penguinpunk.net
configuring emc mirrorview from scratch for use with vmware site recovery manager
revision 0.1

synopsis

The following is a simple document with screenshots providing step-by-step instructions for configuring the EMC CLARiiON with Navisphere Management Server for use with EMC MirrorView (/Synchronous or /Asynchronous) and VMware Site Recovery Manager. It also includes a few notes on operational activities. It is intended to be read in conjunction with the appropriate revision of the EMC Navisphere Manager Help and VMware Site Recovery Manager Administration Guide.

table of contents

table of figures

introduction

Configuring EMC MirrorView on the EMC CLARiiON with Navisphere / Unisphere for use with VMware Site Recovery Manager is a relatively simple process. The aim of this guide is to simply provide the screenshots that should accompany the words in the Administrators Guides. This document assumes that the CLARiiONs are licensed for MirrorView and that fibre-channel connectivity exists between the two arrays. The aim of the document is to cover the process from initial array and zoning configuration through to the establishment of suitable VMware Site Recovery Manager configurations. The screenshots used were taken on CX4-120 arrays running Release 28 FLARE, CX4-960 arrays running Release 30 FLARe and CX700 arrays running Release 26 FLARE. The MirrorView products used include both MirrorView/Synchronous and MirrorView/Asynchronous.

I strongly recommend you familiarise yourself with the following documents available from EMC:

- MirrorView Knowledgebook: FLARE 30 – A Detailed Review (h2417-mirrorview_knowledgebook-flare-wp.pdf);
- EMC Navisphere Command Line Interface (CLI) REFERENCE (300-003-628_a13.pdf);
- EMC MirrorView/Asynchronous Command Line Interface (CLI) Reference (300-001-335_a13.pdf); and
- EMC MirrorView/Synchronous Command Line Interface (CLI) Reference (069001184_a15.pdf).

prerequisites

As mentioned in the introduction, a number of prerequisites exist in order to perform a successful MirrorView implementation.

enablers / flare versions

Depending on the version of MirrorView that is required, you will need to load one or both MirrorView licences (/Synchronous and /Asynchronous) using the NST or USM. It should also be noted that the enablers should be installed on both arrays that are participating in the MirrorView activities. These enabler codes will have shipped from EMC on CD when the licenses were ordered.

zoning requirements / ports

There are a few minor points to notice with the required zoning for MirrorView. Generally speaking, the high port on the front-end is normally identified as the MirrorView port. When you create the zones for MirrorView, you need to zone SAN1-SPA-MV to SAN2-SPA-MV and SAN1-SPB-MV to SAN2-SPB-MV. If you zone SPA-MV to SPB-MV, MirrorView connections simply won't work. You should also note that the MirrorView port can be different if the array was upgraded. For example, a CX3-40f upgraded to a CX4-960 will have MirrorView ports on fe1, rather than fe3, because the MirrorView port on the CX3-40f is fe1.

Model	FC ports	iSCSI ports	FC MV port	iSCSI MV port
CX4-960	0-3	4-5	3	5
CX4-480	0-3	4-5	3	5
CX4-240	0-1	2-3	1	3
CX4-120	0-1	2-3	1	3

figure 1 – cx4 mirrorview ports[1]

[1] MirrorView Knowledgebook: FLARE 30 – A Detailed Review, p13

Model	FC MV port	iSCSI MV port
CX3-80, CX700, CX600 FC	3	--
CX3-40*, CX3-20*, CX500, CX400 FC	1	--
CX3-20, CX3-40 FC/iSCSI	5	3
CX3-10 FC/iSCSI	3	1
AX4-5 FC	1	--

*The MirrorView port (port 1) is the same across all CX3-20 and CX3-40 Fibre Channel-only models.

figure 2 - cx mirrorview ports[2]

To see what ports are assigned as the MirrorView ports on your array in Unisphere, go to System – System Information and click on Manage Data Ports.

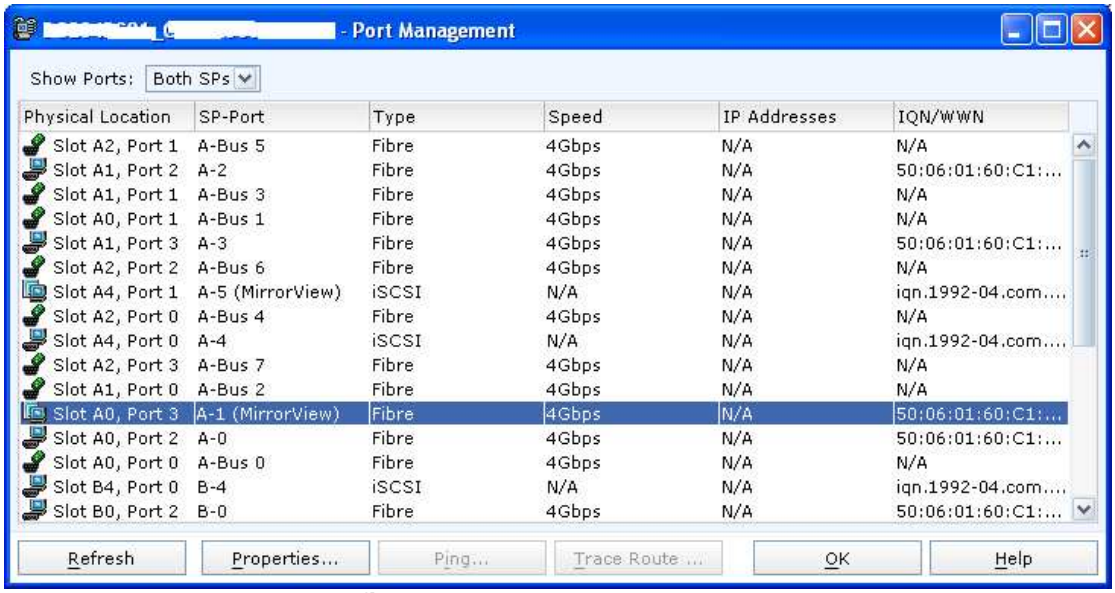

figure 3 - cx4 port management

If you want to query the ports using naviseccli, the following command will yield the results you seek.

C:\>naviseccli -h sanipaddress -user username -scope 0 ioportconfig -list |more

You'll see output that looks something like this.

[2] MirrorView Knowledgebook: FLARE 30 – A Detailed Review, p15

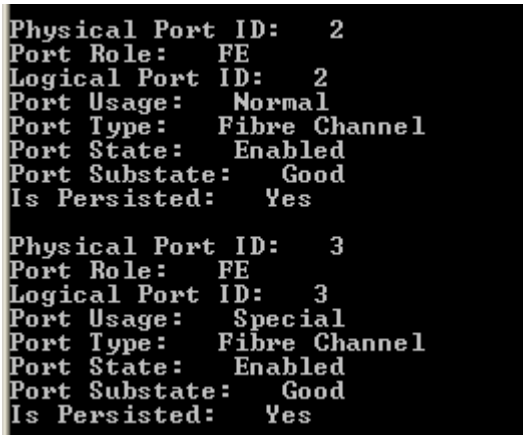

figure 4 - naviseccli ioportconfig -list

Notice that the Port Usage for the MirrorView port is marked as “Special”. Normal frontend ports are marked as “Normal”.

reserved lun pool configuration

If you are using MirrorView/Asynchronous or SnapView, you will need to configure a Reserved LUN Pool (RLP) on each array. If you are performing test failovers with SRM, you will be using SnapView, so you will need to configure the RLP. Broadly speaking, the RLP should be configured to account for 20 – 30% of the replicated storage, and shouldn’t reside on either Vault spindles or spindles that share production workloads. As there is no way you can dictate what RLP LUNs get used in replication operations, it is recommended that you configure the RLP on “good” performing disk. There is an excellent article on EMC’s Powerlink website that provides more information - emc107717 - Best practices for setting up Reserved LUN Pool (RLP).

enable mv/s (or /a)

Once the zoning has been completed, the MirrorView connection between the two arrays can be established.

In Navisphere, right-click on the first array, select MirrorView, and then select Manage Mirror Connections.

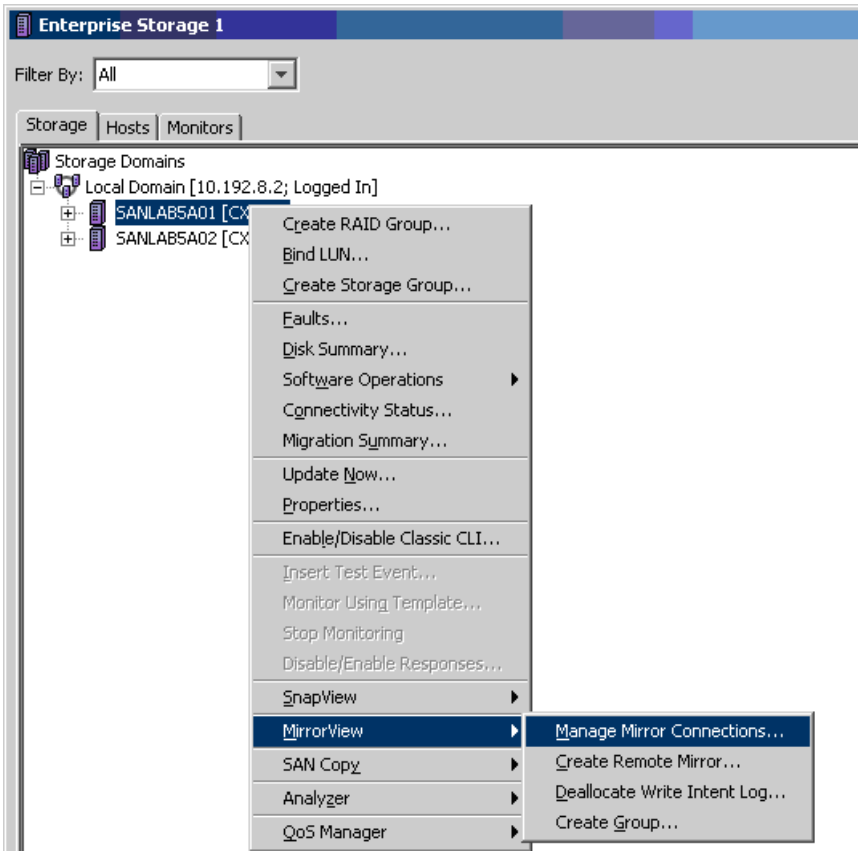

figure 5 - manage mirror connections

You will then be presented with a dialogue box that looks like this.

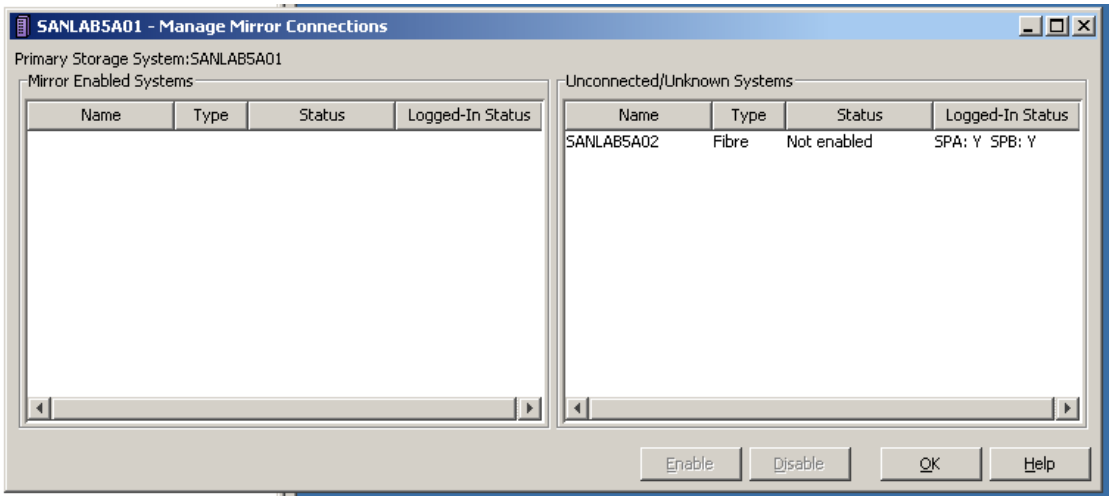

figure 6 - manage mirror connections

Select the system you'd like to connect and then click on Enable. The inevitable moment of self-doubt occurs and you will need to confirm that this is what you want to do. What's the worst that could happen? Click Yes!

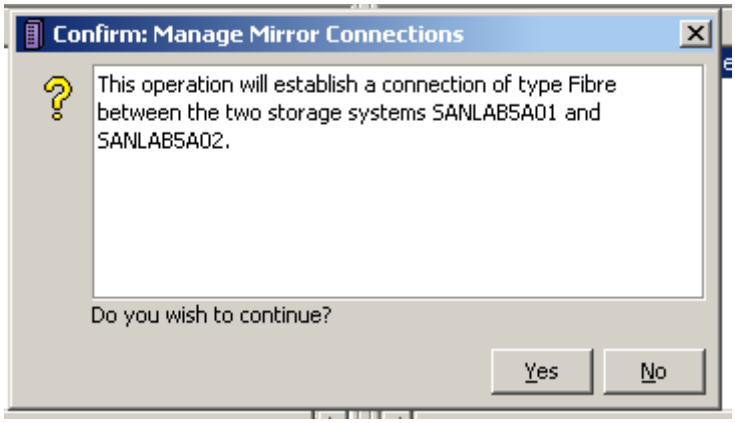

figure 7 - confirm manage mirror connections

Success follows!

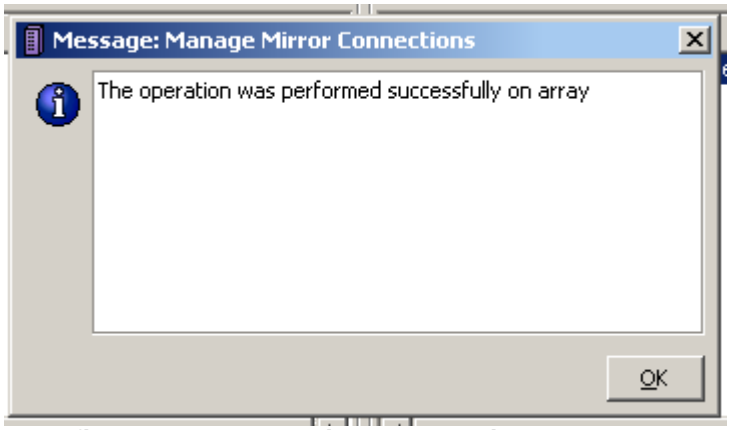

figure 8 - manage mirror connections success

And now the arrays can communicate via MirrorView.

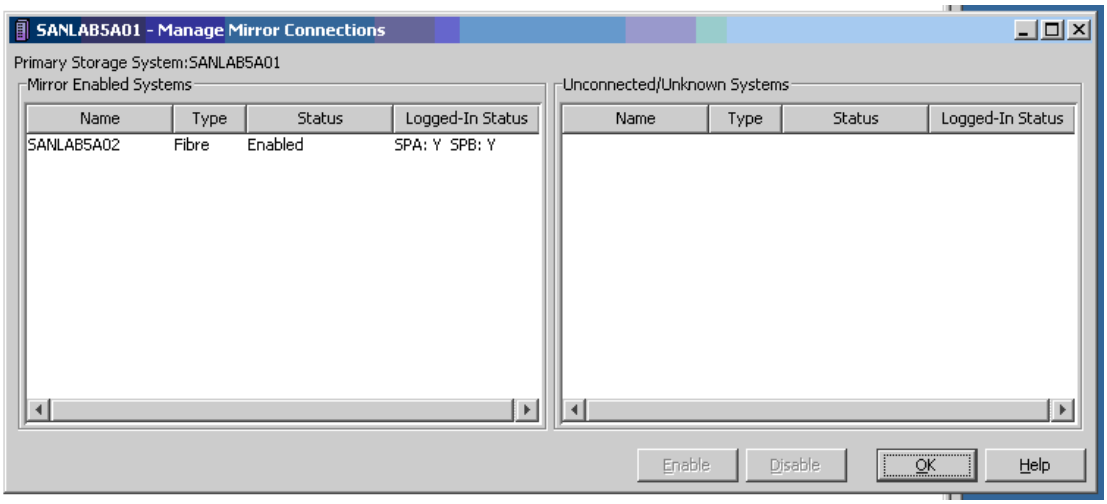

figure 9 - manage mirror connections

At this point I would suggest you attend to things such as creating and configuring Write Intent Log (WIL) LUNs if you are using /Synchronous. 2 * 128MB LUNs (one per SP) are all that is required.

initial mirror creation

The first step is to create a Secondary Image LUN for the LUN to be mirrored. This can either be done on the secondary array, or, by right-clicking on the primary LUN and selecting "Create Secondary Image LUN". A dialogue box is then presented similar to the figure below.

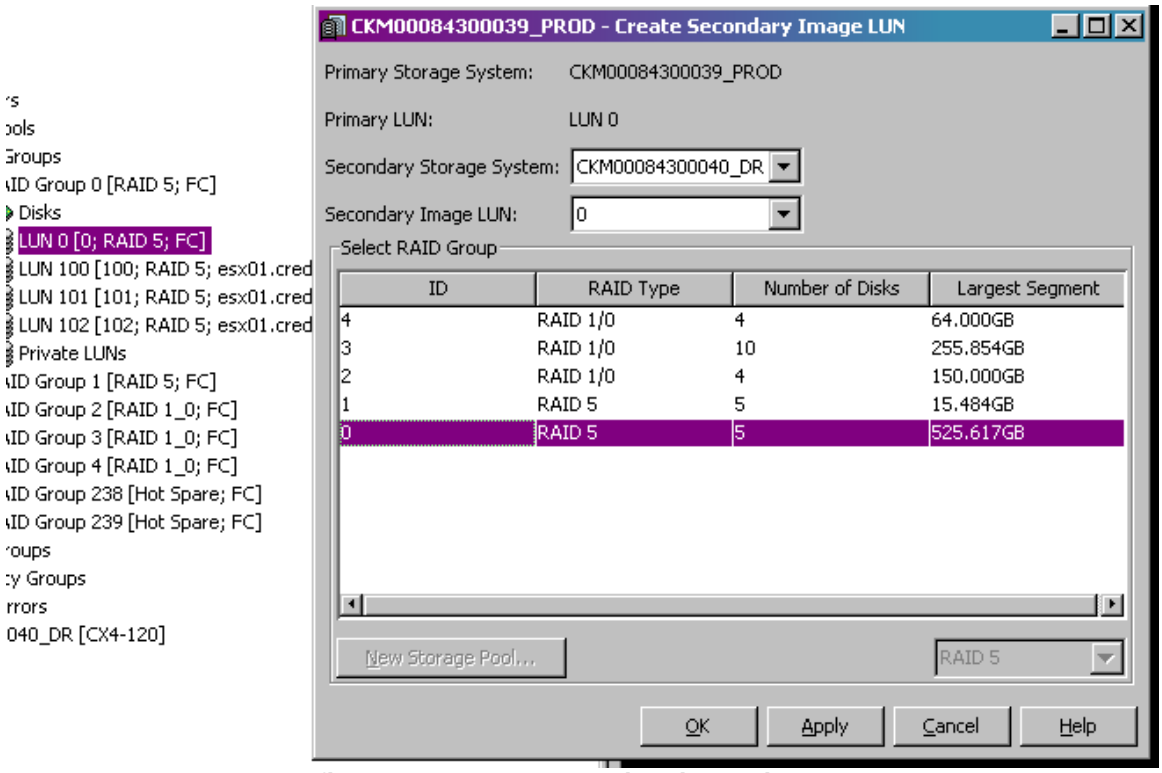

figure 10 - create secondary image lun

The secondary storage system can then be selected (useful if the CLARiiON has more than one MirrorView relationship), a Secondary Image LUN ID can be chosen (common sense would dictate that this ID be the same as the primary), and the RAID Group designated to contain the secondary image can also be selected. It is important to keep in mind that, while there are people around town using RAID Groups with dissimilar performance characteristics, this isn't recommended. If, for some reason, the workload is moved to the secondary site, the secondary images will obviously become primary images, and running ESX guests or Exchange databases on 1TB SATA-II spindles won't be anywhere near as fast as you'll want it to be.

Of course, like most things in Navisphere, you'll find yourself getting prompted time and again to confirm your actions. I don't think this is a bad thing.

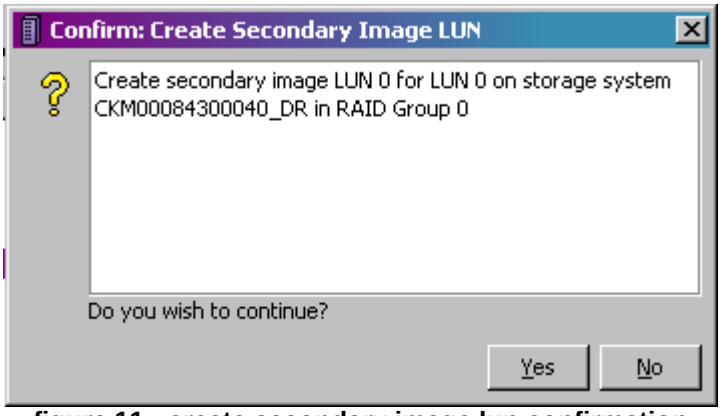

figure 11 - create secondary image lun confirmation

Once the secondary image LUN has been established on the secondary array, the Remote Mirror can then be created. To do this, start by right-clicking on the LUN to be mirrored. Select "MirrorView" – "Create Remote Mirror…".

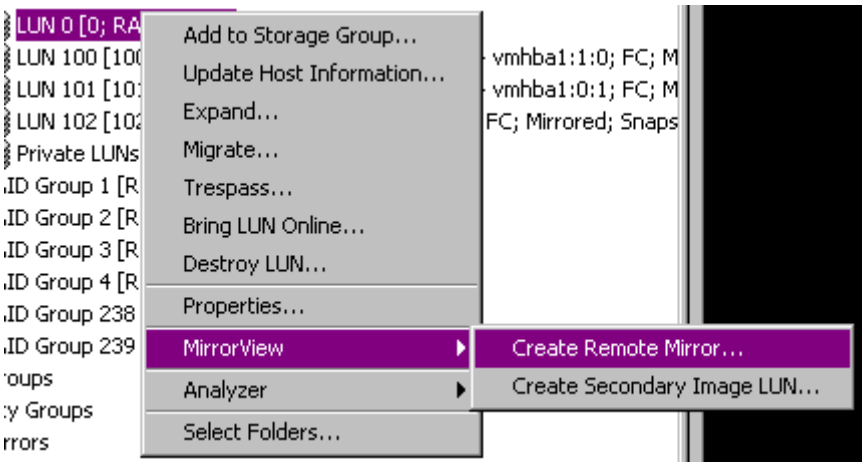

figure 12 – right-click, create remote mirror

The dialogue box will offer a number of choices, depending on the enablers that have been loaded on the array. In this example, the MirrorView relationship will be Asynchronous. The name provided should be something simple that reflects the function of the remote mirror. This should apply for all of your naming standards. This is why my children are called Nest Egg 1 and Nest Egg 2. But I digress.

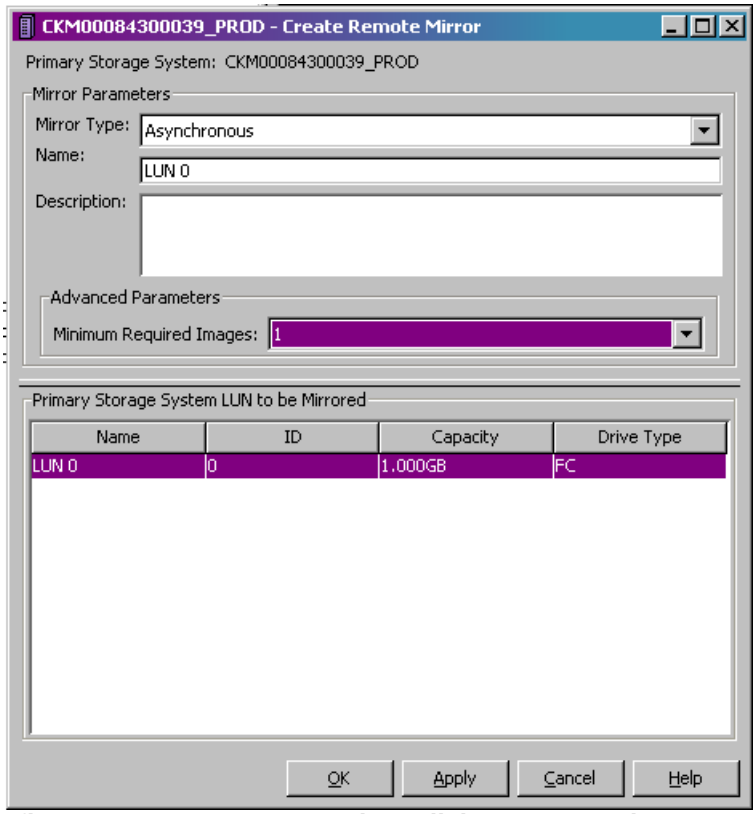

figure 13 - create remote mirror dialogue - asynchronous

Now let's do the same thing, but with a synchronous flavour.

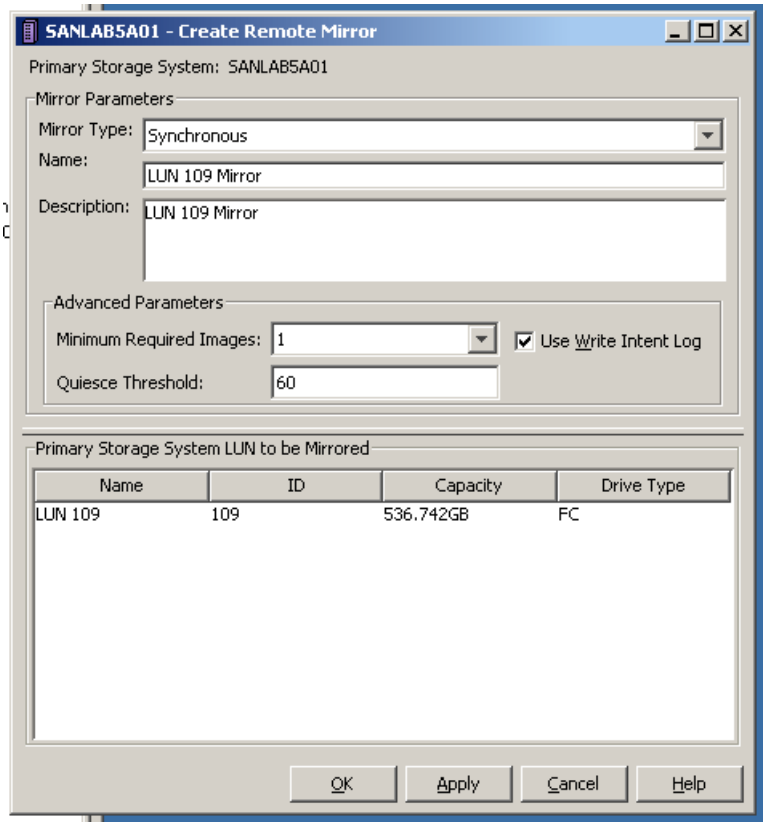

figure 14 - create remote mirror dialogue – synchronous

You'll then have the opportunity to confirm your actions. Note that the mirror isn't a mirror until

the secondary mirror image is added. We'll get to that shortly.

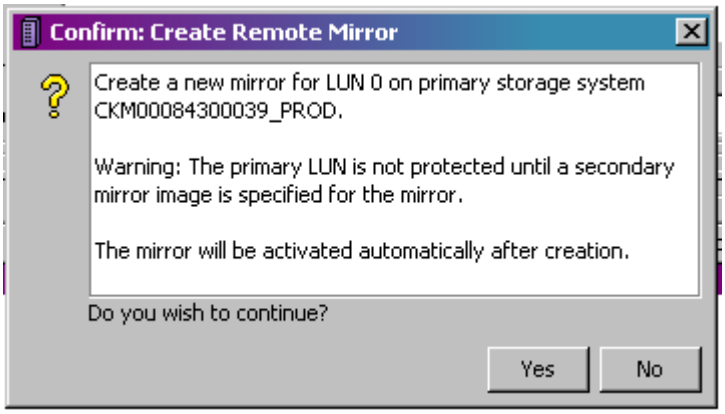

figure 15 - create remote mirror confirmation

Once the mirror has been created, the primary CLARiiON will come up with an F as in faulted. It is really important that you don't panic at this point because, unless you've also had the misfortune of some bizarre hardware fault occurring at the same time, it just means that the mirror needs, well, mirroring.

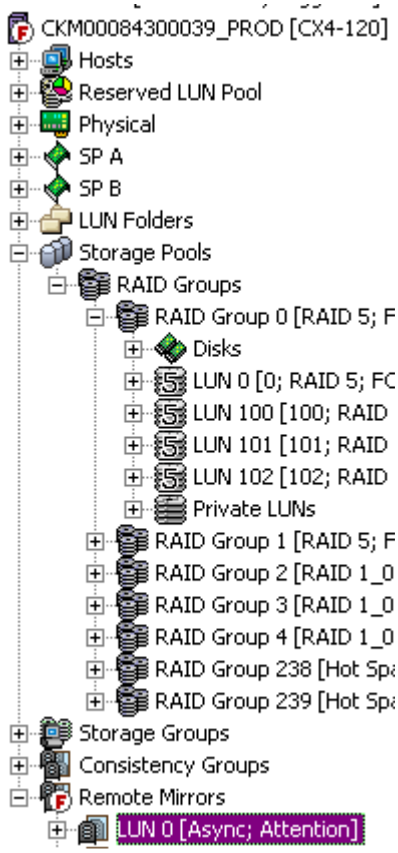

figure 16 - remote mirror fault indicator

Right click on the Remote Mirror and select "Add Secondary Image". This will provide you with the opportunity to establish a relationship between the primary LUN and the Secondary Image LUN that was created back at the start of the process.

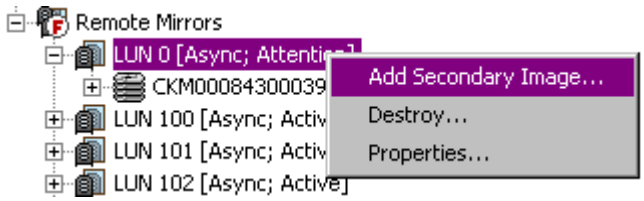

figure 17 - right-click, add secondary image

A Dialogue box will be presented with a number of parameters that can be set. Of note is the "Recovery Policy", which can be set to Automatic or Manual. This dictates whether the array will attempt to recover a MirrorView session if, for some reason, there is a break in communication. The Synchronization rates can be selected (the more mirrors you establish, particularly over Asynchronous iSCSI, the less likely you'll be to want to run things at a High priority). The Update Type, and thus the frequency of updates, can also be set from this dialogue. The update can be Manual, set from Start or Last Update, or End of Last Update. Most people will want to select End of Last Update as the option here, and a figure in number of minutes can be entered here. The figure you enter here will obviously be dependant on your RPO / RTO, and the strength of the link. Obviously, there is no need to think about this with MirrorView/Synchronous as the mirror is "always on".

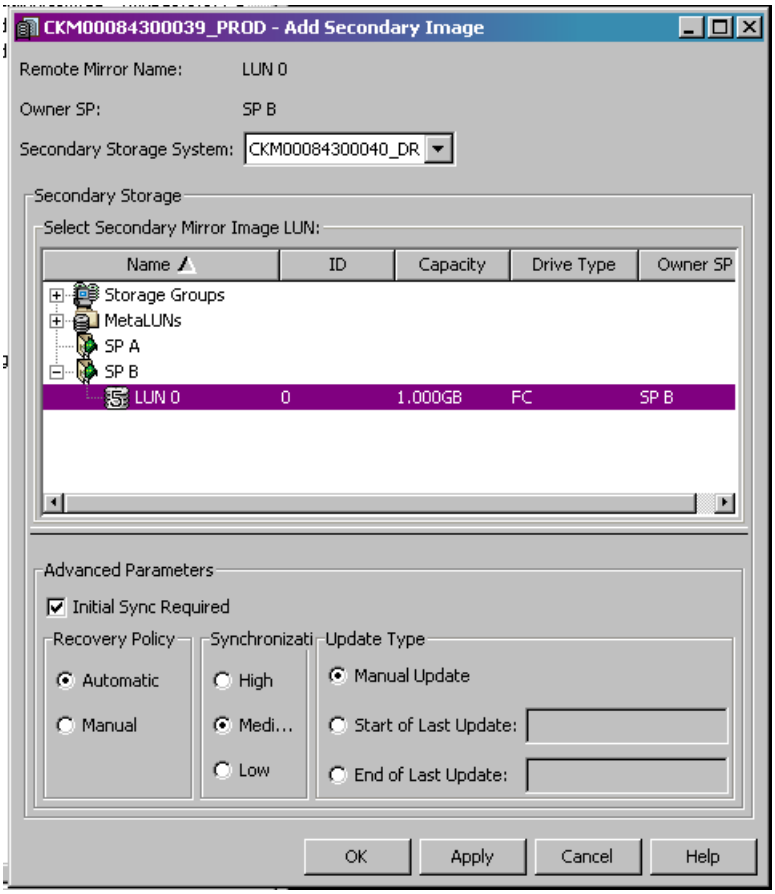

figure 18 - add secondary image dialogue

Just in case you forgot you were working in Navisphere, here's another confirmation box that will pop up once you've clicked okay.

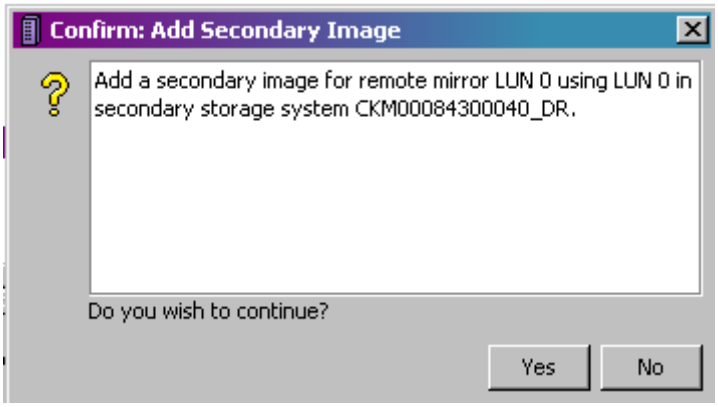

figure 19 - add secondary image confirmation

Notice that the scary little red F has now turned into a friendly blue T that means that the state of the mirror's health is "transitioning" (from bad to good, generally).

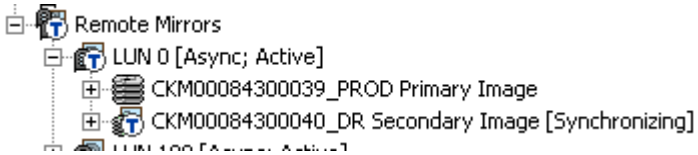

figure 20 - secondary image transition

Here's one I prepared earlier …

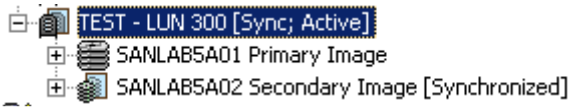

figure 21 - secondary image synchronized

To view information about the Mirror, right-click on the Remote Mirror and select "Properties…"

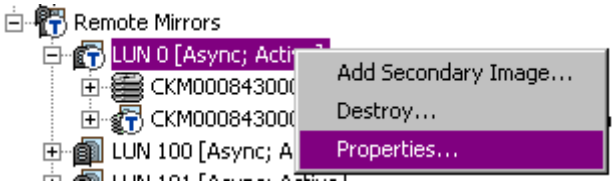

figure 22 - remote mirror properties

At this point you can see general information about the Remote Mirror image, including the mirror type, the storage system it lives on, its name, and the Unique ID. You could also take this opportunity to provide a more verbose description of the remote mirror, or perhaps you could adjust the number of minimum required images.

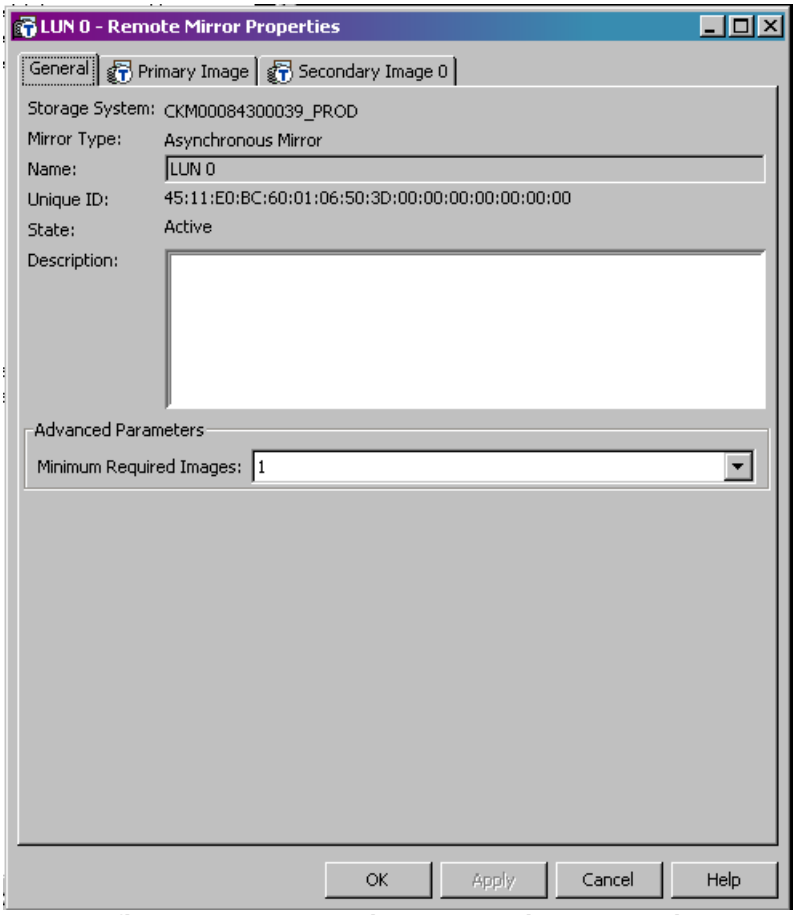

figure 23 - remote mirror properties - general

The Primary Image tab provides information about, well, the primary image in the MirrorView relationship.

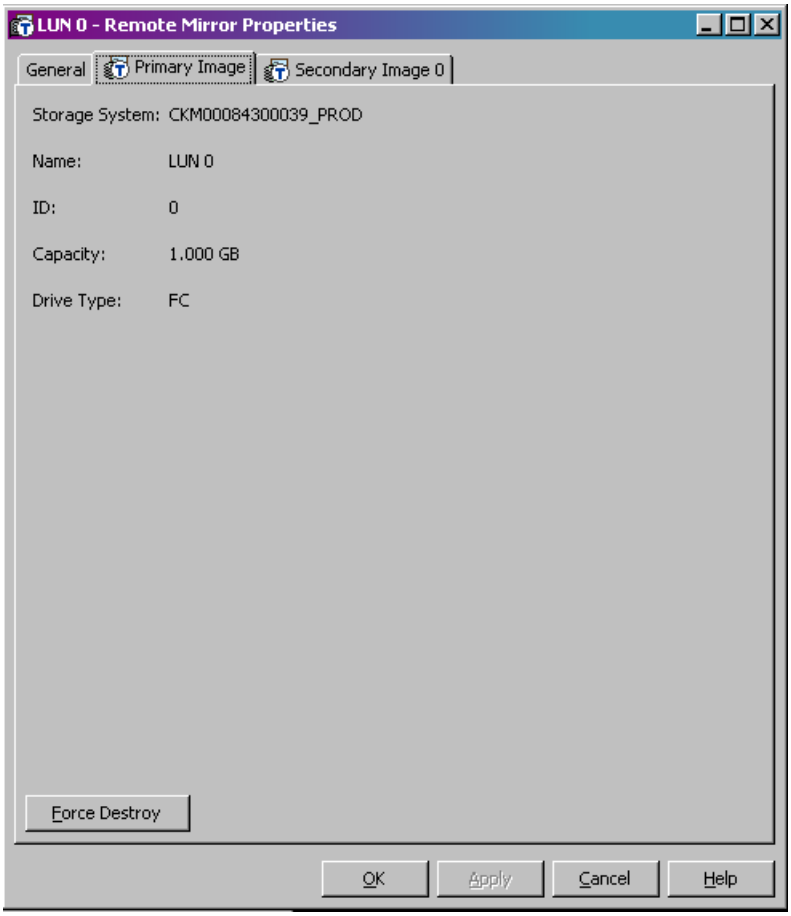

figure 24 - remote mirror properties - primary

That's right, the Secondary Image tab provides information about the secondary image, including its location, the condition of the image, the state and its name. This tab can also be used to modify the parameters that were initially set with the image. The Update Information is useful in determining the approximate progress of a synchronization operation.

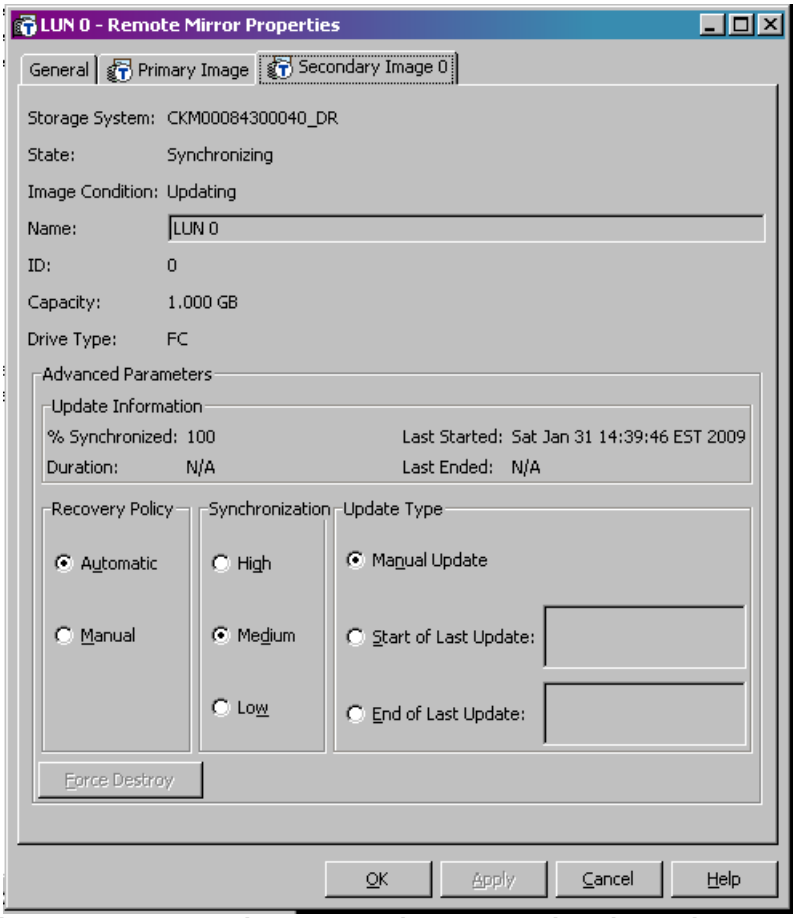

figure 25 - remote mirror properties – secondary (asynchronous)

Here's what the synchronous version looks like.

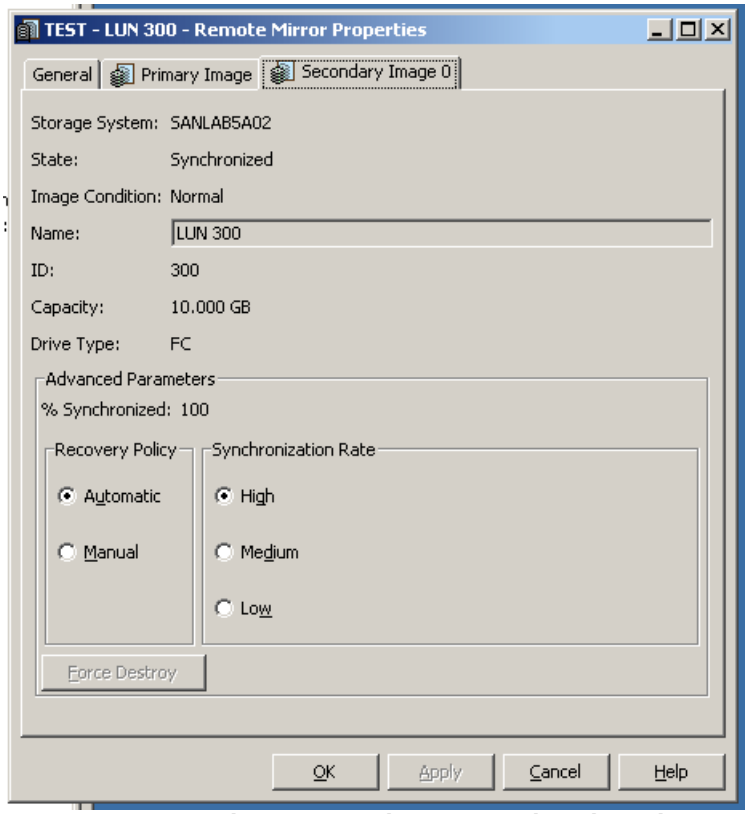

figure 26 - remote mirror properties - secondary (synchronous)

Note in the following screenshot the Image Condition has changed from updating to Normal. I know - it's very exciting isn't it?

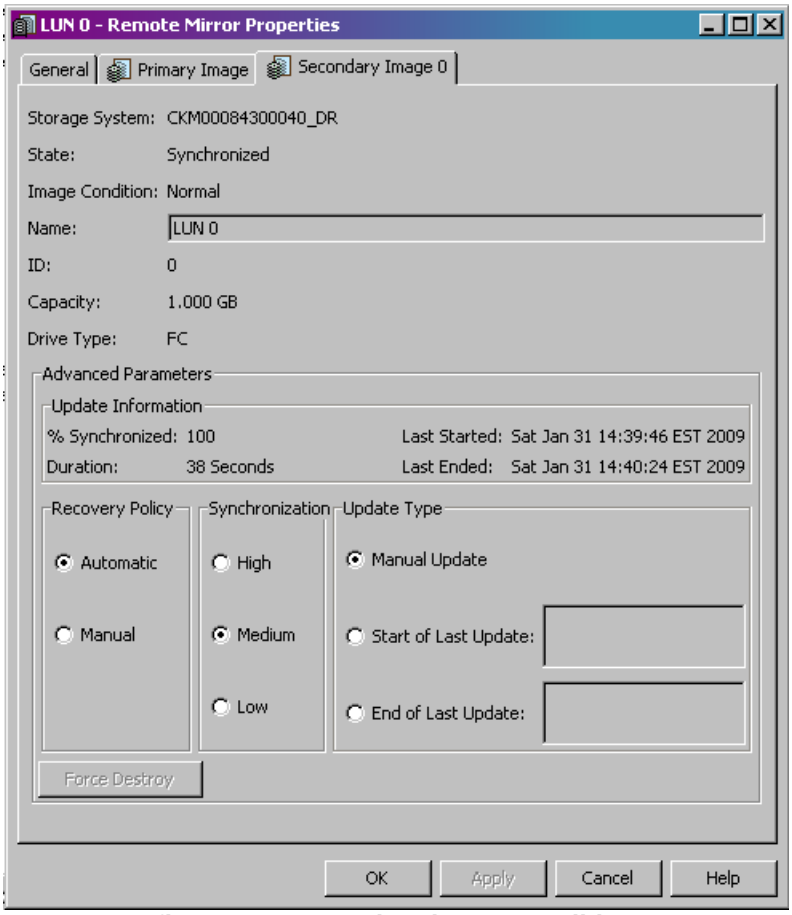

figure 27 - secondary image condition

srm configuration

This document doesn't go into the steps required on the SRM side of the fence to ensure a successful implementation – I'll be doing that part as a complimentary article. Nonetheless, here're the required steps, once you're rolling with MirrorView, to ensure that the SRM implementation will go smoothly.

add mirrors to a consistency group

VMware no longer require replicas to live in a consistency group on the array. But I think it's a helpful thing to have as part of your setup. I find it's simpler to have them in a Consistency Group that corresponds with a recovery group on the SRM server.

To create a Consistency Group in Navisphere, right-click on Consistency Groups and select Create Group.

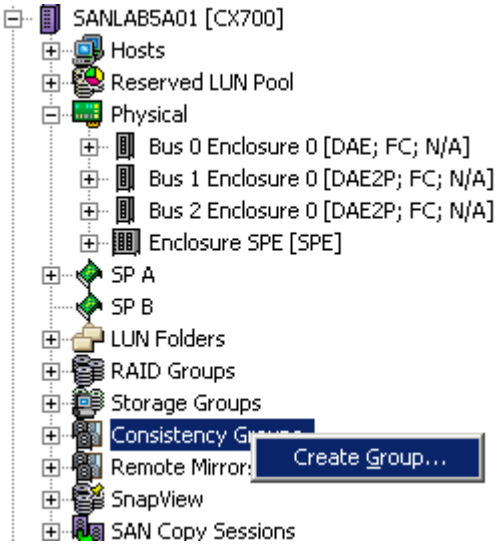

figure 28 - create consistency group

Once this is done, a dialogue box is presented with various options for the group. Note that in my lab example, as I have Asynchronous and Synchronous enablers installed on the array, I can choose to make the Consistency Group either one or the other. However, if I don't have Asynchronous mirrors configured, I will not see anything under the Available Remote Mirrors. So while I could create an Asynchronous Consistency Group, I could not put anything in it.

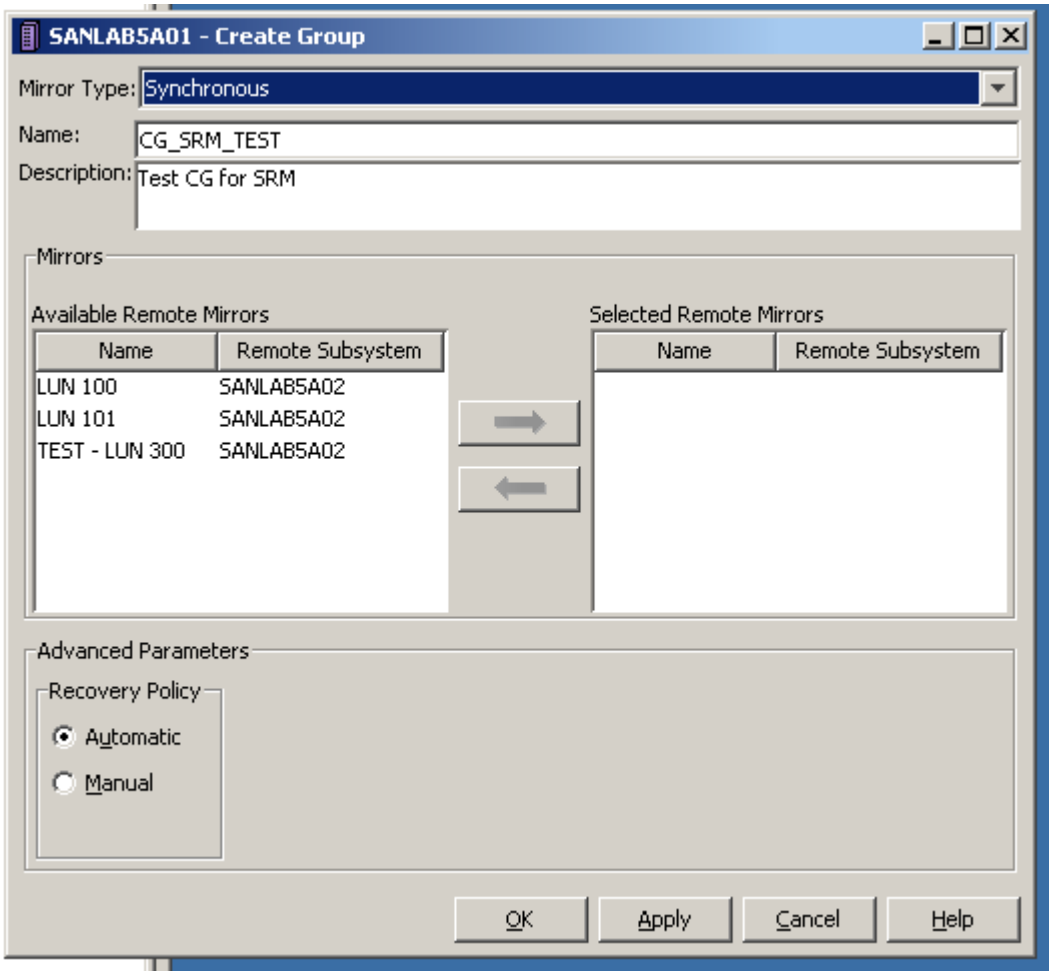

figure 29 - create group

If you don't have both enablers installed, you will not be able to change the Mirror Type. Pictures or it didn't happen? Sure thing.

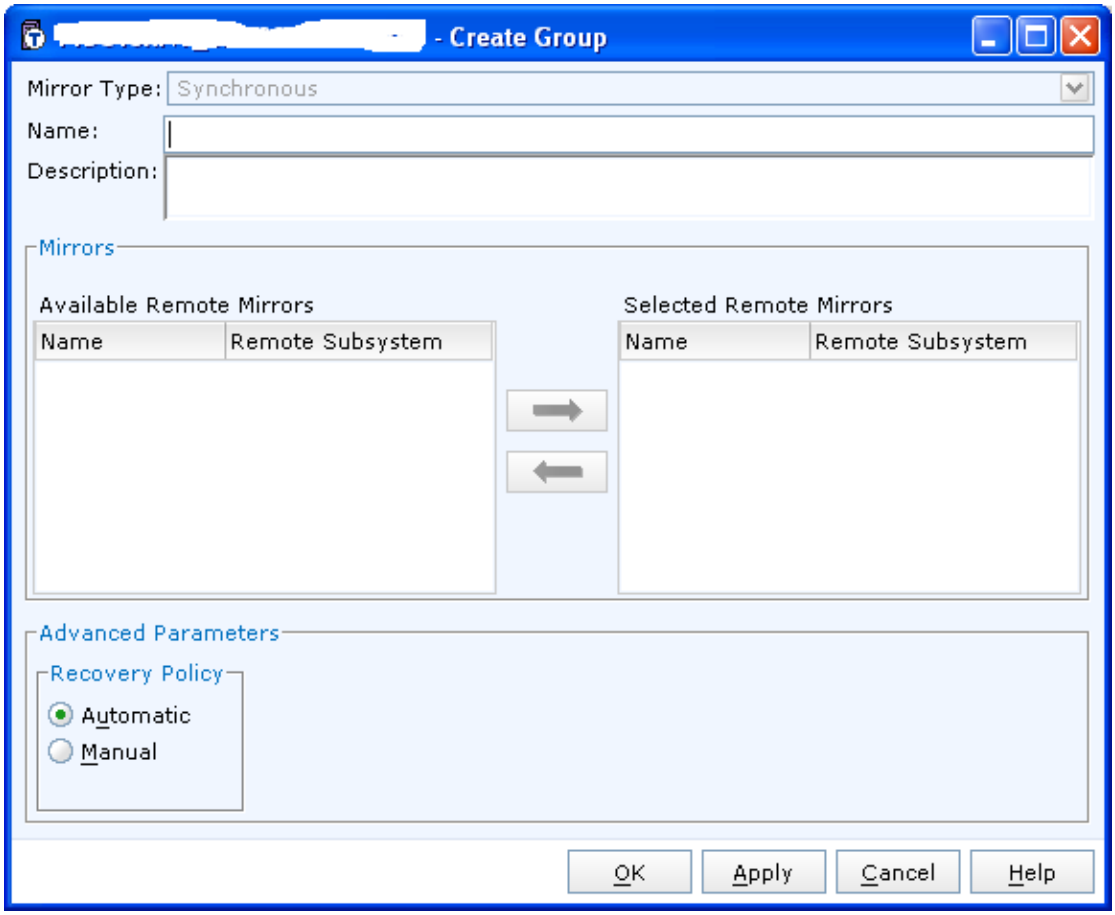

figure 30 - create group - synchronous only

Once you've chosen a suitable Mirror Type, you should add the required mirrors to the group. This was, after all, the point of the exercise. Select the mirrors you require and click on the arrow to move them to Selected Remote Mirrors.

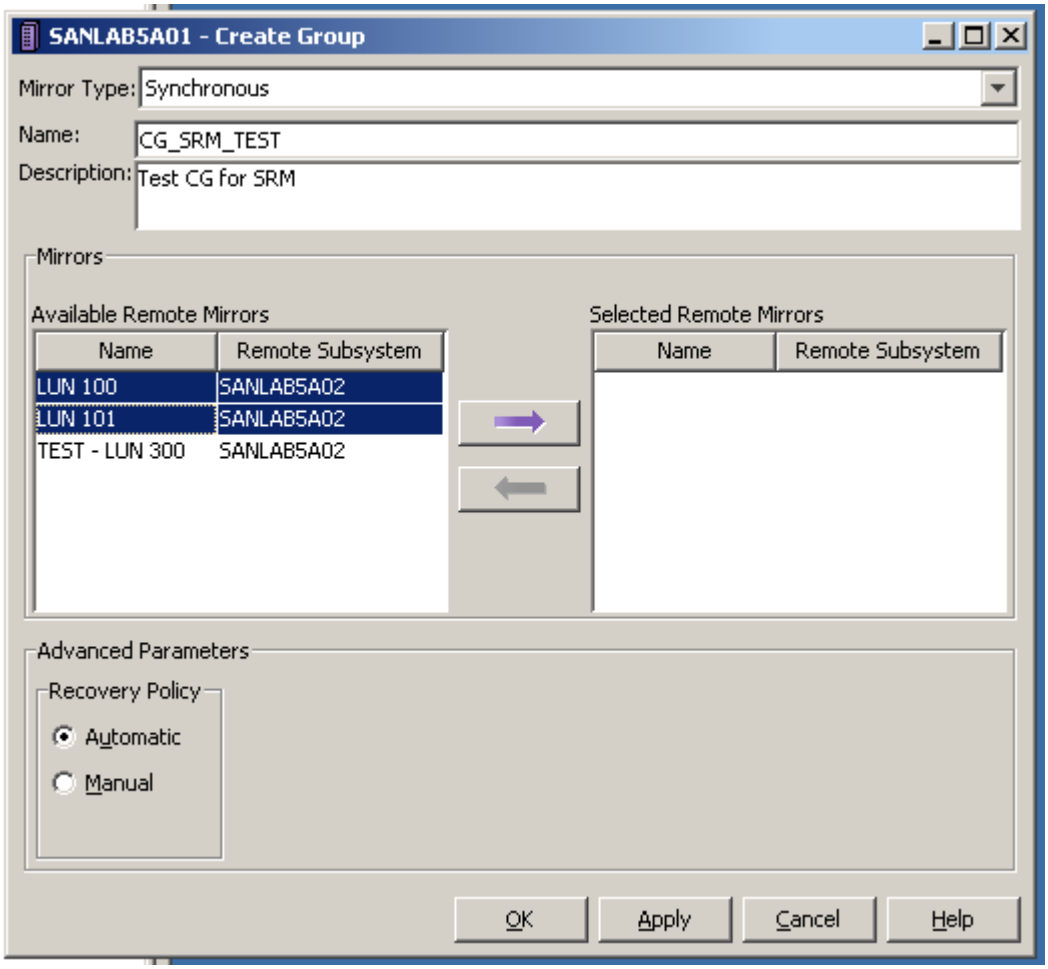

figure 31 - create group - select available remote mirrors

The mirrors will then appear in the Selected Remote Mirrors column. Click on OK and you're good to go.

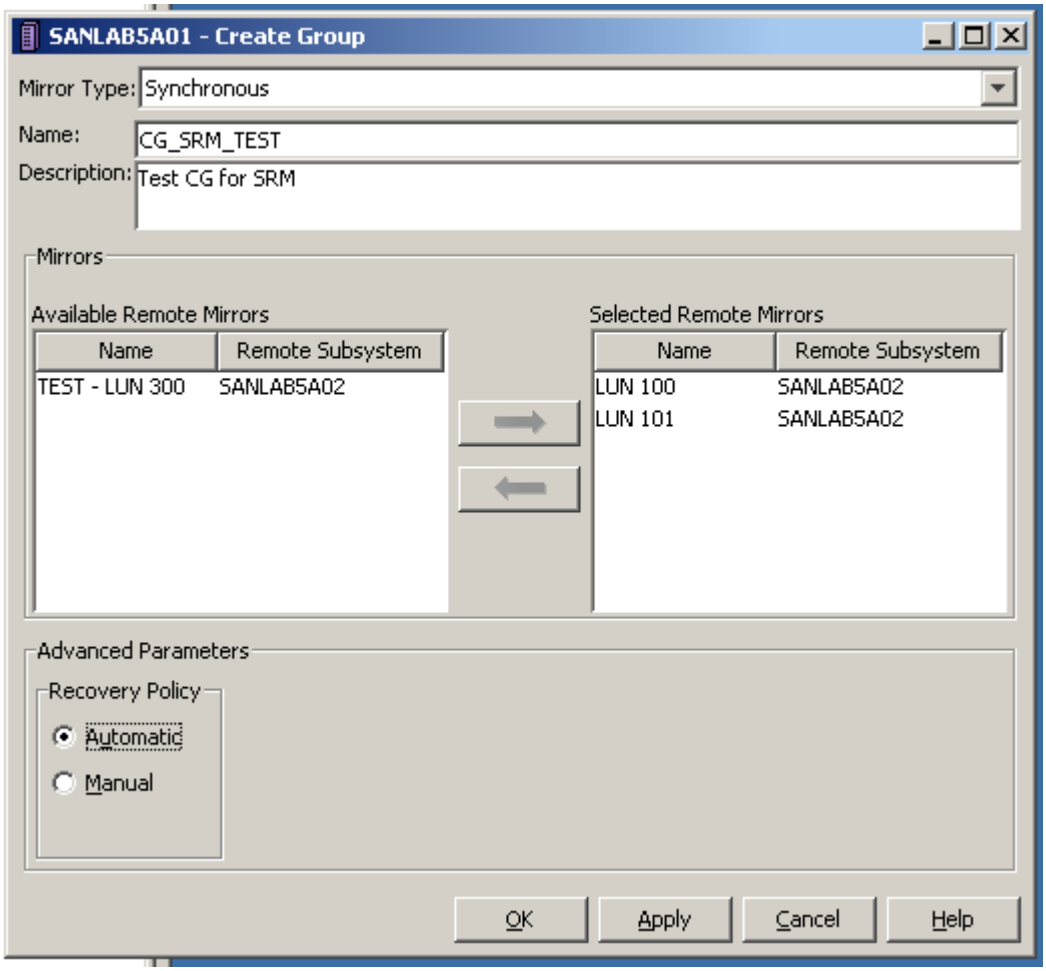

figure 32 - create group - selected remote mirrors

To do the same operation with Unisphere, go to Replicas – Mirrors, and click on Create Clone Consistency Group.

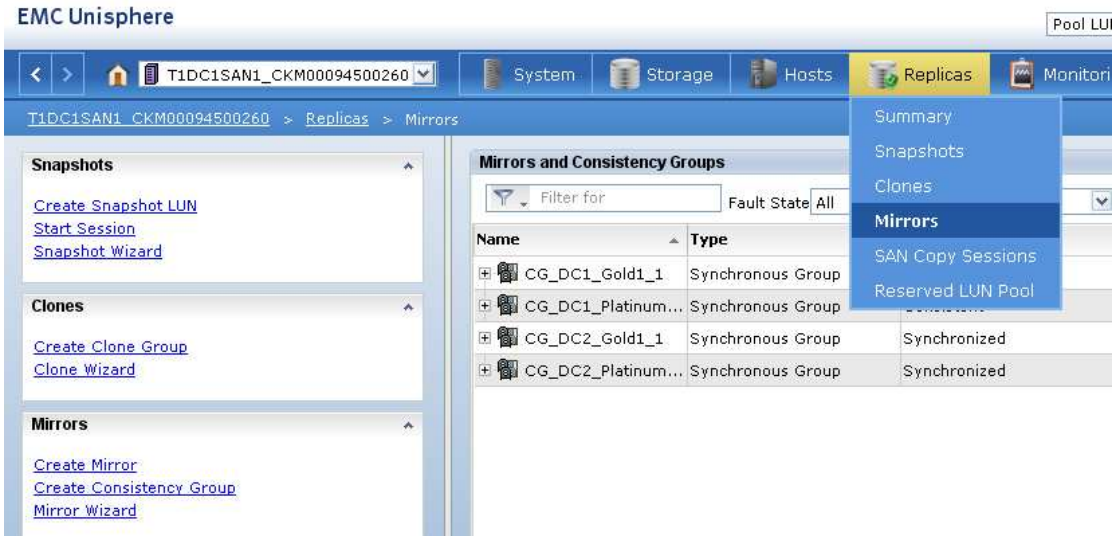

figure 33 - create consistency group – unisphere

From here the process is the same as with Navisphere.

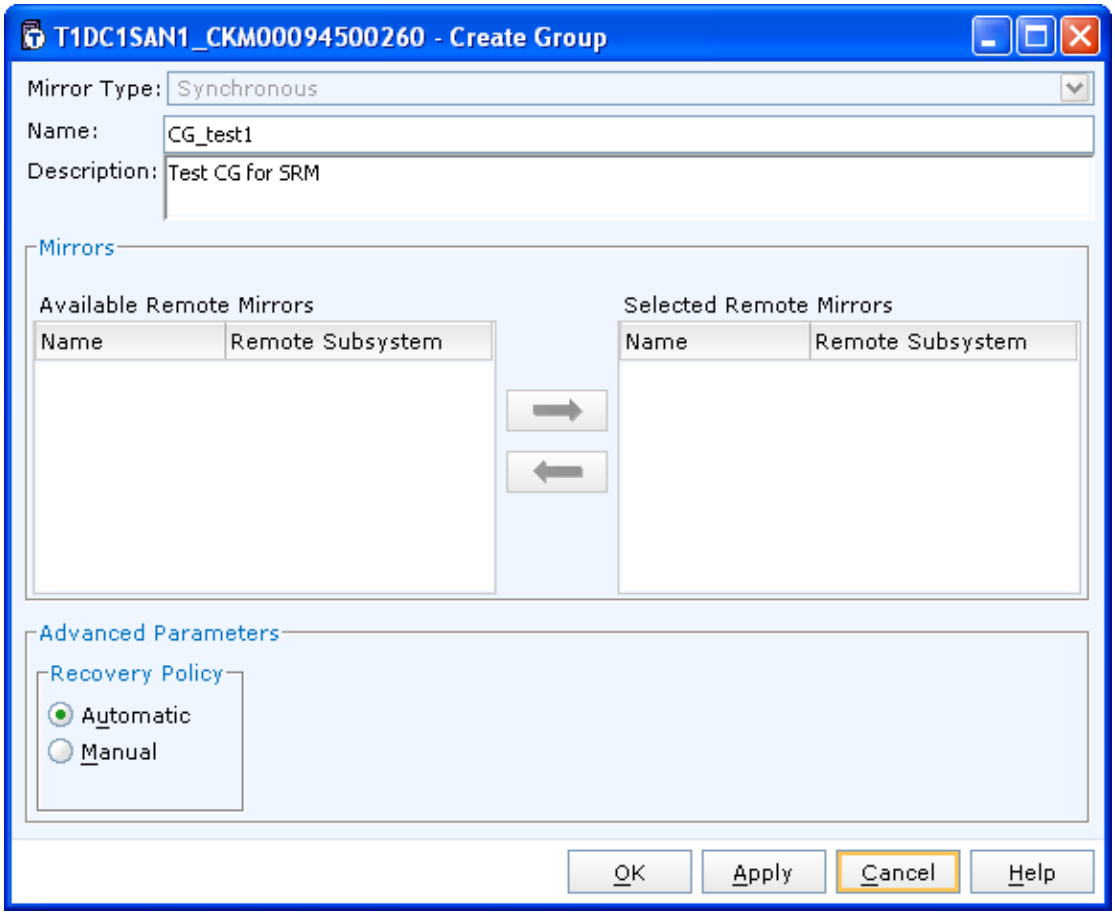

figure 34 - create consistency group - unisphere

adding a replica to the storage group

In order for test failovers to work, the replica of the primary storage should be added to the storage group of the hosts at the recovery site. This is done in the same fashion as normal LUN / Storage Group presentation activities are performed.

create snapshot for test failover

You will also need to create snapshots of the replicas on the recovery site array in order to facilitate test failovers with SRM. It is critical that these snapshots have the text VMWARE_SRM_SNAP in the name somewhere in order for EMC's SRA to work properly.

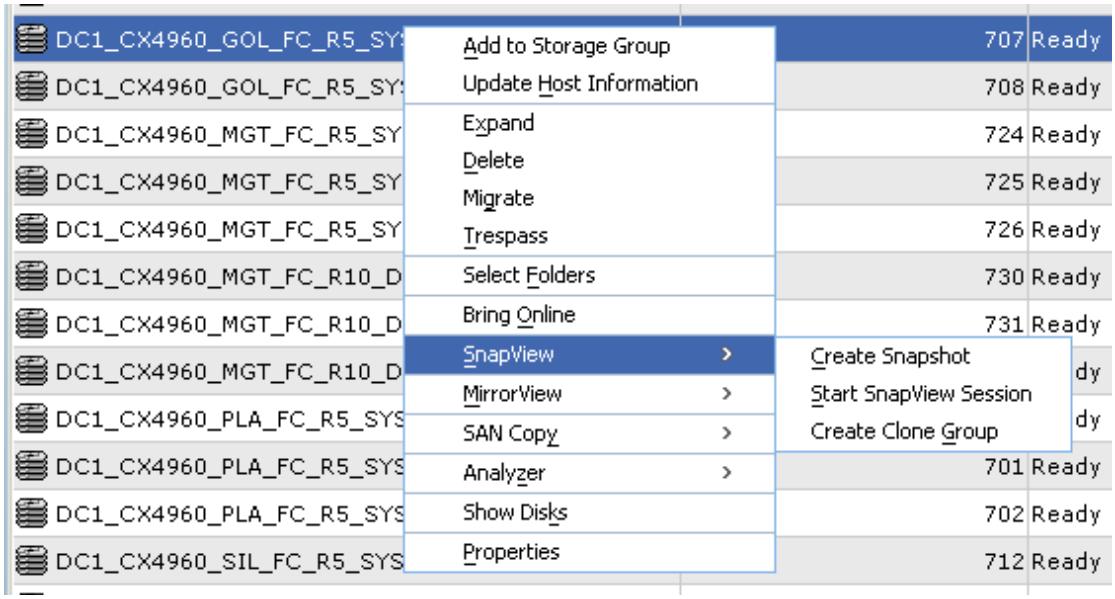

figure 35 - snapview - create snapshot

As stated previously, it is imperative that the snapshot has VMWARE_SRM_SNAP in the name somewhere.

figure 36 - provide a snapshot name

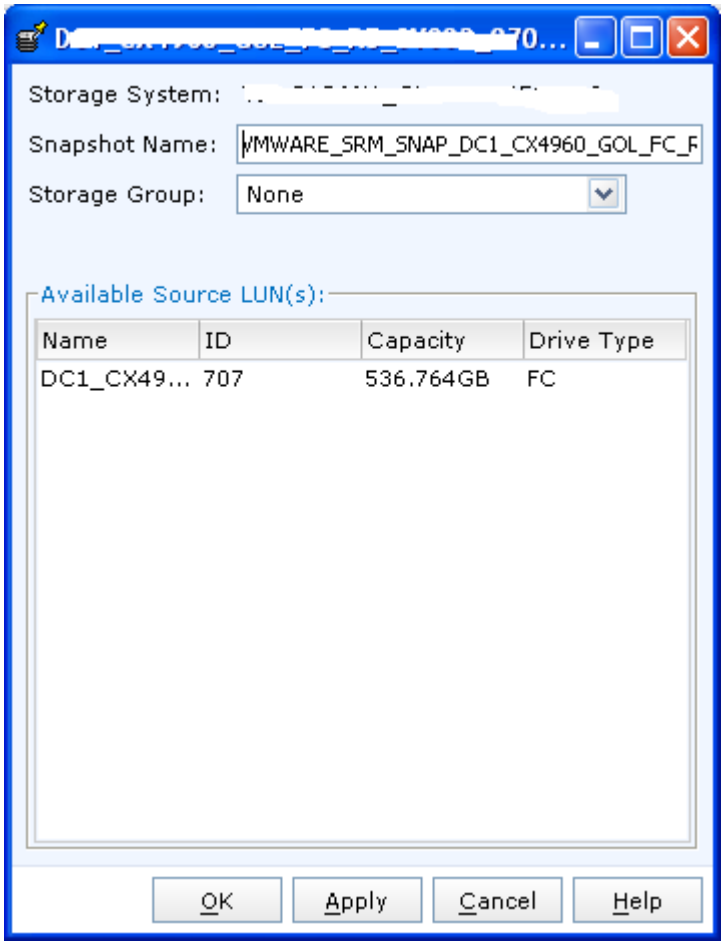

figure 37 - provide snapshot name

Once you've entered the appropriate details, you'll need to confirm that you think you know what you're doing.

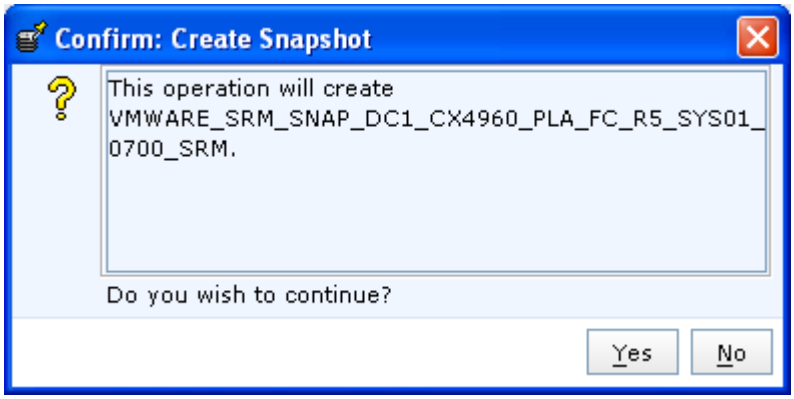

figure 38 - confirm snapshot creation

Navisphere and Unisphere will both confirm that you've been successful in creating the snapshot.

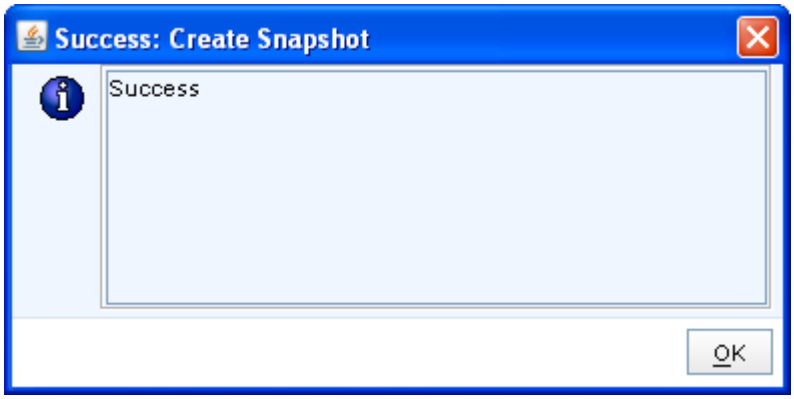

figure 39 - create snapshot - success

add the snapshot to storage group

Once the snapshot has been created on the recovery site it needs to be added to the storage group on the array in order for test failover activities to be performed. You can do this in the same way as you added your replica to the recovery site's machine's storage group, which is the same way as you would present any LUN to any host.

manual operations

I'd like to add some more to this section, but manually fracturing mirrors was all I could think of for the moment.

fracture mirrors manually (admin fracture)

Occasionally there'll be a need to pause replication between arrays. In MirrorView speak, the replicas become fractured. System fractured is where communication between the arrays (either FC or iSCSI) has been lost. This is usually a bad thing, depending on the nature of the event that caused the loss of communication.

However, at times you'll want to pause replication if you are doing a code upgrade on the array or need to perform some sort of maintenance activity. In this instance, you want the mirrors to be Admin Fractured.

Here're the steps to do that.

Right-click on the Secondary Image and select Fracture.

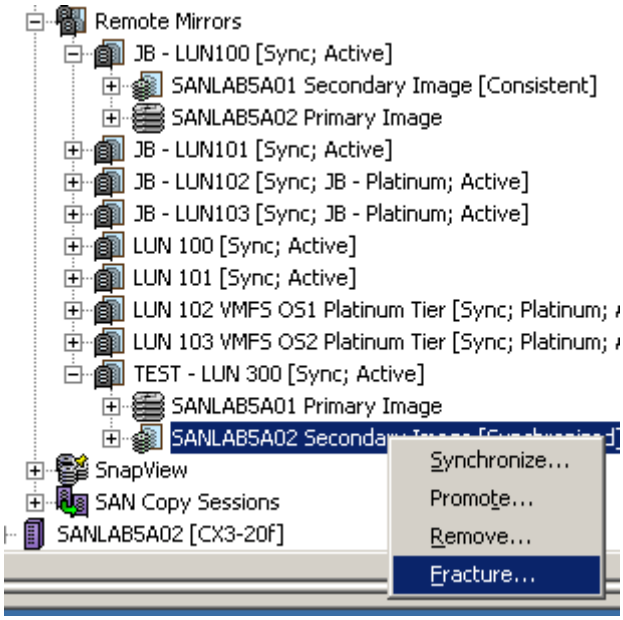

figure 40 - admin fracture secondary image

You'll be warned that things could get a little rocky. Like all good IT guys, you'll ignore the warning and proceed regardless. Good for you.

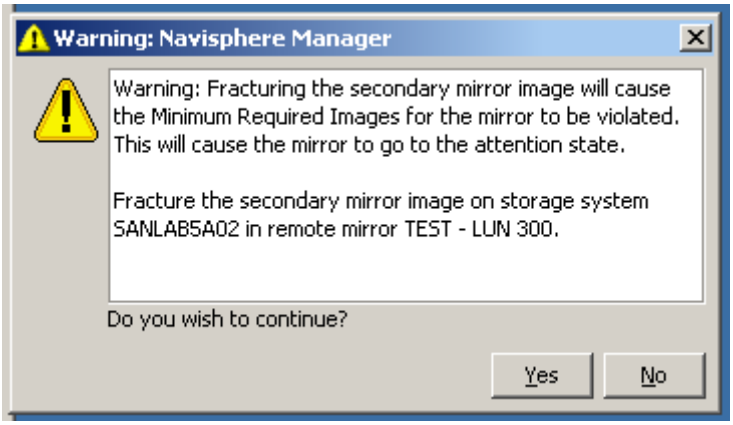

figure 41 - warning - admin fracture

Once the replicas have been successfully fractured, a message will pop up indicating as much.

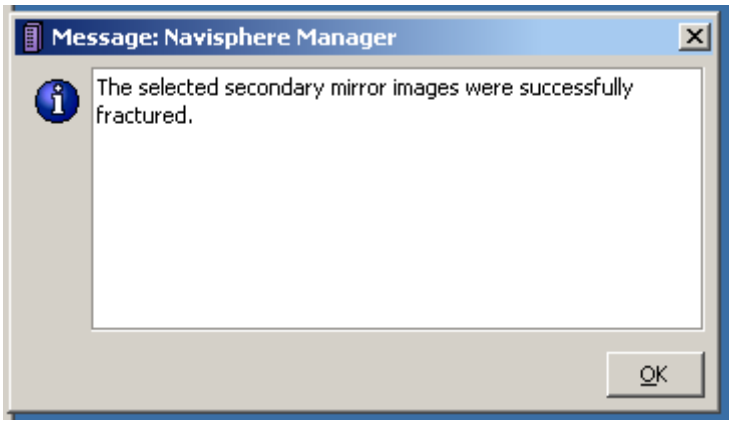

figure 42 - admin fracture success

At this stage, the big, red F will appear on the mirrors that you've fractured. This is to remind you that something's not quite right.

figure 43 - remote mirror attention

So go ahead, do your maintenance or whatever. When you're ready for everything to continue replicating, right-click on the Secondary Image and select Synchronize.

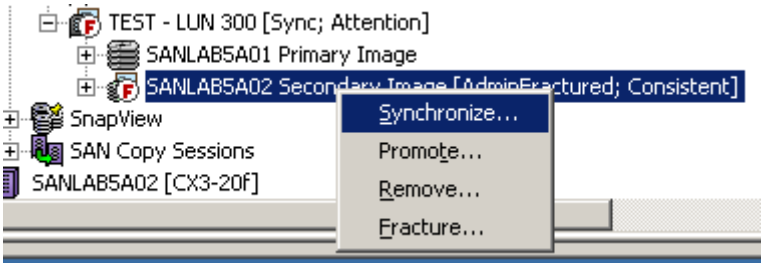

figure 44 - re-synchronize

Another warning will pop up, just in case you forgot what it was that you were doing.

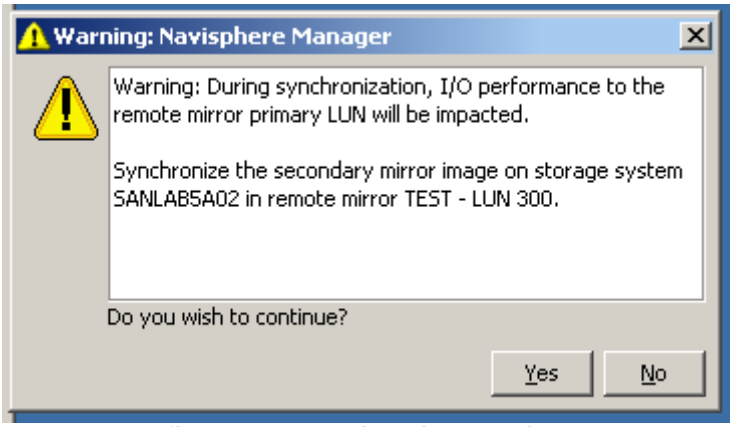

figure 45 - synchronize warning

Once the replicas have started synchronizing again, you'll receive a message to let you know that the synchronization has been successfully initiated. Where once there was an F, now there's a blue, calming T to indicate that the state of the mirror is transitioning. Navisphere / Unisphere is nothing if not chatty.

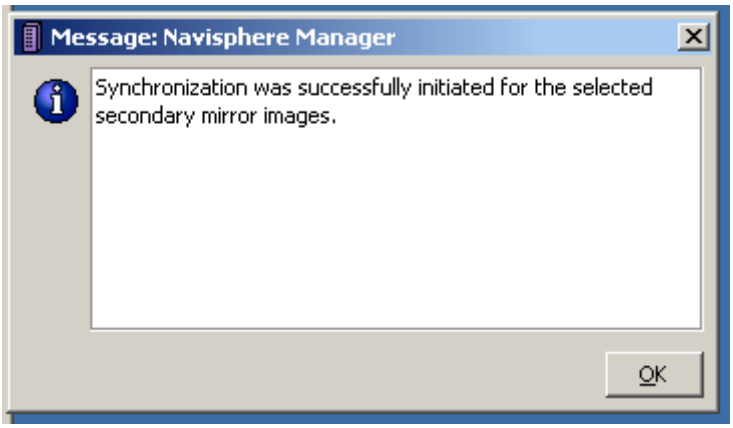

figure 46 - synchronization success

conclusion

EMC MirrorView doesn't have to be difficult. So don't let people tell you it is. Sure, wrapping your brain around how the updates are actually done under the hood can be a little confusing at first, but from a simple implementation / operational perspective, this document should demonstrate that it just isn't that hard. I hope this has been useful.