PowerNSX – Convert2EntityBelongsTo.ps1

Author: Dale Coghlan (dcoghlan@vmware.com)

When it comes to configuring NSX for vSphere (NSX-v), we often see customers configuring NSX Security Groups in such a way that they want to include a NSX Security Tag but they configure the NSX Security Group with a dynamic criteria of "Security Tag" & "Equals to". Whilst this achieves the desired outcome, under the covers this is doing a regex styled lookup for the name of the security tag to match against, and because it is configured to be an exact match, there should only be 1 NSX Security Tag that matches. This regex styled lookup incurs additional overhead on the NSX Manager. But never fear, there is a solution that will lighten the load on your NSX Manager, and that is to convert from "Security Tag" & "Equal to" to "Entity" & "Belongs to".

But what happens when you've got potentially hundreds, or thousands of these littered throughout your security group configurations? Well we've written a PowerNSX script that can help you out in this department. It is hosted in the Examples folder of the PowerNSX GitHub repo, along with other example scripts you can play with.

PowerNSX Github Repository
https://github.com/vmware/powernsx

Convert2EntityBelongsTo.ps1
https://github.com/vmware/powernsx/blob/master/Examples/Convert2EntityBelongsTo.ps1

Important Note: The script requires PowerNSX to run.

Running the script with no parameters will just read your configuration and see if it finds any instances of dynamic criteria configured with "Security Tag" & "Equals to".

```
C:\Users\Administrator\Documents> .\Convert2EntityBelongsTo.ps1
18:19:51 : Line(187) : Log file not found... creating a new one
18:19:51 : Line(187) : Logging to file .\Convert2EntityBelongsTo_2018_02_28_18_19_51.log
18:19:51 : Line(187) : Retrieving local NSX Security Groups
18:19:51 : Line(199) : Searching for Security Groups with 'Security Tag' & 'Equals'
configured in dynamic criteria
18:19:51 : Line(299) : Script complete in 00:00:00.2184004
18:19:51 : Line(300) : Debug log saved to .\Convert2EntityBelongsTo_2018_02_28_18_19_51.log
C:\Users\Administrator\Documents>
```

The output shown above is from a freshly deployed NSX 6.4.0 test environment with no NSX Security Tags or NSX Security Groups configured.

To setup some test objects, you can specify the **-CreateTestEnvironment** parameter, and after a small amount of time, it will create some test NSX Security Tags and NSX Security Groups.

Now run the script again and see the output.

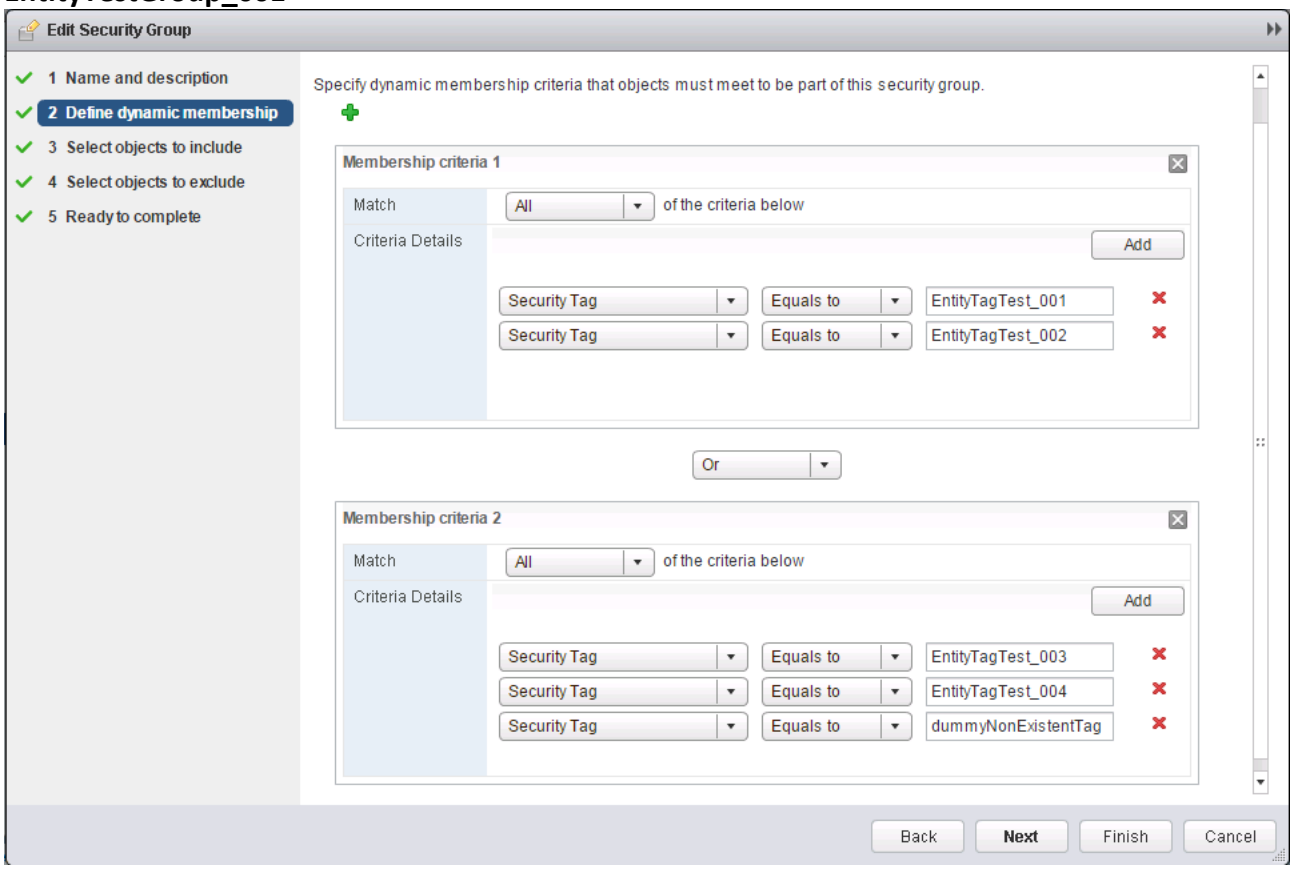
```
C:\Users\Administrator\Documents> .\Convert2EntityBelongsTo.ps1
18:23:25 : Line(187) : Log file not found... creating a new one
18:23:25 : Line(187) : Logging to file .\Convert2EntityBelongsTo_2018_02_28_18_23_25.log
18:23:25 : Line(187) : Retrieving local NSX Security Groups
18:23:26 : Line(199) : Searching for Security Groups with 'Security Tag' & 'Equals'
configured in dynamic criteria
18:23:26 : Line(208) : Instances where 'Security Tag' & 'equals' is configured : 12
18:23:26 : Line(209) : Adding Security Group names to logfile.
18:23:26 : Line(299) : Script complete in 00:00:00.2184003
18:23:26 : Line(300) : Debug log saved to .\Convert2EntityBelongsTo_2018_02_28_18_23_25.log
C:\Users\Administrator\Documents>
```

If we take a look in the log file, it will show which security groups contain the 12 dynamic criteria instances we are interested in.

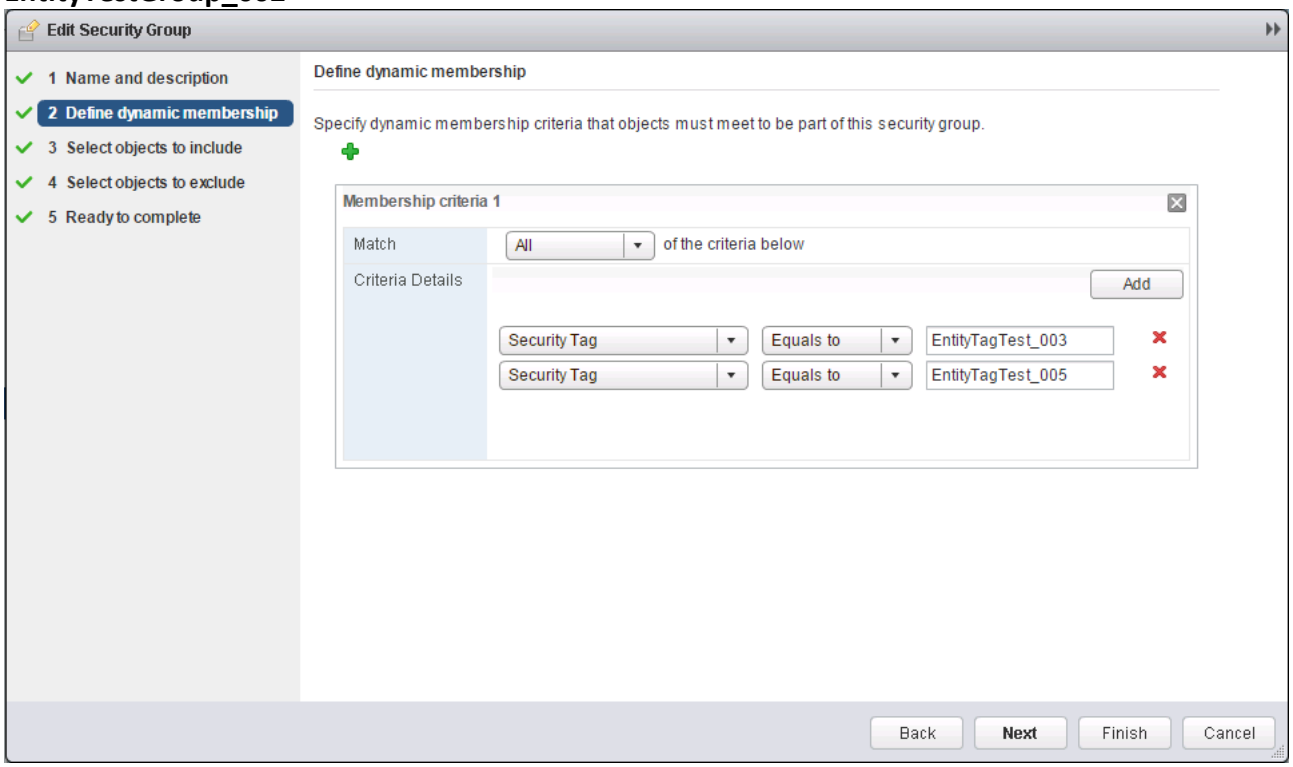
```
<snip>
18:23:26 : Line(199) : Searching for Security Groups with 'Security Tag' & 'Equals' configured in dynamic
criteria
18:23:26 : Line(208) : Instances where 'Security Tag' & 'equals' is configured : 12
18:23:26 : Line(209) : Adding Security Group names to logfile.
18:23:26 : Line(217) : Adding to hashtable (securityGroupsAffected) : EntityTestGroup_003(securitygroup-12)
18:23:26 : Line(217) : Adding to hashtable (securityGroupsAffected) : EntityTestGroup_004(securitygroup-13)
18:23:26 : Line(217) : Adding to hashtable (securityGroupsAffected) : EntityTestGroup_002(securitygroup-11)
18:23:26 : Line(217) : Adding to hashtable (securityGroupsAffected) : EntityTestGroup_001(securitygroup-10)
</snip>
```

Take a look at what those 12 instances actually are (keep scrolling):

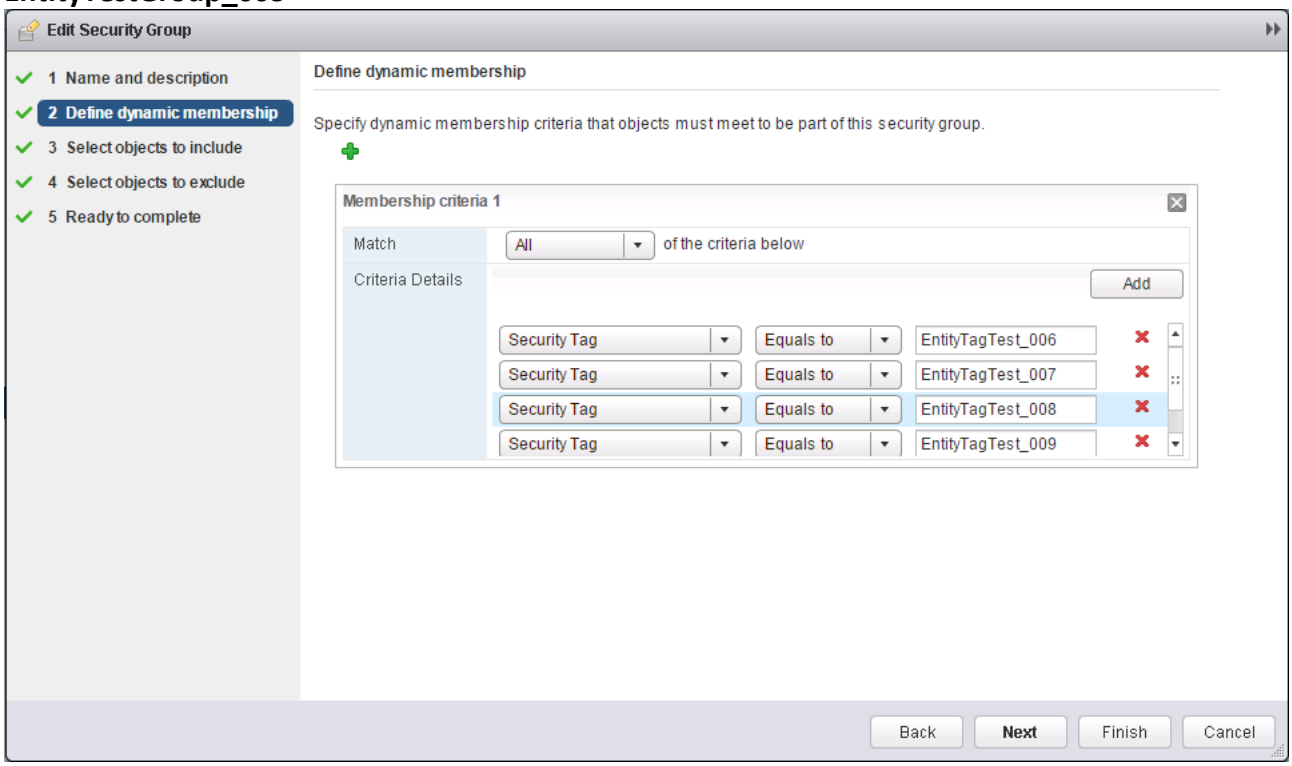
EntityTestGroup_001



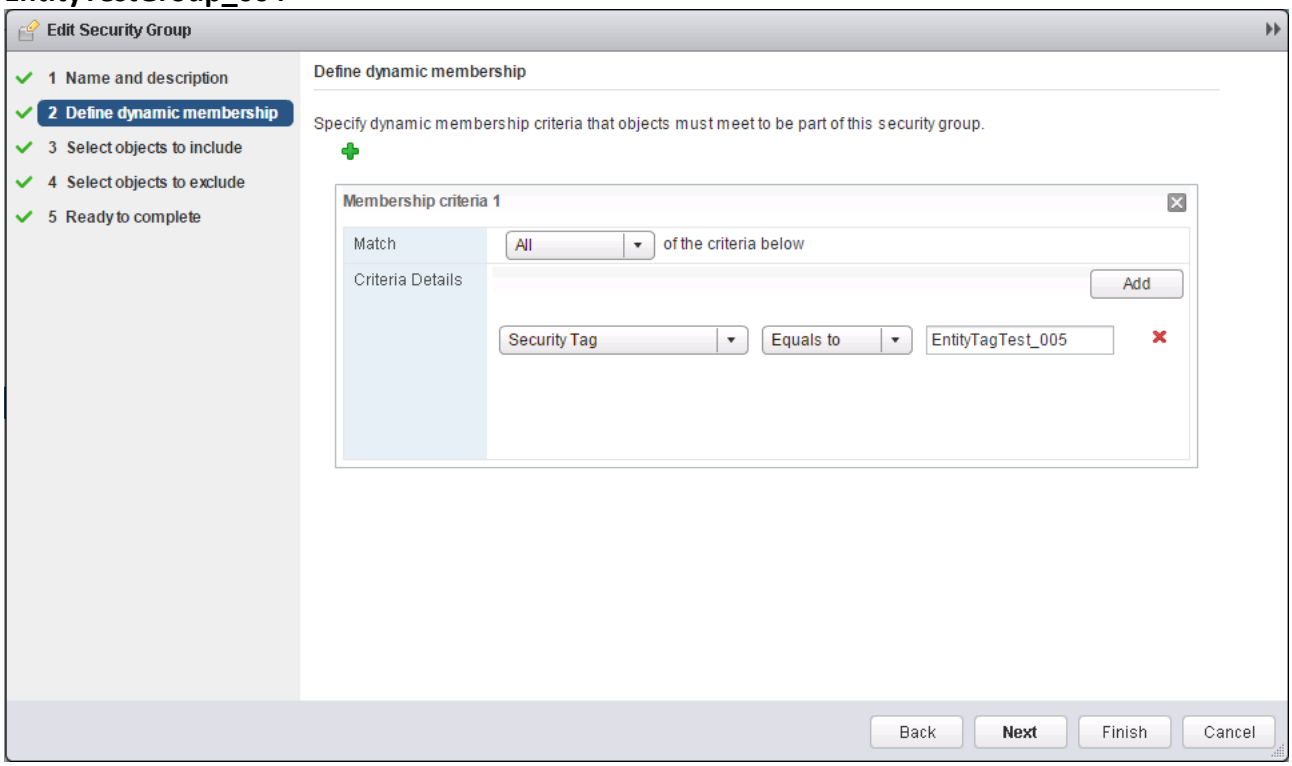
EntityTestGroup_002



EntityTestGroup_003



EntityTestGroup_004



The script can also optimise the configuration so that if it finds a valid corresponding NSX Security Tag, it will replace "Security Tag" & "Equals to" with "Entity" & "Belongs to". If there is no NSX Security Tag configured that matches the name specified, the script will not touch the criteria.

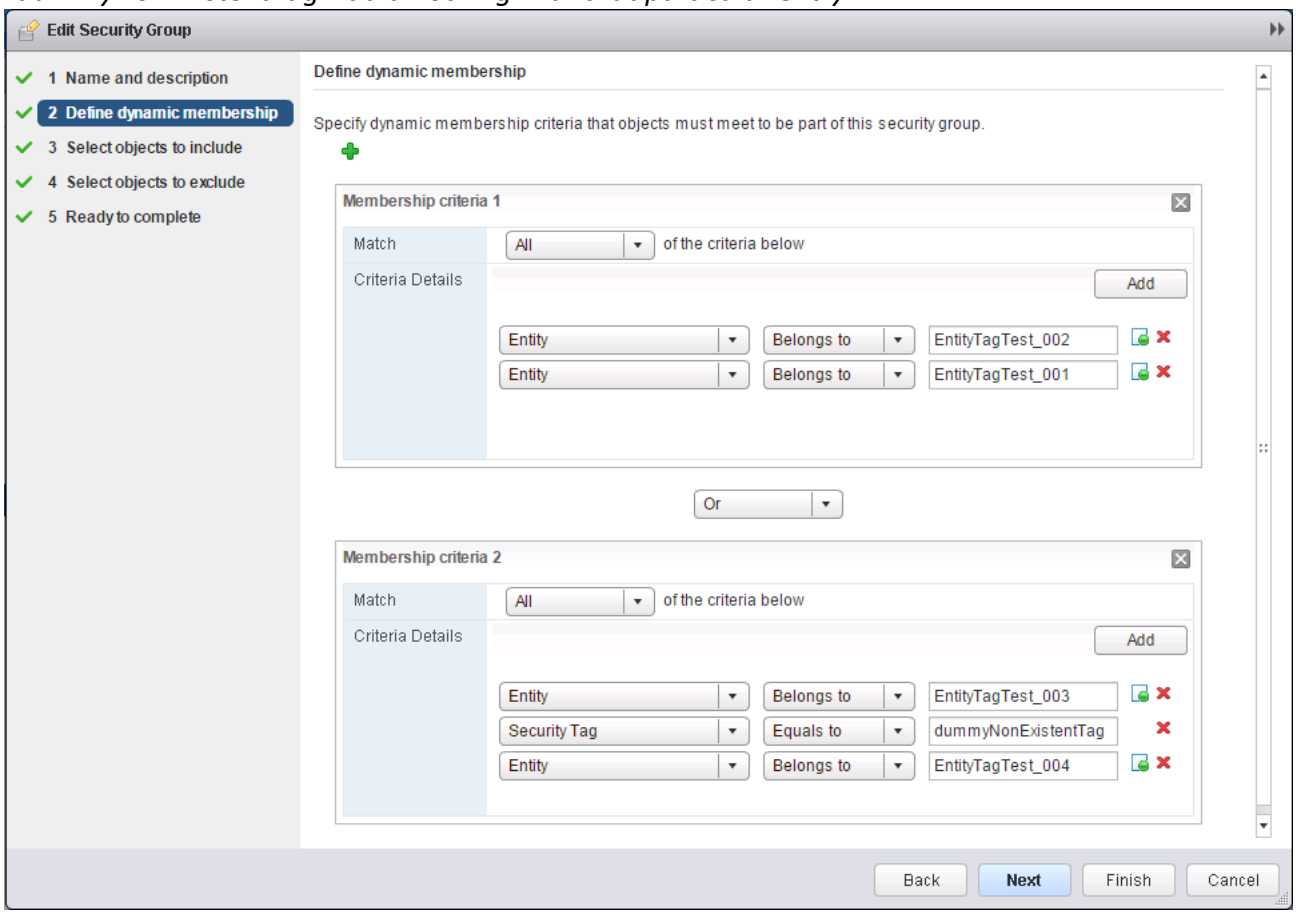
And lastly, if there is only a single criteria configured in the criteria set with "Security Tag" & "Equals to", it will add the NSX Security Tag as a static include member and remove the criteria set from the security group, all automagically for you.

To run the script so that it will make the required changes, supply the -**DoTheNeedful** parameter, and you'll see something similar to the following on the screen:

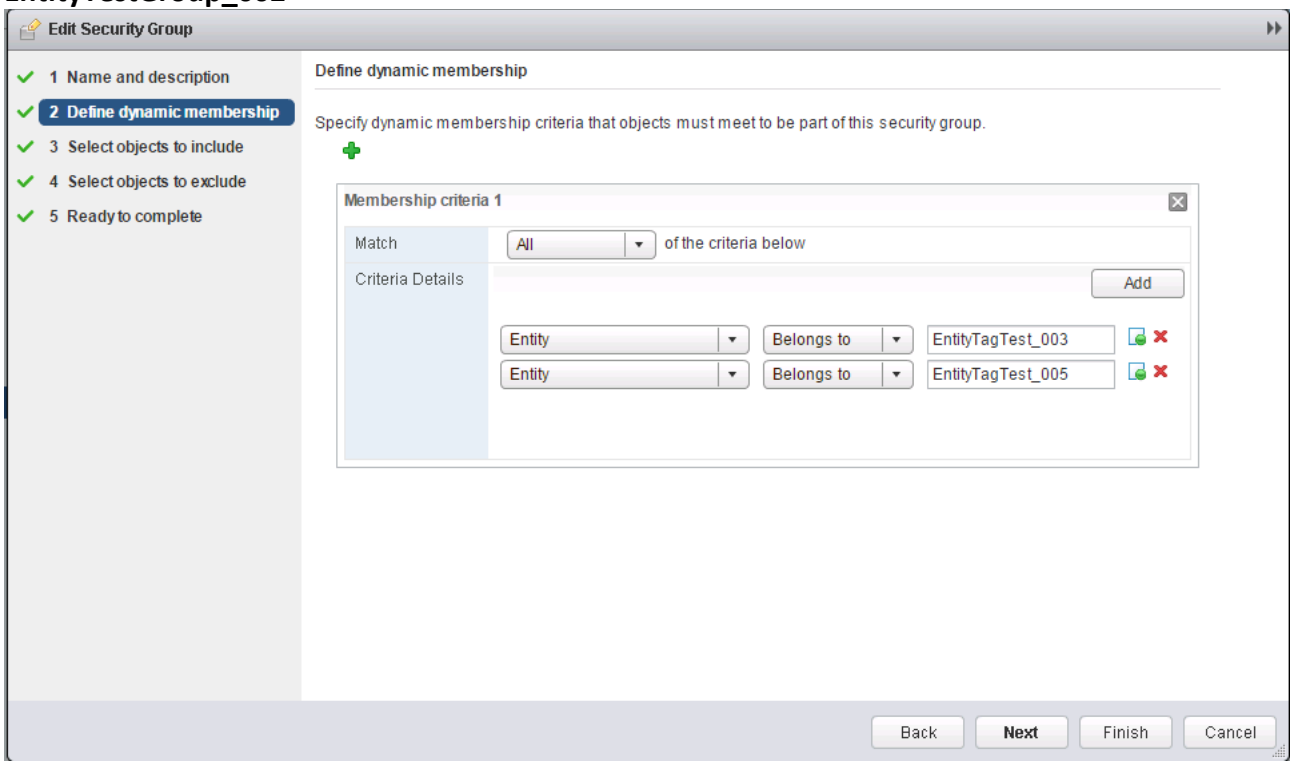
```
C:\Users\Administrator\Documents> .\Convert2EntityBelongsTo.ps1 -DoTheNeedful
18:40:31 : Line(187) : Log file not found... creating a new one
18:40:31 : Line(187) : Logging to file .\Convert2EntityBelongsTo_2018_02_28_18_40_31.log
18:40:31 : Line(187) : Retrieving local NSX Security Groups
18:40:31 : Line(199) : Searching for Security Groups with 'Security Tag' & 'Equals'
configured in dynamic criteria
18:40:31 : Line(208) : Instances where 'Security Tag' & 'equals' is configured : 12
18:40:31 : Line(209) : Adding Security Group names to logfile.
18:40:31 : Line(225) : Retrieving local NSX Security Tags
18:40:31 : Line(235) : Retrieving latest Security Group configuration :
EntityTestGroup_003(securitygroup-12)
18:40:31 : Line(260) : Found exact match for Security Tag : EntityTagTest_007(securitytag-18)
18:40:32 : Line(260) : Found exact match for Security Tag : EntityTagTest_008(securitytag-19)
18:40:32 : Line(260) : Found exact match for Security Tag : EntityTagTest_009(securitytag-20)
18:40:32 : Line(260) : Found exact match for Security Tag : EntityTagTest_006(securitytag-17)
18:40:32 : Line(288) : Updating Security Group : EntityTestGroup_003(securitygroup-12)
18:40:33 : Line(235) : Retrieving latest Security Group configuration :
EntityTestGroup_004(securitygroup-13)
18:40:33 : Line(260) : Found exact match for Security Tag : EntityTagTest_005(securitytag-16)
18:40:33 : Line(288) : Updating Security Group : EntityTestGroup_004(securitygroup-13)
18:40:34 : Line(235) : Retrieving latest Security Group configuration :
EntityTestGroup_001(securitygroup-10)
18:40:34 : Line(260) : Found exact match for Security Tag : EntityTagTest_001(securitytag-12)
18:40:34 : Line(260) : Found exact match for Security Tag : EntityTagTest_002(securitytag-13)
18:40:34 : Line(260) : Found exact match for Security Tag : EntityTagTest_003(securitytag-14)
18:40:34 : Line(260) : Found exact match for Security Tag : EntityTagTest_004(securitytag-15)
18:40:34 : Line(258) : Criteria Skipped - Exact match NOT found for Security Tag :
dummyNonExistentTag
18:40:34 : Line(288) : Updating Security Group : EntityTestGroup_001(securitygroup-10)
18:40:35 : Line(235) : Retrieving latest Security Group configuration :
EntityTestGroup_002(securitygroup-11)
18:40:36 : Line(260) : Found exact match for Security Tag : EntityTagTest_003(securitytag-14)
18:40:36 : Line(260) : Found exact match for Security Tag : EntityTagTest_005(securitytag-16)
18:40:36 : Line(288) : Updating Security Group : EntityTestGroup_002(securitygroup-11)
18:40:37 : Line(299) : Script complete in 00:00:06.2942426
18:40:37 : Line(300) : Debug log saved to .\Convert2EntityBelongsTo_2018_02_28_18_40_31.log
C:\Users\Administrator\Documents>
```

This is what the security groups look like after the modifications (keep scrolling):

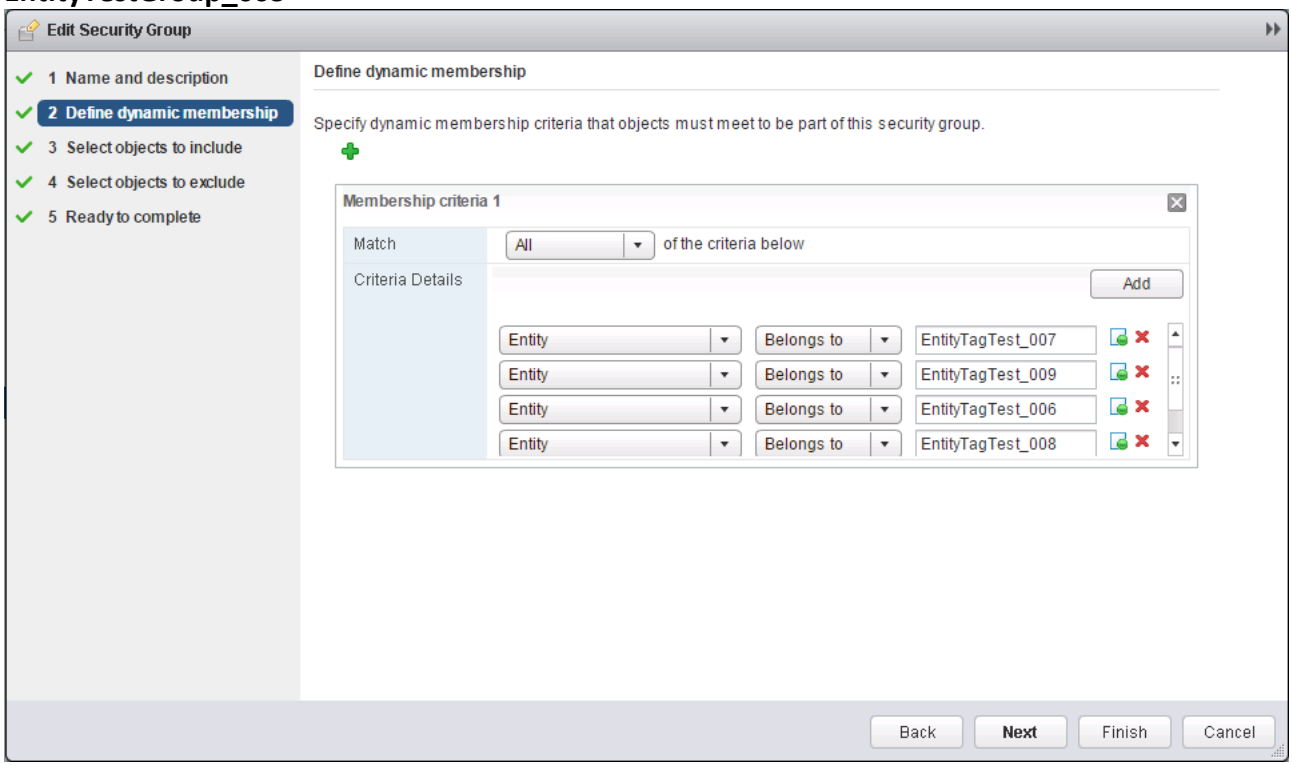
EntityTestGroup_001 - *Notice that when it couldn't find the NSX Security Tag "dummyNonExistentTag" it did nothing with that particular entry*



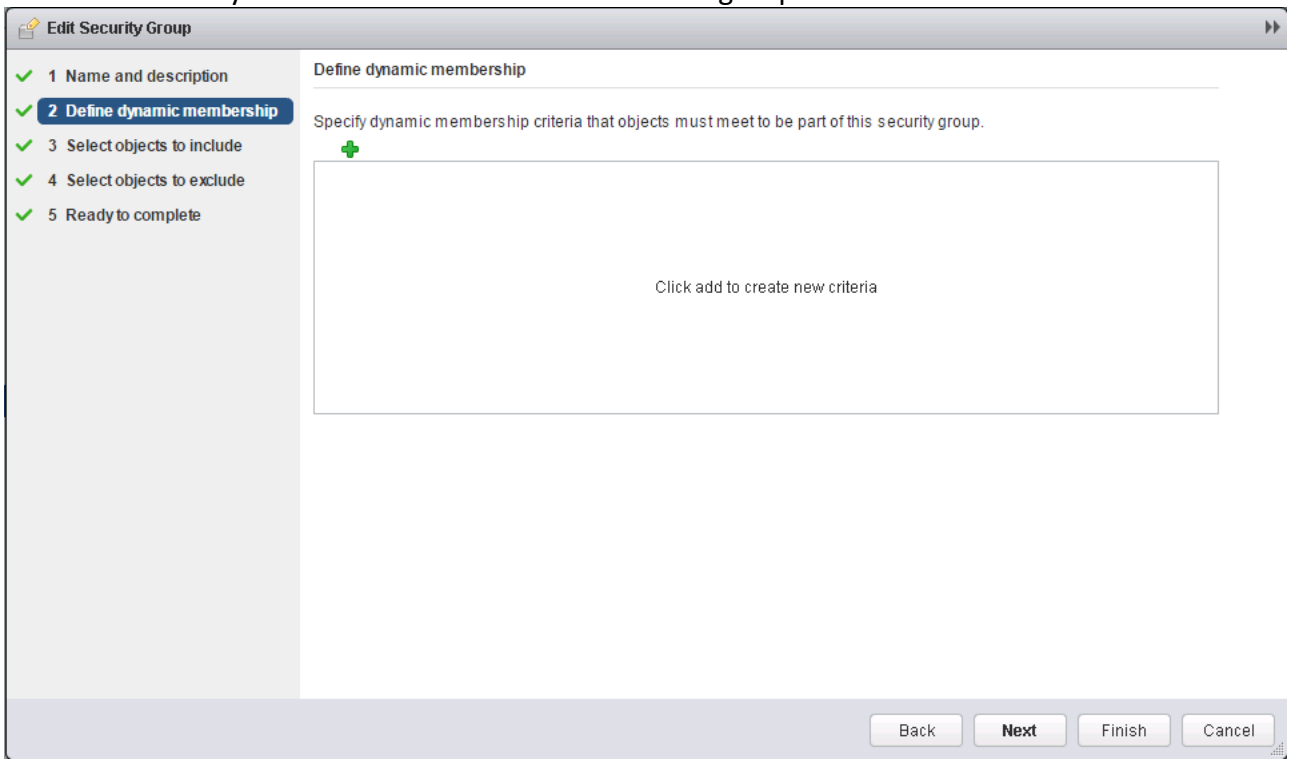
EntityTestGroup_002



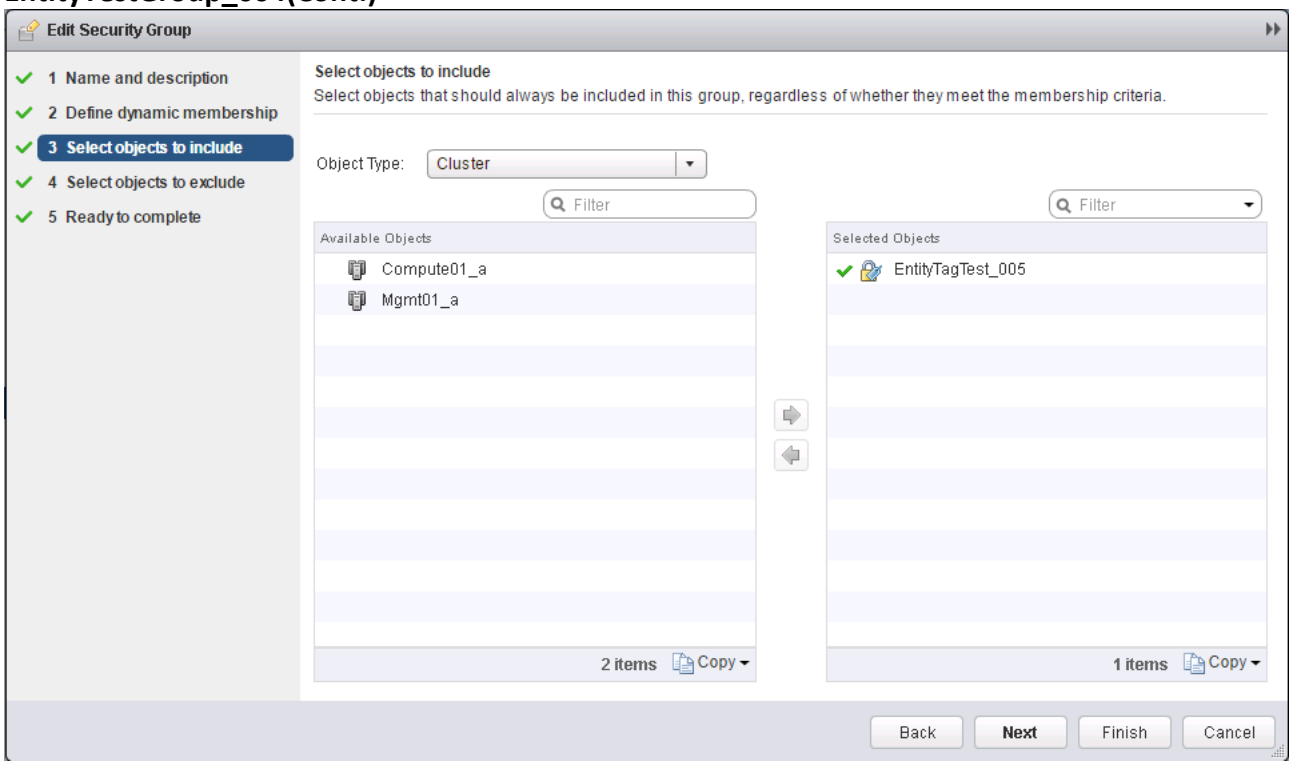
EntityTestGroup_003



EntityTestGroup_004 - The dynamic criteria configuration has been removed and the tag has been statically added as an include member of the group



EntityTestGroup_004(Cont.)



As you can see, converting from "Security Tag" & Equals to" over to "Entity" & "Belongs to" doesn't have to an exercise of repeated clicking in the UI.

One last thing, if you want to clean-up the test environment, just run the script again with the **-TrashTestEnvironment** parameter and it will clear up all the test objects it just created.

If you want to clean them up and re-create the test objects to run the script again, just supply both the **-CreateTestEnvironment** & **-TrashTestEnvironment** parameters when executing the script to start with the objects freshly re-created.

Go forth and automate with PowerNSX.