LSI / Dell / IBM iSCSI SAN Status Custom Service

About

The SAN Status custom service is a combination of a powershell script and service designed to monitor the performance metrics and health status of the virtual/logical disks as well as the controllers.

The script creates and uses a custom WMI class to act as the proxy repository of the information retrieved from Exchange.

The script is run

Configuration

The commands to access the information in the SAN differ between the Dell and IBM (and possibly the reference LSI) implementations. There are two blocks of variables in the script to set the commands and the health status response depending on whether the target is a Dell or IBM SAN. Both scripts feed the data into the same WMI class. Both Dell- and IBM-specific configurations are provided in this package.

There are two lines that need to be configured:

1. $smcli – This is the full path to the Storage Manager CLI executable.
2. $interval – This is the length of the performance metric monitoring interval, in seconds. This value is passed to the SMCLI as the length of time for which the CLI retrieves the performance metrics from the SAN.

The script takes one command line parameter, which is the *Name* of the SAN array from which to retrieve information. While this parameter is *not* case-sensitive, the WMI StorageArray instance parameter *is,* and the parameter to the Powershell script is passed to WMI unchanged. It's recommended to match the name in the parameter to the case used in naming the array so there will consistency throughout the information chain.

The Powershell script must be run on a device with the SAN Manager, and the WMI metrics are added to this device. This means that in N-Central, the SAN Status service is added to the management station, not to the SAN device itself. Considering that the SAN device may also have been added for connectivity and possibly for SNMP traps, this means the SAN information is being presented in two different places.

Verification

Place the powershell script in whatever script folder you use. Logon to a session as the account under which you plan to run the script. Open Powershell *in Elevated mode* and execute the script, passing the name of the SAN array as the parameter (ie, ". c:\scripts\dellsan.ps1 MD3220i_Bottom"). The script will write output to the console as it retrieves performance metrics from the SAN and as it retrieves the health status. You should receive no errors. The most common causes of errors are:

- Running the script in a regular Powershell session – authorization error
- Errors in the SAN name

With WBEMTEST you can confirm the parent class, the subclass, and the instances were created. Connect to root\cimv2, and enumerate the NCentral class. You will find the NCentral_SANStatus class. Open this and then click on the Instances button and you will find an instance for each controller and virtual/logical disk. The instance names are of the format "<SANName> – Virtual Disk <DiskName>". Note these instances names – *they are case sensitive* – to configure the instances of the custom class in N-Central.

Import the custom service into the NAC – it is titled SANStatus. Add the service to your management station and enter the instance name in the Storage Array field on the Service details tab.

Implementation

Upload the script to your script repository. Schedule the script to run on your array management station, passing the name of the SAN as a command line argument.