sizes of the four flar images we create.

```
-rw-r--r-- 1 root root 6703188916 Sep 28 21:05 20120928_sun4u_WS_baseline.flare  
-rw-r--r-- 1 root root 6703139605 Sep 28 22:00 20120928_sun4u_WS_baseline_withd.flare  
-rw-r--r-- 1 root root 4554119568 Sep 30 09:59  
20120928_sun4u_WS_baseline_withdc.flare  
-rw-r--r-- 1 root root 4554205572 Sep 30 10:38  
20120928_sun4u_WS_baseline_withdcs.flare
```

If your policy allows, using the -S flag can considerably reduce the time taken to create a flar image and the -c flag can reduce the output file size. Given that you are probably issuing the command and returning later, it is a personal choice on using -S. The -c flag is a personal choice, especially if drive space or file transfer times (wan?) are not a consideration. The -D flag is very important especially on your flar repository. Notice in our file listing above that the flar sizes did not grow each time we created a new flar due to the exclusion of the rpool/flare dataset.

Now that we have mastered flar imaging, we will begin work on our jumpstart server in the next article.

<<end of document>>