Making the Most of the Data ONTAP PowerShell Toolkit

Welcome

Thank you for using the Data ONTAP PowerShell Toolkit! We are committed to continuous improvement of our PowerShell solution. Please keep the great suggestions coming.

If you are new to the Toolkit, see our Getting Started slides. If you are considering embedding PowerShell in a larger .NET application, see our C# PowerShell embedding example or check out the Data ONTAP PowerShell Toolkit Proxy DLL.

Contents
Making the Most of the Data ONTAP PowerShell Toolkit .................................................................................................................. 1
Welcome ......................................................................................................................................................................................... 1
Data ONTAP PowerShell Toolkit 3.2.1 ........................................................................................................................................ 6
  Full Data ONTAP 8.3 API support ......................................................................................................................................... 6
  Other enhancements ................................................................................................................................................................. 6
  New cmdlets .............................................................................................................................................................................. 6
  Issues Fixed .............................................................................................................................................................................. 8
Data ONTAP PowerShell Toolkit 3.2 ........................................................................................................................................ 8
  Data ONTAP 8.3 API support .................................................................................................................................................. 8
  Other enhancements ................................................................................................................................................................. 9
  New Cmdlets .............................................................................................................................................................................. 9
  Issues Fixed .............................................................................................................................................................................. 10
Data ONTAP PowerShell Toolkit 3.1 ........................................................................................................................................ 10
  Perfstat data collection .......................................................................................................................................................... 10
  Full Data ONTAP 8.2.1 API support .................................................................................................................................. 11
  Other enhancements ................................................................................................................................................................. 12
  New Cmdlets .............................................................................................................................................................................. 16
  Issues Fixed .............................................................................................................................................................................. 16
Data ONTAP PowerShell Toolkit 3.0 ........................................................................................................................................ 17
  Sysstat for clustered Data ONTAP ...................................................................................................................................... 17
  Full Data ONTAP 8.2 API support ...................................................................................................................................... 18
<table>
<thead>
<tr>
<th>Data ONTAP PowerShell Toolkit 2.4</th>
<th>Data ONTAP PowerShell Toolkit 2.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Disk Space Reclamation</td>
<td>VSS Requester</td>
</tr>
<tr>
<td>VMDK to VHD/VHDX with IDE driver injection for Windows Server 2003</td>
<td>Installer</td>
</tr>
<tr>
<td>Data ONTAP 8.2 API Support</td>
<td>SCSI XCOPY</td>
</tr>
<tr>
<td>Other Enhancements</td>
<td>Private Key Authentication for SSH</td>
</tr>
<tr>
<td>New Cmdlets</td>
<td>Simplified Queries with PowerShell 3</td>
</tr>
<tr>
<td>Issues Fixed</td>
<td>Other Enhancements</td>
</tr>
<tr>
<td></td>
<td>Issues Fixed</td>
</tr>
<tr>
<td>Data ONTAP PowerShell Toolkit 2.2</td>
<td>Data ONTAP PowerShell Toolkit 2.1</td>
</tr>
<tr>
<td>VHDX Support</td>
<td>Stable output types</td>
</tr>
<tr>
<td>IPv6 Support</td>
<td></td>
</tr>
<tr>
<td>Storage Efficiency Calculations for Cluster-Mode</td>
<td></td>
</tr>
<tr>
<td>PowerShell Toolkit Proxy Layer</td>
<td></td>
</tr>
<tr>
<td>Other Enhancements</td>
<td></td>
</tr>
<tr>
<td>New Cmdlets</td>
<td></td>
</tr>
<tr>
<td>Issues Fixed</td>
<td></td>
</tr>
</tbody>
</table>

Other Enhancements ................................................................. 19
New Cmdlets .................................................................................. 20
Issues Fixed .................................................................................. 21
Data ONTAP PowerShell Toolkit 2.4 .......................................................... 21
Virtual Disk Space Reclamation ............................................................. 21
VMDK to VHD/VHDX with IDE driver injection for Windows Server 2003 .............................................. 22
Data ONTAP 8.2 API Support .................................................................... 22
Other Enhancements ............................................................................... 25
New Cmdlets ...................................................................................... 26
Issues Fixed ....................................................................................... 26
Data ONTAP PowerShell Toolkit 2.3 .......................................................... 26
VSS Requester ..................................................................................... 26
Installer ............................................................................................. 28
SCSI XCOPY .......................................................................................... 28
Private Key Authentication for SSH ........................................................ 29
Simplified Queries with PowerShell 3 ....................................................... 29
Other Enhancements ............................................................................... 30
Issues Fixed ....................................................................................... 31
Data ONTAP PowerShell Toolkit 2.2 .......................................................... 31
VHDX Support ..................................................................................... 31
IPv6 Support ...................................................................................... 32
Storage Efficiency Calculations for Cluster-Mode .......................................... 33
PowerShell Toolkit Proxy Layer ............................................................... 34
Other Enhancements ............................................................................... 35
New Cmdlets ...................................................................................... 36
Issues Fixed ....................................................................................... 36
Data ONTAP PowerShell Toolkit 2.1 .......................................................... 36
Stable output types ............................................................................. 36
Sub-LUN cloning a Windows file out of a snapshot ................................................................. 37
Complete Data ONTAP 8.1 coverage ......................................................................................... 37
Other enhancements ................................................................................................................ 37
New cmdlets ............................................................................................................................. 38
Issues fixed in Toolkit 2.1 .......................................................................................................... 39

Data ONTAP PowerShell Toolkit 2.0 ....................................................................................... 40
Data ONTAP 8.1 Cluster-Mode ................................................................................................. 40
Dynamic VHD file to thin, fixed VHD file conversion .............................................................. 40
LUN to VHD file conversion ..................................................................................................... 41
VHD file to LUN conversion ..................................................................................................... 41
NDMP copy client ..................................................................................................................... 41
More complete support for CIFS shares .................................................................................. 42
Installation scripts .................................................................................................................. 42
Other enhancements ................................................................................................................ 42
New cmdlets ............................................................................................................................. 43
Issues fixed in Toolkit 2.0 .......................................................................................................... 44

Data ONTAP PowerShell Toolkit 1.7 ....................................................................................... 45
V2V utilities ............................................................................................................................... 45
VHD file alignment ................................................................................................................... 45
Data ONTAP 8.1 (Cluster-Mode) ............................................................................................. 45
File cloning for CIFS ................................................................................................................ 47
Space reclamation for Cluster Shared Volumes ....................................................................... 47
Other enhancements ................................................................................................................ 47
New cmdlets ............................................................................................................................. 49
Issues fixed in Toolkit 1.7 .......................................................................................................... 49

Data ONTAP PowerShell Toolkit 1.6 ....................................................................................... 50
VHD file utilities ....................................................................................................................... 50
Space reclamation .................................................................................................................... 51
Windows disk and volume management ................................................................. 51
File cloning .................................................................................................................. 53
SSH setup .................................................................................................................... 53
Vfiler migration ......................................................................................................... 53
More complete API coverage .................................................................................. 53
Data ONTAP 8.1 (Cluster-Mode) ............................................................................ 53
Other enhancements ................................................................................................ 55
New cmdlets ............................................................................................................. 56
Issues fixed in Toolkit 1.5 ........................................................................................ 57
Data ONTAP PowerShell Toolkit 1.5 ....................................................................... 58
Storage efficiency calculations .................................................................................. 58
Host-side iSCSI & FC utilities .................................................................................. 59
ONTAP log parsing and monitoring ........................................................................ 61
Disk signature manipulation ..................................................................................... 62
Faster host disk & VM discovery ............................................................................... 62
Vfiler DR .................................................................................................................... 63
SnapLock .................................................................................................................... 63
Secure password handling ....................................................................................... 63
Other enhancements ................................................................................................ 63
New cmdlets ............................................................................................................. 64
Issues fixed in Toolkit 1.5 ........................................................................................ 64
Data ONTAP PowerShell Toolkit 1.4 ....................................................................... 65
Sysstat ....................................................................................................................... 65
Updated Data ONTAP support .................................................................................. 69
New cmdlet categories ............................................................................................. 69
Improved error handling .......................................................................................... 69
Simpler quota cmdlets .............................................................................................. 69
Improved snapshot cmdlets ...................................................................................... 70
Data ONTAP PowerShell Toolkit 1.2

- Data formatters
- Microsoft Hyper-V integration
- Network configuration
- New cmdlets
- Issues fixed in Toolkit 1.2

Data ONTAP PowerShell Toolkit 1.3

- SSH invocation
- Credentials cache
- FlexShare
- PowerShell Provider
- ShouldProcess
- ToString
- Sorting
- Connection enhancements
- Parameter aliases
- New cmdlets
- Issues fixed in Toolkit 1.3

Data ONTAP PowerShell Toolkit 1.4

- Table column header aliases
- Provider enhancements
- Hashed options
- Raw API invocation
- API list cmdlets
- Improved cluster support in Get-NaHyperV
- Issues fixed in Toolkit 1.4
Data ONTAP PowerShell Toolkit 3.2.1

Full Data ONTAP 8.3 API support
There are as many as 141 cmdlets introduced in Toolkit 3.2.1 and some of which are new categories as well. There are several updates to existing cmdlets too.

Other enhancements
- The following cmdlets have new parameters added:
  - **Add-NcLunMap**: Node
  - **New-NcLunMap**: ForeignDisk, ThinProvisioningSupportEnabled, Comment
  - **Set-NcAutosupportConfig**: MaxUploadRate, MaxDownloadRate, DisableOndemandDownload
  - **New-NcClone**: EnableAutodelete, TokenUuid
  - **Enable-NcClone**: Force
  - **Set-NcDiskOwner**: IsRootPartition, IsDataPartition, CopyOwnerFrom
  - **Clear-NcDiskOwner**: IsRootPartition, IsDataPartition
  - **Start-NcSystemImageUpdate**: IgnoreOffline
  - **Add-NcFileSystemSecurityPolicyTask**: AccessControl
  - **New-NcFileSystemAudit**: AuditEvents
  - **Set-NcFileSystemAudit**: AuditEvents
  - **New-NcFpolicyPolicy**: IsPassthroughReadEnabled
  - **Set-NcFpolicyPolicy**: IsPassthroughReadEnabled
  - **New-NcFpolicyExternalEngine**: RecvBufferSize, SendBufferSize, SessionTimeout
  - **Set-NcFpolicyExternalEngine**: RecvBufferSize, SendBufferSize, SessionTimeout
  - **Update-NcGpo**: ForceReapplyAllSettings
  - **Set-NcPerfPreset**: ExpirationLength
  - **Set-NcPerfArchiveConfig**: PerfstatSamplingPeriod
  - **Set-NcQosControlConfiguration**: PolicyDeactivation
  - **Remove-NcQosPolicyGroup**: Force
  - **New-NcQosPolicyReadAhead**: MinFileHistogram
  - **Set-NcQosPolicyReadAhead**: MinFileHistogram
  - **New-NcQosPolicyReadAhead**: MinFileHistogram
  - **Set-NcQosPolicyReadAhead**: MinFileHistogram
  - **New-NcSisPolicy**: PolicyType, ChangelogThresholdPercent
  - **Set-NcSisPolicy**: PolicyType, ChangelogThresholdPercent
  - **New-NcLdapClient**: IsNetgroupByHostEnabled, NetgroupByHostScope, AllowSsl
  - **Set-NcLdapClient**: IsNetgroupByHostEnabled, NetgroupByHostScope, AllowSsl, DisallowSsl
  - **Set-NaToolkitConfiguration**: LogToFile, NoConsoleOutput
  - **Set-NcLdapClientSchema**: EnableRfc2307bis, GroupOfUniqueNamesObjectClass, UniqueMemberAttribute, NisMapEntryAttribute, NisMapNameAttribute, NisObjectClass
  - **Restart-NcNode**: AllowInhibitTakeover, Reason, IsDump, AllowLifMigrationBeforeReboot, IgnoreQuorumWarnings
  - **Stop-NcNode**: AllowInhibitTakeover, Reason, IsDump, AllowLifMigrationBeforeShutdown, IgnoreQuorumWarnings

- You can now redirect ZAPI debugging output to a file, optionally with console print. See updates to **Set-NaToolkitConfiguration** above

New cmdlets

```powershell
PS C:\Users\rathnam\Documents\pstk> get-nccommand -category nameservice
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Family</th>
<th>Api</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Get-nCommand - Category netgroup

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Family</th>
<th>Api</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-NcNetgroupFile</td>
<td>netgroup</td>
<td>{cluster, vserver}</td>
<td>{netgroups-file-get-iter}</td>
</tr>
<tr>
<td>Remove-NcNetgroupFile</td>
<td>netgroup</td>
<td>{cluster, vserver}</td>
<td>{netgroups-file-delete}</td>
</tr>
</tbody>
</table>

### Get-nCommand - Category ntp_server

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Family</th>
<th>Api</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-NcNtpServer</td>
<td>ntp_server</td>
<td>{cluster}</td>
<td>{ntp-server-get-iter}</td>
</tr>
<tr>
<td>New-NcNtpServer</td>
<td>ntp_server</td>
<td>{cluster}</td>
<td>{ntp-server-create}</td>
</tr>
<tr>
<td>Remove-NcNtpServer</td>
<td>ntp_server</td>
<td>{cluster}</td>
<td>{ntp-server-delete}</td>
</tr>
<tr>
<td>Reset-NcNtpServer</td>
<td>ntp_server</td>
<td>{cluster}</td>
<td>{ntp-server-reset}</td>
</tr>
<tr>
<td>Test-NcNtpServer</td>
<td>ntp_server</td>
<td>{cluster}</td>
<td>{ntp-server-modify}</td>
</tr>
</tbody>
</table>

### Get-nCommand - Category tape_mc

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Family</th>
<th>Api</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear-NcTapeMcAlias</td>
<td>tape_mc</td>
<td>{cluster}</td>
<td>{tape-mc-info-alias-clear}</td>
</tr>
<tr>
<td>Disable-NcTapeMcTrace</td>
<td>tape_mc</td>
<td>{cluster}</td>
<td>{tape-mc-info-trace}</td>
</tr>
<tr>
<td>Enable-NcTapeMcTrace</td>
<td>tape_mc</td>
<td>{cluster}</td>
<td>{tape-mc-info-reset}</td>
</tr>
<tr>
<td>Get-NcTapeMc</td>
<td>tape_mc</td>
<td>{cluster}</td>
<td>{tape-mc-get-iter}</td>
</tr>
<tr>
<td>Reset-NcTapeMc</td>
<td>tape_mc</td>
<td>{cluster}</td>
<td>{tape-mc-info-reset}</td>
</tr>
<tr>
<td>Test-NcTapeMc</td>
<td>tape_mc</td>
<td>{cluster}</td>
<td>{tape-mc-info-test-reserve, tape-mc-info-test-release}</td>
</tr>
</tbody>
</table>

### Get-nCommand - Category *image

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Family</th>
<th>Api</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-NcClusterImage</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-get-iter}</td>
</tr>
<tr>
<td>Get-NcClusterImageDownload</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-get-download-progress}</td>
</tr>
<tr>
<td>Get-NcClusterImagePackage</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-package-local-get-iter}</td>
</tr>
<tr>
<td>Get-NcClusterImageUpdateHistory</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-update-history-get-iter}</td>
</tr>
<tr>
<td>Get-NcClusterImageUpdateLog</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-update-log-get-iter}</td>
</tr>
<tr>
<td>Get-NcClusterImageUpdateProgress</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-update-progress-get-iter}</td>
</tr>
<tr>
<td>Remove-NcClusterImagePackage</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-package-delete}</td>
</tr>
<tr>
<td>Resume-NcClusterImageUpdate</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-update-resume}</td>
</tr>
<tr>
<td>Start-NcClusterImagePackageDownload</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-package-download}</td>
</tr>
<tr>
<td>Start-NcClusterImageUpdate</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-update}</td>
</tr>
<tr>
<td>Stop-NcClusterImageUpdate</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-update-cancel}</td>
</tr>
<tr>
<td>Suspend-NcClusterImageUpdate</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-update-resume}</td>
</tr>
<tr>
<td>Test-NcClusterImage</td>
<td>cluster image</td>
<td>{cluster}</td>
<td>{cluster-image-validate}</td>
</tr>
</tbody>
</table>
PS C:\Users\rathnam\Documents\pstk> get-nccommand -category lock
Name                        Category        Family               Api
----                        --------         -------                ----
Get-NcLock                   lock             {cluster, vserver}  [lock-get-iter]
Remove-NcLock                lock             {cluster, vserver}  [lock-break-iter]

PS C:\Users\rathnam\Documents\pstk> get-nccommand -category coredump
Name                        Category        Family               Api
----                        --------         -------                ----
Invoke-NcCoredump            coredump        {cluster}            [coredump-trigger]

Issues Fixed
Not applicable.

Data ONTAP PowerShell Toolkit 3.2

Data ONTAP 8.3 API support
There are 126 new cmdlets in Toolkit 3.2, many of which are new for Data ONTAP 8.3. Many other cmdlets have been enhanced with new parameters (see Other enhancements section below for more details).

There are several categories introduced in Data ONTAP 8.3 and Toolkit 3.2 introduces two of them, namely MetroCluster and Storage Pool.

PS C:\Program Files (x86)\Netapp\Data ONTAP PowerShell Toolkit> get-nccommand -category Metrocluster
Name                        Category        Family               Api
----                        --------         -------                ----
Get-NcMetrocluster          metrocluster     {cluster}            [metrocluster-get]
Get-NcMetroclusterAggregateCheck metrocluster {cluster}            [metrocluster-check-aggregate-get-iter]
Get-NcMetroclusterAggregateEligibilityGetIter metrocluster {cluster}            [metrocluster-check-aggregate-elibility-get-iter]
Get-NcMetroclusterCheck     metrocluster     {cluster}            [metrocluster-check-get-iter]
Get-NcMetroclusterConfigDiff metrocluster     {cluster}            [metrocluster-config-diff-get-iter]
Get-NcMetroclusterConfigReplication metrocluster {cluster}            [metrocluster-config-replication-get]
Get-NcMetroclusterConfigReplicationGet metrocluster {cluster}            [metrocluster-check-config-replication-get]
Get-NcMetroclusterFailedLifPlacement metrocluster {cluster}            [metrocluster-show-lif-placement-failures-get-iter]
Get-NcMetroclusterInterconnect metrocluster {cluster}            [metrocluster-interconnect-get-iter]
Get-NcMetroclusterInterconnectAdapterGetIter metrocluster {cluster}            [metrocluster-interconnect-adaptor-get-iter]
Get-NcMetroclusterInterconnectAdapterSwitchGetIter metrocluster {cluster}            [metrocluster-interconnect-adaptor-switch-get-iter]
Get-NcMetroclusterNodeCheck metrocluster     {cluster}            [metrocluster-node-get-iter]
Get-NcMetroclusterNodeCheckNodeGetIter metrocluster {cluster}            [metrocluster-check-node-get-iter]
Get-NcMetroclusterOperation metrocluster     {cluster}            [metrocluster-operation-get-iter]
Get-NcMetroclusterProgress metrocluster     {cluster}            [metrocluster-progress-table-get-iter]
Get-NcMetroclusterVserver metrocluster     {cluster}            [metrocluster-vserver-get-iter]
Invoke-NcMetroclusterCheck metrocluster     {cluster}            [metrocluster-check-run]
Invoke-NcMetroclusterCheckCheck metrocluster {cluster}            [metrocluster-check-check]
Invoke-NcMetroclusterHeal metrocluster     {cluster}            [metrocluster-heal]
Invoke-NcMetroclusterSwitchback metrocluster {cluster}            [metrocluster-switchback]
Invoke-NcMetroclusterSwapover metrocluster {cluster}            [metrocluster-swapover]
Register-NcMetrocluster metrocluster     {cluster}            [metrocluster-configure]
Repair-NcMetroclusterLifPlacement metrocluster {cluster}            [metrocluster-check-lif-repair-placement]
Set-NcMetroclusterConfigReplication metrocluster {cluster}            [metrocluster-config-replication-modify]
Sync-NcMetroclusterVserver metrocluster     {cluster}            [metrocluster-vserver-resync]
Test-NcMetrocluster metrocluster     {cluster}            [metrocluster-is-configured]
Unregister-NcMetrocluster metrocluster {cluster}            [metrocluster-unconfigure]

PS C:\Program Files (x86)\Netapp\Data ONTAP PowerShell Toolkit> get-nccommand -category "Storage pool"
Name                        Category        Family               Api
----                        --------         -------                ----
Add-NcStoragePool           storage pool   {cluster}            [storage-pool-add]
Get-NcStoragePool           storage pool   {cluster}            [storage-pool-get-iter]
Get-NcStoragePoolAggr       storage pool   {cluster}            [storage-pool-aggregate-get-iter]
Get-NcStoragePoolDisk       storage pool   {cluster}            [storage-pool-disk-get-iter]
Other enhancements

Several cmdlets have been updated to include new parameters. The following cmdlets have new parameters in the clustered ONTAP cmdlet set:

- **Add-NcAggr**: CacheRaidGroupSize
- **New-NcAggr**: ExcludeFromAutobalance
- **Set-NcAggr**: Force (-Offline option)
- **Add-NcCifsServer**: Comment, NetbiosAlias
- **Add-NcCifsShare**: MaxConnectionsPerShare, ForceGroupForCreate, DisablePathValidation
- **Add-NcCifsShareAcl**: UserGroupType
- **Remove-NcCifsServer**: ForceAccountDelete
- **Remove-NcCifsServerAcl**: UserGroupType
- **Set-NcCifsOption**: IsCopyOffloadDirectCopyEnabled, IsUnixNtAclEnabled, ClientSessionTimeout, IsDacEnabled, AnonymousRestrictionLevel, IsReadOnlyDeleteEnabled, IsFakeOpenEnabled, IsUnixExtensionsEnabled, IsSearchShortNamesEnabled
- **Set-NcCifsSecurity**: IsAesEncryptionEnabled, LmCompatibilityLevel, AdminCredential
- **Set-NcCifsServer**: Comment, AddNetbiosAlias, RemoveNetbiosAlias
- **Set-NcCifsShare**: MaxConnectionsPerShare, ForceGroupForCreate, IsPathValidationEnabled
- **Set-NcCifsShareAcl**: UserGroupType
- **Add-NcClusterPeer**: AddressFamily, Passphrase, OfferExpiration
- **Set-NcClusterPeer**: AuthStatusAdmin, Passphrase, OfferExpiration
- **Get-NcNetSanLifPlacement**: Subnet, BroadcastDomain, Ipspace
- **New-NcNetInterface**: Subnet, Wwpn, ForceSubnetAssociation
- **Ping-NcHost**: WaitResponse
- **Set-NcNetInterface**: Subnet, Wwpn, ForceSubnetAssociation
- **Set-NcNetPort**: Ipspace, MtuAdmin
- **Invoke-NcSnapmirrorBreak**: RestoreDestinationSnapshot
- **Invoke-NcSnapmirrorCheck**: CheckFileListSource, CheckFileListDestination, SourceSnapshot, MaxTransferRate
- **Invoke-NcSnapmirrorRestore**: CreateCheckSnapshot, UseNetworkCompression, RestoreFileList
- **Invoke-NcSnapmirrorResync**: QuickResync
- **Invoke-NcSnapmirrorUpdate**: EnableStorageEfficiency
- **New-NcVol**: CachingPolicy, ExcludedFromAutobalance, Comment
- **New-NcVolCone**: CachingPolicy
- **New-NcVserver**: Ipspace, Subtype
- **Set-NcVserver**: VolumeDeleteRetentionHours
- **New-NcVserverPeerTransition**: LocalLif
- **Set-NcVserverPeerTransition**: LocalLif

New Cmdlets

There are 126 new cmdlets in Toolkit 3.2 making a grand total of 1603 cmdlets!

The following categories in the clustered ONTAP set have new cmdlets:

- **aggr** (16 cmdlets)
- **cifs** (13 cmdlets)
- **metrocluster** (26 cmdlets)
- **net** (48 cmdlets)
  - **ipspace** (4 cmdlets)
  - **firewall** (10 cmdlets)
- failover group (4 cmdlets)
- subnet (5 cmdlets)
- port broadcast domain (7 cmdlets)
- route (4 cmdlets)
- arp (5 cmdlets)
- ndp (9 cmdlets)
- storagepool (8 cmdlets)
- snapmirror (1 cmdlets)
- vserver (7 cmdlets)

**Issues Fixed**

- **ConvertTo-NaVmdk** could fail when connecting to a vserver management LIF that is also a CIFS data LIF and a network name was used.
- **ConvertTo-NaVmdk** could fail when connecting to a vserver by name when DNS round robin is used.
- **Set-NcTime** would fail for Data ONTAP 8.2 and higher.
- **Reset-NcCifsPassword** did not have a default parameter set.
- **Get-NaVol** could fail in the vfiler context.

**Data ONTAP PowerShell Toolkit 3.1**

**Perfstat data collection**

Perfstat is a tool used to collect performance data from a Data ONTAP controller. The data is collected in a format that can then be parsed and analyzed by end tools.

Toolkit 3.1 introduces the ability to collect basic Perfstat information with **Invoke-NaPerfstat** and **Invoke-NcPerfstat**. Only **FILER** data is collected—no **HOST** or **SWITCH** commands are run. All data collection is done via SSH, with no **SYSTEMSHELL** commands. More advanced Perfstat data collection can be done by the Perfstat tool: [http://support.netapp.com/NOW/download/tools/perfstat/](http://support.netapp.com/NOW/download/tools/perfstat/)

Both **Invoke-NaPerfstat** and **Invoke-NcPerfstat** generate output files that can be several MBs in size. The **OutputDirectory** parameter is used to specify which directory to place the output files. A new folder with a unique name will be generated in the **OutputDirectory**.

The following command collects perfstat data for the currently connected 7-mode controller:

```powershell
PS C:\Toolkit\3.1> Invoke-NaPerfstat -Directory C:\Perfstats -Time 60 -Iterations 2 -IterationSleepTime 30
```

The parallel commands are run for 60 seconds. Two iterations are performed with 30 seconds between iterations. The output files are located in the **C:\Perfstats** directory.

```powershell
PS C:\Toolkit\3.1> ls C:\Perfstats

Directory: C:\Perfstats

Mode                LastWriteTime       Length Name
---                --------------       ------ ----
d----            1/23/2014 3:28 PM   perfstat_20140123_152829
```

```powershell
PS C:\Toolkit\3.1> ls C:\Perfstats\perfstat_20140123_152829

Directory: C:\Perfstats\perfstat_20140123_152829
```
The following command collects perfstat information for one node in the current cluster. Two iterations are performed with 10 seconds between iterations.

```
PS C:\Toolkit\3.1> Invoke-NcPerfstat -Directory C:\Perfstats -Node sfp-cmode-02-01 -Iterations 2 -IterationSleepTime 10
```

The C:\Perfstats directory contains the output.

```
PS C:\Toolkit\3.1> ls C:\Perfstats
```

```
Directory: C:\Perfstats
```

```
Mode                LastWriteTime     Length  Name
----                -------------     ------  ----
-a---                1/23/2014 3:37 PM   20506925 output.data
-a---                1/23/2014 3:37 PM      49575 output.index
```

**Full Data ONTAP 8.2.1 API support**

There are 58 new cmdlets in Toolkit 3.1, many of which are new for Data ONTAP 8.2.1. Many other cmdlets have been enhanced with new parameters (see Other enhancements section below for more details).

There are several completely new categories introduced in Data ONTAP 8.2.1 and Toolkit 3.1.

```
PS C:\Toolkit\3.1> Get-NaCommand -Category ucm
```

```
Name                                Category        Api
----                                --------        ----
Get-NaUcmAdapter                    ucm             {ucm-adapter-list-info}
Set-NaUcmAdapter                    ucm             {ucm-adapter-modify}
```
Data ONTAP 8.2.1 introduces the ability to export qtrees. A new parameter on `New-NcQtree` and `Set-NcQtree`, `ExportPolicy`, allows an export policy (created using the cmdlets in the "exports" category) to be applied to a qtree.

```
PS C:\Toolkit\3.1> Get-NcCommand -Category "active directory"
Name                     Category        Family                  Api
------ ------------------ --------------- ----------------------
Get-NcActiveDirectoryAccount active direct... [cluster, vserver] {active-directory-account-get-iter}
New-NcActiveDirectoryAccount  active direct... [vserver] {active-directory-account-create}
Remove-NcActiveDirectoryAccount active direct... [vserver] {active-directory-account-delete}
Reset-NcActiveDirectoryAccount active direct... [vserver] {active-directory-account-password-reset}
Set-NcActiveDirectoryAccount active direct... [vserver] {active-directory-account-modify, active-di...}
```

```
PS C:\Toolkit\3.1> Get-NcCommand -Category ucm
Name                     Category        Family                  Api
------ ------------------ --------------- ----------------------
Get-NcUcmAdapter          ucm             [cluster] {ucm-adapter-get-iter}
Set-NcUcmAdapter          ucm             [cluster] {ucm-adapter-modify}
```

```
PS C:\Toolkit\3.1> Get-NcCommand -Category vscan
Name                     Category        Family                  Api
------ ------------------ --------------- ----------------------
Disable-NcVscan           vscan           [vserver] {vscan-status-modify}
Disable-NcVscanOnAccessPolicy vscan           [vserver] {vscan-on-access-policy-modify}
Enable-NcVscan            vscan           [vserver] {vscan-status-modify}
Enable-NcVscanOnAccessPolicy vscan           [vserver] {vscan-on-access-policy-modify}
Get-NcVscanConnection     vscan           [cluster, vserver] {vscan-connection-status-all-get-iter}
Get-NcVscanConnectionStats vscan           [cluster, vserver] {vscan-connection-extended-status-get-iter}
Get-NcVscanOnAccessPolicy vscan           [cluster, vserver] {vscan-on-access-policy-get-iter}
Get-NcVscanScannerPool    vscan           [cluster, vserver] {vscan-scanner-pool-get-iter}
Get-NcVscanScannerPoolActive vscan           [cluster, vserver] {vscan-active-scanner-pool-get-iter}
Get-NcVscanStatus         vscan           [cluster, vserver] {vscan-status-get-iter}
New-NcVscanOnAccessPolicy vscan           [vserver] {vscan-on-access-policy-create}
New-NcVscanScannerPool    vscan           [vserver] {vscan-scanner-pool-create}
Remove-NcVscanOnAccessPolicy vscan           [vserver] {vscan-on-access-policy-delete}
Remove-NcVscanScannerPool vscan           [vserver] {vscan-scanner-pool-delete}
Reset-NcVscan             vscan           [vserver] {vscan-reset}
Set-NcVscanOnAccessPolicy vscan           [vserver] {vscan-on-access-policy-modify}
Set-NcVscanScannerPool    vscan           [vserver] {vscan-scanner-pool-modify}
```

Other enhancements

**Get-NaHostDisk** is used to enumerate disks on the host system. Partner information has been added to the output of **Get-NaHostDisk** for 7-mode systems.

```
PS C:\Toolkit\3.1> Get-NaHostDisk 4 | fl *
```

```
Disk : 4
Size  : 2198888778240
```
ControllerPath          : fas2040rre1:/vol/vol1/biggerlun
ControllerMode          : Classic
HostDrivePath           : E:\
HostDiskName            :
HostVolume              :
HostVolumeIsCsv         : False
HostGptGuid             : 0d7d22034-f5d8-4d3d-a54b-1b41b63b7093
HostMbrSignature        :
HostDiskIndex           : 4
DiskSerialNumber        : P4DahZjB/4vx
DiskSize                : 219888778240
DiskWmiPath             :
ControllerName          : fas2040rre1
ControllerIgroup        : win2k8devrre0
ControllerIgroupType    : Windows
ControllerAddresses     : {192.168.0.8, 10.61.169.28}
ControllerLunPath       : /vol/vol1/biggerlun
ControllerVolumeName    : vol1
ControllerLunOsType     : windows_2008
ControllerSnapshot      :
ControllerQtree         : /vol/vol1
ControllerOntapRevision : 8.0.1 7-Mode
ControllerModel         : FAS2040
IScsiTarget             : 10.61.169.28:3260
PartnerControllerName   : fas2040rre2
IsHaEnabled             : True
IsHaIcConnected         : True

Two new cmdlets, Get-NcLunSignature and Set-NcLunSignature, work like their Na- counterparts for clustered ONTAP. Manipulating LUN signatures is especially useful when cloning LUNs. The following example script shows the usage of Set-NcLunSignature after cloning a LUN.

function Clone-Disk {
    param($DiskIndex)

    # Get the Disk to clone
    $disk = Get-NaHostDisk -Disk $DiskIndex

    # Make sure it's clustered ONTAP
    if(-not ($disk.ControllerMode -eq "Cluster")) { Write-Error "Not a clustered ONTAP LUN."; return }

    # Connect to the controller (Assuming credentials are in the cache)
    $controller = Connect-NcController $disk.ControllerAddresses[0] -Vserver $disk.ControllerName -Transient
    if($controller -eq $null) { Write-Error "Could not connect to controller."; return }

    # Clone the LUN
    $cloneLunPath = $disk.ControllerLunPath + ".$((Get-Date).ToString('yyyyMMdd_hhmmss'))"
    $destinationPath = $cloneLunPath.Substring("/vol/$($disk.ControllerVolumeName)".Length)
    if(-not $?) { Write-Error "Could not create LUN clone."; return }

    # Write a new signature
    Set-NcLunSignature -Path $cloneLunPath -Controller $controller | Out-Null
    if(-not $?) { Write-Error "Could not change LUN signature."; return }

    # Map the LUN
    Add-NcLunMap -Path $cloneLunPath -InitiatorGroup $disk.ControllerIgroup -Controller $controller | Out-Null
    if(-not $?) { Write-Error "Could not map the LUN."; return }

    # Wait for the LUN to come online
    Start-NaHostDiskRescan
    Wait-NaHostDisk -ControllerName $disk.ControllerName -ControllerLunPath $cloneLunPath
}

A Transient switch has been added to Start-NaNdmpCopy to prevent the NDMP copy operation status object being added to the global variable. The status can then be tracked using the status object emitted by Start-NaNdmpCopy.
Please note: when using the *Transient* switch, the NDMP copy status will *not* be returned by `Get-NaNdmpCopy`. The status *must* be captured from `Start-NaNdmpCopy`.

```powershell
PS C:\Toolkit\3.1> $ndmp_operation = Start-NaNdmpCopy -SrcController 10.61.167.213 -SrcPath /beam01/powershell/cifs/VHDX/redhat.vhdx -SrcCredential root -SrcAuthType md5 -DstController 10.61.167.86 -DstPath /vol/powershell/redhat -DstCredential root -DstAuthType md5 -Transient

PS C:\Toolkit\3.1> $ndmp_operation
Id   State          SrcPath                    DstPath                BackupBytesProcessed    BackupBytesRemain
--   ------          -------                    -------                ----------------------    ----------------------
1    RUNNING        /beam01/powershell/cifs... /vol/powershell/redhat/                    0                    0

PS C:\Toolkit\3.1> $ndmp_operation.Refresh()

PS C:\Toolkit\3.1> $ndmp_operation
Id   State          SrcPath                    DstPath                BackupBytesProcessed    BackupBytesRemain
--   ------          -------                    -------                ----------------------    ----------------------
1    RUNNING        /beam01/powershell/cifs... /vol/powershell/redhat/                    300 MB                25 GB

PS C:\Toolkit\3.1> $ndmp_operation.Refresh()

PS C:\Toolkit\3.1> $ndmp_operation
Id   State          SrcPath                    DstPath                BackupBytesProcessed    BackupBytesRemain
--   ------          -------                    -------                ----------------------    ----------------------
1    RUNNING        /beam01/powershell/cifs... /vol/powershell/redhat/                    827 MB                24 GB

PS C:\Toolkit\3.1> $ndmp_operation.LogMessages
[10.61.167.213] Log Message: LogType=0, MessageId=1, Message=DUMP: creating "/beam01/powershell/..../snapshot_for_backup.1" snapshot.
[10.61.167.213] Log Message: LogType=0, MessageId=2, Message=DUMP: Using Partial Volume Dump of selected subtrees
[10.61.167.213] Log Message: LogType=0, MessageId=3, Message=DUMP: Using snapshot_for_backup.1 snapshot
[10.61.167.213] Log Message: LogType=0, MessageId=5, Message=DUMP: Date of last level 0 dump: the epoch.
[10.61.167.213] Log Message: LogType=0, MessageId=6, Message=DUMP: Dumping /beam01/powershell/cifs/VHDX/redhat.vhdx to NDMP connection
[10.61.167.213] Log Message: LogType=0, MessageId=7, Message=DUMP: mapping (Pass I)[regular files]
[10.61.167.213] Log Message: LogType=0, MessageId=8, Message=DUMP: Reading file names from NDMP.
[10.61.167.213] Log Message: LogType=0, MessageId=9, Message=DUMP: mapping (Pass II)[directories]
[10.61.167.213] Log Message: LogType=0, MessageId=10, Message=DUMP: estimated 26269713 KB.
[10.61.167.213] Log Message: LogType=0, MessageId=11, Message=DUMP: dumping (Pass III) [directories]
[10.61.167.213] Log Message: LogType=0, MessageId=12, Message=DUMP: dumping (Pass IV) [regular files]
```

During an NDMP copy operation, the source and/or destination controllers will send log messages to the host. These log messages are now captured in the NDMP copy status object.
**Connect-NaController** and **Connect-NcController** will now prompt for credentials when no credentials are supplied and no credentials are present in the credentials cache.

**Set-NaToolkitConfiguration** now allows logging to be configured for four different loggers: `DataONTAP.PowerShell` for 7-mode cmdlet logging, `DataONTAP.C.PowerShell` for clustered ONTAP cmdlet logging, `DataONTAP.HostUtilities` for host utility logging, and `NetApp.Ontapi` for logging of Data ONTAP API calls. Changing the logging configuration no longer requires a restart of PowerShell.

```
PS C:\Toolkit\3.1> Set-NaToolkitConfiguration DEBUG NetApp.Ontapi
Log
---
```

If the Logger is not specified, all four loggers will be modified.

```
PS C:\Toolkit\3.1> Set-NaToolkitConfiguration DEBUG
Log
---
```

Several of the cmdlets in the 7-mode net category modify both the active and persistent configuration. Sometimes modifying the persistent configuration is not desirable. The 7-mode net cmdlets that modify the persistent configuration have a new `ActiveOnly` to indicate only the active configuration should be modified. The following cmdlets contain the new switch: `Add-NaNetRoute`, `Add-NaNetVlan`, `New-NaNetVif`, `Remove-NaNetRoute`, `Remove-NaNetVif`, `Remove-NaNetVlan`, `Set-NaNetInterface`, `Set-NaNetVif`, `Set-NaNetVlan`

The following example shows how the `ActiveOnly` switch can remove Vlans from the active configuration and keep them in the persistent configuration:

```
PS C:\Toolkit\3.1> Remove-NaNetVlan -Interface e0c -Vlans 40, 50 -ActiveOnly
PS C:\Toolkit\3.1> Get-NaNetActiveConfig

ConfigStatus : {}
Ifgrps       : {}
Interfaces   : {e0a, e0b, e0c, e0d...}
Routes       : {default}
Vlans        : {}
```

```
PS C:\Toolkit\3.1> Get-NaNetPersistentConfig

ConfigStatus : {}
Ifgrps       : {}
Interfaces   : {e0a, e0b, e0c, e0d...}
Routes       : {default}
Vlans        : {e0c-40, e0c-50}
```

Several cmdlets have been updated to include new parameters. The following cmdlets have new parameters in the 7-mode cmdlet set:

- **Add-NaNetRoute**: `ActiveOnly`
- **Add-NaNetVlan**: `ActiveOnly`
- **New-NaNetVif**: `ActiveOnly`
- **Remove-NaNetRoute**: `ActiveOnly`
- **Remove-NaNetVif**: `ActiveOnly`
- **Remove-NaNetVlan**: `ActiveOnly`
- **Restore-NaSnapshotFile**: `SpaceEfficientSplitDisabled`, `Force`
The following cmdlets have new parameters in the clustered ONTAP cmdlet set:

- **Add-NcCifsShare**: VscanProfile
- **Get-NcQosWorkload**: PolicyGroup
- **Invoke-NcSnapmirrorInitialize**: SourceSnapshot
- **New-NcFileserviceAudit**: Format
- **New-NcLdapClient**: UseStartTls
- **New-NcQtree**: ExportPolicy
- **Restore-NcSnapshotFile**: Force
- **Restore-NcSnapshotVolume**: Force
- **Set-NcCifsOption**: IsTrustedDomainEnumSearchEnabled, FileSystemSectorSize
- **Set-NcCifsSecurity**: UseStartTlsForAdLdap
- **Set-NcCifsShare**: VscanProfile
- **Set-NcFileserviceAudit**: Format
- **Set-NcLdapClient**: UseStartTls
- **Set-NcQtree**: ExportPolicy

### New Cmdlets

There are 58 new cmdlets in Toolkit 3.1, for a grand total of 1437 cmdlets!

The following categories in the 7-Mode set have new cmdlets:

- `cf` (2 cmdlets)
- `toolkit` (1 cmdlet)
- `ucm` (2 cmdlets)

The following categories in the clustered ONTAP set have new cmdlets:

- `active directory` (5 cmdlets)
- `cf` (6 cmdlets)
- `cifs` (6 cmdlets)
- `file` (1 cmdlet)
- `fileservice audit` (1 cmdlet)
- `lun` (4 cmdlets)
- `net` (6 cmdlets)
- `perf` (1 cmdlet)
- `security` (2 cmdlets)
- `snapshot` (1 cmdlet)
- `toolkit` (1 cmdlet)
- `ucm` (2 cmdlets)
- `vscan` (17 cmdlets)

### Issues Fixed

- Compliance clock info XML parsing for Data ONTAP 8.2.
- **Rename-NaQtree** did not work when the qtree contained spaces in the name.
- Set-NaTime with DateTime set UTC time erroneously.
- WriteBytesPerSecond value incorrect for LUN stats in Invoke-NcSysstat.

Data ONTAP PowerShell Toolkit 3.0

Sysstat for clustered Data ONTAP

Toolkit 3.0 introduces Invoke-NcSysstat, a cmdlet designed to report live performance data of the cluster. Like Invoke-NaSysstat, Invoke-NcSysstat monitors several performance counters to report on components of the system. The following performance statistics can be retrieved: System, FCP, NFSv3, NFSv4, CIFS, iSCSI, Volume, Ifnet, LUN, and Disk.

To use Invoke-NcSysstat, simply provide a switch indicating which type of performance statistics to retrieve. Use the Name parameter to monitor only specific performance objects. For example, to monitor volumes named “powershell” in the cluster:

```
PS C:\Toolkit\3.0> Invoke-NcSysstat -Volume -Name powershell -Count 5
```

<table>
<thead>
<tr>
<th>Name</th>
<th>RdOps</th>
<th>WrOps</th>
<th>TotOps</th>
<th>RdLat</th>
<th>WrLat</th>
<th>TotLat</th>
<th>Read</th>
<th>Written</th>
</tr>
</thead>
<tbody>
<tr>
<td>powershell</td>
<td>430</td>
<td>434</td>
<td>0.0</td>
<td>0.5</td>
<td>0.6</td>
<td>1 KB</td>
<td>27 MB</td>
<td></td>
</tr>
<tr>
<td>powershell</td>
<td>515</td>
<td>519</td>
<td>13.3</td>
<td>0.7</td>
<td>0.8</td>
<td>1 KB</td>
<td>32 MB</td>
<td></td>
</tr>
<tr>
<td>powershell</td>
<td>622</td>
<td>631</td>
<td>0.1</td>
<td>0.8</td>
<td>0.9</td>
<td>2 KB</td>
<td>39 MB</td>
<td></td>
</tr>
<tr>
<td>powershell</td>
<td>600</td>
<td>604</td>
<td>4.6</td>
<td>1.0</td>
<td>1.0</td>
<td>2 KB</td>
<td>37 MB</td>
<td></td>
</tr>
<tr>
<td>powershell</td>
<td>590</td>
<td>599</td>
<td>0.1</td>
<td>0.9</td>
<td>1.0</td>
<td>2 KB</td>
<td>37 MB</td>
<td></td>
</tr>
</tbody>
</table>

The Name parameter also accepts wildcards:

```
PS C:\Toolkit\3.0> Invoke-NcSysstat -Lun -Name /vol/powershell/*
```

<table>
<thead>
<tr>
<th>Read</th>
<th>Written</th>
<th>RdOps</th>
<th>WrOps</th>
<th>TotOps</th>
<th>TotLat</th>
<th>LunPath</th>
</tr>
</thead>
<tbody>
<tr>
<td>135 B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>/vol/powershell/disk1_gpt</td>
</tr>
<tr>
<td>0</td>
<td>128 B</td>
<td>0</td>
<td>128</td>
<td>128</td>
<td>217.0</td>
<td>/vol/powershell/luns/cluster</td>
</tr>
</tbody>
</table>

Invoke-NcSysstat works in both the cluster and Vserver context for Data ONTAP 8.2 and up. For Data ONTAP versions previous to 8.2, Invoke-NcSysstat must be run in the cluster context. The following performance statistics can be retrieved in the Vserver context: FCP, NFSv3, NFSv4, CIFS, iSCSI, Volume, and LUN.

When run in the cluster context, select monitored performance objects can be filtered by node or Vserver. Invoke-NcSysstat will then only monitor the performance objects associated with the given node or Vserver. For example, to monitor all of the volumes on a specific Vserver:

```
PS C:\Toolkit\3.0> Invoke-NcSysstat -Volume -Vserver beam01
```

<table>
<thead>
<tr>
<th>Name</th>
<th>RdOps</th>
<th>WrOps</th>
<th>TotOps</th>
<th>RdLat</th>
<th>WrLat</th>
<th>TotLat</th>
<th>Read</th>
<th>Written</th>
</tr>
</thead>
<tbody>
<tr>
<td>beam01_root_vol</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>clusterdisks</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>davidCModeIscsiMoun1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ndmp_destination</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>powershell</td>
<td>370</td>
<td>370</td>
<td>0.0</td>
<td>0.2</td>
<td>0.2</td>
<td>0</td>
<td>23 MB</td>
<td></td>
</tr>
<tr>
<td>testvol</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>v1NfsSrCMode</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vmstorage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The following example demonstrates how to monitor specific network ports on the node MFIT-01:

```
PS C:\Toolkit\3.0> Invoke-NcSysstat -Ifnet -Name e0* -Node MFIT-01
```
The following performance statistics can be filtered by Vserver: FCP, Volume, CIFS, iSCSI. The following performance statistics can be filtered by node: FCP, Ifnet, Disk, Volume, CIFS, iSCSI, System.

Additionally, Invoke-NcSysstat can aggregate the performance statistics for select objects by Vserver or node. Instead of displaying the performance results for each individual object, the performance statistics are aggregated for all the objects on the given Vserver or node. The following example aggregates the volume performance for all of the volumes in Vserver “beam01”:

```powershell
PS C:\Toolkit\3.0> Invoke-NcSysstat -Volume -Vserver beam01 -Aggregated -Count 5

Name               RdOps WrOps TotOps RdLat WrLat TotLat Read Written
beau01  0  257  266 0.0  0.2  0.2  0  16 MB
beau01  0  606  614 0.0  0.2  0.3  0  38 MB
beau01  0  357  363 0.0  0.3  0.3  0  22 MB
beau01  1  22  24 0.0  0.2  2.6 341 B 1 MB
beau01  0  1  8 0.0  0.1  0.1  0  2 KB
```

The following example aggregates the CIFS performance for all of the CIFS servers on the node “MFIT-01”:

```powershell
PS C:\Toolkit\3.0> Invoke-NcSysstat -Cifs -Node MFIT-01 -Aggregated -Count 5

Name               RdOps WrOps TotOps RdLat WrLat TotLat
MFIT-01  0  152  155 0.0  0.3  0.3
MFIT-01  0  153  156 0.0  0.3  0.3
MFIT-01  0  121  123 0.0  0.3  0.3
MFIT-01  0  154  158 0.0  0.3  0.3
MFIT-01  0  155  157 0.0  0.3  0.3
```

The CIFS, iSCSI, FCP, and Volume performance statistics can be aggregated by node or Vserver.

**Full Data ONTAP 8.2 API support**

Toolkit 3.0 completes the API support for Data ONTAP 8.2.

In addition to the 67 new cmdlets added for Toolkit 3.0, there are several cmdlets that have been updated with new parameters.

The following parameters have been added to the 7-mode cmdlet set:

- **ConvertTo-NaVhd**: FollowParent
- **ConvertTo-NaVhdx**: FollowParent
- **Read-NaDirectory**: Encode
- **Set-NaDiskOwner**: All, DiskCount, DiskType
- **Start-NaClone**: IgnoreStreams, IgnoreLocks

The following parameters have been added to the clustered ONTAP cmdlet set:

- **Add-NcSnmpCommunity**: VserverContext
- **Copy-NcLdapClientSchema**: VserverContext
- **Get-NcNfsService**: Query
- **Get-NcPerfInstance**: VserverContext
- **Invoke-NcClusterHaTakeover**: BypassOptimization, SkipLifMigration
- **New-NcClone**: IgnoreStreams, IgnoreLocks, QosPolicyGroup
- **New-NcFlexcachePolicy**: MetafilesTimeToLive, SymbolicTimeToLive, OtherTimeToLive, DelegationLruTimeout, PreferLocalCache, Vserver
- **New-NcLdapClient**: VserverContext
- **New-NcUser**: Comment
- **Read-NcFile**: Stream
- **Remove-NcFlexcachePolicy**: Vserver
- **Remove-NcLdapClient**: VserverContext
- **Remove-NcLdapClientSchema**: VserverContext
- **Remove-NcSnmpCommunity**: VserverContext
- **Set-NcAutoSupportConfig**: IsOndemandEnabled, IsOndemandRemoteDiagEnabled, OndemandServerUrl, OndemandPollingInterval, MinimumPrivateDataLength
- **Set-NcDiskOwner**: All, DiskCount, DiskType
- **Set-NcFlexcachePolicy**: MetafilesTimeToLive, SymbolicTimeToLive, OtherTimeToLive, DelegationLruTimeout, PreferLocalCache, Vserver
- **Set-NcLdapClient**: VserverContext
- **Set-NcLdapClientSchema**: VserverContext
- **Set-NcRoleConfig**: MinPasswordSpecialCharacter, PasswordExpirationDuration, MaxFailedLoginAttempts, LockoutDurationDays, RequireInitialPasswordUpdate
- **Set-NcUser**: Comment
- **Write-NcFile**: Stream

### Other Enhancements

**Get-NaVirtualDiskAlignment** and **Repair-NaVirtualDiskAlignment** are able to detect and repair alignment issues with VHD format virtual disks. Toolkit 3.0 extends the functionality of these two cmdlets to include VHDX format virtual disks.

```bash
PS C:\Toolkit\3.0> Get-NaVirtualDiskAlignment C:\ClusterStorage\Volume2\VirtualDisks\Win2k3_VM1\win2k3-boot.vhdx

    VirtualDisk: C:\ClusterStorage\Volume2\VirtualDisks\Win2k3_VM1\win2k3-boot.vhdx
                    IsBootable AbsoluteStartingLba Size IsExtendedBootRecord IsAligned
     -------------- ----------------- ---- ----------------- -------- --------
        True             63   20 GB             False      False

PS C:\Toolkit\3.0> Repair-NaVirtualDiskAlignment C:\ClusterStorage\Volume2\VirtualDisks\Win2k3_VM1\win2k3-boot.vhdx

Using host-based copy
A host-based copy will be used. This may take several minutes. Are you sure you wish to continue?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"):

Mode                LastWrite Time   Length Name
     ----                ----------- ------- -----
        -a---    7/11/2013  4:21 PM 21479030784 win2k3-boot.vhdx

Set-NaVirtualDiskSize enables the growing and shrinking of VHD format virtual disks. Shrinking a disk immediately reclaims the free space and growing a disk does not consume any additional space until data is written to the disk. Toolkit 3.0 extends the functionality of **Set-NaVirtualDiskSize** to include growing and shrinking VHDX format virtual disks. Please note: shrinking dynamic VHDX format virtual disks is not supported.

```bash
PS C:\Toolkit\3.0> Set-NaVirtualDiskSize C:\ClusterStorage\Volume2\VirtualDisks\Win2k3_VM1\win2k3-boot.vhdx +10GB

Grow VHD
```
Are you sure you want to grow the VHD?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"):

Mode                LastWriteTime      Length  Name
----                ---------------      ------  -----
-a-                7/12/2013 11:58 AM  32216449024  win2k3-boot.vhdx

The NDMP copy cmdlets (Start-NaNdmpCopy and Invoke-NaNdmpCopy) use the credentials cache to look up credentials to use to authenticate the NDMP session when no credentials are given in the cmdlet parameters. Before Toolkit 3.0, the credentials would only be looked up based on the IP addresses supplied in the cmdlet parameters. If the credentials are stored by hostname, the cmdlet would not find them and the NDMP session would not successfully authenticate. With Toolkit 3.0, the NDMP copy cmdlets will look up the credentials based on the IP addresses given and the hostnames associated with the IP addresses.

Prior to Toolkit 3.0, both SrcController and DstController were required parameters on Start-NaNdmpCopy and Invoke-NaNdmpCopy. In Toolkit 3.0, DstController is no longer required. If the DstController parameter is omitted, the cmdlet uses the SrcController as the destination.

PS C:\Toolkit\3.0> Invoke-NaNdmpCopy -SrcController 10.61.169.29 -SrcPath /vol/vol2/cifs/dynamic.vhdx -DstPath /vol/powershell/vol2dst -SrcAuthType md5 -DstAuthType md5

Copy-NaHostFile has been enhanced to offload copying to Data ONTAP when cloning cannot be used. This behavior is enabled for files in CIFS shares in Data ONTAP 7.3.5 and higher (excluding Data ONTAP 8.0), and for files in a LUN in Data ONTAP 8.1.1 and higher. Currently, only 7-Mode is supported.

The below table shows the performance increase for using Copy-NaHostFile to offload the copy operation of a 5GB data file to Data ONTAP when cloning is not possible:

<table>
<thead>
<tr>
<th></th>
<th>CIFS</th>
<th>LUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy-NaHostFile</td>
<td>29.744 seconds</td>
<td>37.454 seconds</td>
</tr>
<tr>
<td>Copy-Item</td>
<td>2 minutes 45 seconds</td>
<td>1 minute 48 seconds</td>
</tr>
</tbody>
</table>

Copy-NaHostFile has also been enhanced to accept a directory in the DestinationFile parameter. The destination file will be placed in the given directory with the same name as the source file.

Show-NaHelp now displays help for aliased cmdlets in the clustered ONTAP online help documentation.

New Cmdlets

There are 67 new cmdlets included in Toolkit 3.0, for a total of 1379 cmdlets!

The following clustered ONTAP categories contain new cmdlets:

- autosupport (1 cmdlets)
- cf (2 cmdlets)
- clone (2 cmdlets)
- diagnosis (1 cmdlets)
- file directory security (21 cmdlets)
- fileservice audit (6 cmdlets)
- ndmp (1 cmdlets)
• perf (8 cmdlets)
• sectrace (6 cmdlets)
• security certificate (8 cmdlets)
• security ssl (2 cmdlets)
• service processor (9 cmdlets)

Issues Fixed
• Timestamp output sometimes not returned correctly for Get-NcPerfData.
• Get-NaSnapmirrorConnection address pairs were missing for Data ONTAP 8.2.
• Controller connection serialized then deserialized with ConvertTo-SerializedString and ConvertFrom-SerializedString would not initialize ONTAPI version properly.
• Several of the "Related Links" in the online help documentation were broken.
• NDMP copy cmdlets (Invoke-NaNdmpCopy and Start-NaNdmpCopy) could hang if the source path does not exist.
• Performance of SCSI unmap (used for New-NaVirtualDisk, Remove-NaHostFile, Invoke-NaHostVolumeSpaceReclaim, etc) has been improved.
• Divide-by-zero error could occur in Get-NcEfficiency.

Data ONTAP PowerShell Toolkit 2.4

Virtual Disk Space Reclamation
When running virtual machines in a thin environment, the space savings of thin provisioning will decline over time. As files are added and removed from the virtual machine, the storage controller hosting the virtual disk may not be aware of the newly-freed blocks. These blocks will remain used while the virtual machine is running. This behavior results in long-running virtual machines consuming nearly all of the space allocated even when the virtual machine is not full.

Toolkit 2.4 introduces a new cmdlet, Invoke-NaNVirtualDiskSpaceReclaim, which enables the space reclamation of virtual disks (VHD and VHDX format, NTFS file system). With Windows Server 2012, it is possible to perform space reclamation on a running virtual machine by taking a Hyper-V snapshot of the running virtual machine, running Invoke-NaNVirtualDiskSpaceReclaim, and then removing the Hyper-V snapshot.

Consider the following example which reclaims space on a Windows Server 2008R2 virtual machine:

```
PS C:\Toolkit\2.4> Get-NcLun /vol/clusterdisks/disk1 | select Path, SizeUsed
Path                      SizeUsed
----                      --------
/vol/clusterdisks/disk1   83905994752

PS C:\Toolkit\2.4> Measure-Command -Expression {
>> Checkpoint-VM -Name Win2K8R2_VM1 -SnapshotName base
>> Invoke-NaNVirtualDiskSpaceReclaim -Name C:\ClusterStorage\Volume2\VirtualDisks\win2k8r2_vm1.vhdx -Confirm:$false
>> Remove-VMSnapshot -Name base -VMName Win2K8R2_VM1
>> }

Days                  : 0
Hours                 : 0
Minutes               : 0
Seconds               : 23
Milliseconds         : 974
Ticks                 : 239748180
TotalDays            : 0.000277486319444444
TotalHours           : 0.00665967166666667
TotalMinutes         : 0.3995803
TotalSeconds         : 23.974818
```
TotalMilliseconds : 23974.818

PS C:\Toolkit\2.4> Get-NcLun /vol/clusterdisks/disk1 | select Path, SizeUsed

<table>
<thead>
<tr>
<th>Path</th>
<th>SizeUsed</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vol/clusterdisks/disk1</td>
<td>78674378752</td>
</tr>
</tbody>
</table>

PS C:\Toolkit\2.4> (83905994752 - 78674378752)/1gb
4.87232208251953

We reclaimed just under 5GB of space in 24 seconds—all without having to take the VM offline!

**VMDK to VHD/VHDX with IDE driver injection for Windows Server 2003**

If you have ever tried to convert a SCSI VMDK containing Windows Server 2003 to VHD or VHDX, you are probably all too familiar with the "STOP: 0x0000007b" error. This occurs when booting the resulting VHD or VHDX after the conversion—requiring a system repair to remedy.

Toolkit 2.4 introduces the Win2K3ScsiToIde switch to ConvertTo-NaVhd and ConvertTo-NaVhdx. When this switch is present, the Toolkit will automatically install and configure the IDE drivers in a Windows Server 2003 virtual disk. This will allow the virtual disk to boot successfully in Hyper-V after the conversion.

PS C:\Toolkit\2.4> ConvertTo-NaVhdx -SourceVmdk C:\ClusterStorage\Volume2\VirtualDisks\VMDK\disk1.vmdk -DestinationVhdx C:\ClusterStorage\Volume2\VirtualDisks\VMDK\disk1.vhdx -Win2K3ScsiToIde

<table>
<thead>
<tr>
<th>Mode</th>
<th>LastWriteTime</th>
<th>Length</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a-</td>
<td>4/23/2013 10:27 AM</td>
<td></td>
<td>disk1.vhdx</td>
</tr>
</tbody>
</table>

**Data ONTAP 8.2 API Support**

Data ONTAP 8.2 includes a plethora of new and updated APIs. The Data ONTAP PowerShell Toolkit 2.4 includes API coverage for many of the new APIs.

The Data ONTAP licensing system has been updated for Data ONTAP 8.2. For clustered ONTAP, the new cmdlets in the 'license' category enable the management of licenses:

PS C:\Toolkit\2.4> Get-NcCommand -Category license

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Family</th>
<th>Api</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add-NcLicense</td>
<td>license</td>
<td>{cluster}</td>
<td>{license-v2-add}</td>
</tr>
<tr>
<td>Get-NcLicense</td>
<td>license</td>
<td>{cluster}</td>
<td>{license-v2-list-info}</td>
</tr>
<tr>
<td>Remove-NcLicense</td>
<td>license</td>
<td>{cluster}</td>
<td>{license-v2-delete, license-v2-delete-uns...}</td>
</tr>
</tbody>
</table>

For 7-Mode, the existing license cmdlets have been updated to work with the new licensing APIs. The output for Get-NaLicense looks slightly different for Data ONTAP 8.2:

PS C:\Toolkit\2.4> Get-NaLicense -Controller $DOT_8_2

<table>
<thead>
<tr>
<th>Service</th>
<th>Licensed Owner</th>
<th>Expiration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFS</td>
<td>True</td>
<td>stl2240-2-1</td>
</tr>
<tr>
<td>FlexClone</td>
<td>True</td>
<td>stl2240-2-1</td>
</tr>
<tr>
<td>iSCSI</td>
<td>True</td>
<td>stl2240-2-1</td>
</tr>
<tr>
<td>NFS</td>
<td>True</td>
<td>stl2240-2-1</td>
</tr>
<tr>
<td>SnapMirror</td>
<td>True</td>
<td>stl2240-2-1</td>
</tr>
<tr>
<td>SnapRestore</td>
<td>True</td>
<td>stl2240-2-1</td>
</tr>
<tr>
<td>SnapVault</td>
<td>True</td>
<td>stl2240-2-1</td>
</tr>
</tbody>
</table>

Notice all of the features listed are licensed. In order to see the status of all of the features, including those that are currently not licensed, use the new Get-NaFeatureStatus cmdlet:
In addition to the new cmdlets (see the “New Cmdlets” section below for more information on all the new cmdlets in Toolkit 2.4), many cmdlets have been updated to support Data ONTAP 8.2. In all, over 60 cmdlets have been updated for Toolkit 2.4!

The following parameters have been added to the 7-Mode cmdlet set:

- **Set-NaAggr**: Force
- **New-NaAggr**: SparePool, AllowMixedRpm, IgnorePoolChecks, PreCheck
- **Add-NaAggr**: AllowMixedRpm, IgnorePoolChecks, PreCheck, ForceCacheSize, CacheRaidType
- **New-NaAggrMirror**: AllowMixedRpm, IgnorePoolChecks, PreCheck
- **Set-NaSis**: EnableIdd, QuickCheckFileSize
- **Start-NaSis**: ScanAll, BuildMetadata, QosPolicy
- **Set-NaSnapmirrorSchedule**: IsCompressed, TcpWindowSize
- **Set-NaSnapmirrorSyncSchedule**: IsCompressed, TcpWindowSize
- **Set-NaVolAutosize**: MinimumSize, GrowThresholdPercent, ShrinkThresholdPercent, Reset, EnableGrow, EnableGrowShrink
- **New-NaVolClone**: VolumeType

The following parameters have been added to the clustered ONTAP cmdlet set:

- **New-NcAggr**: AllowMixedRpm, IgnorePoolChecks, PreCheck
- **Add-NcAggr**: AllowMixedRpm, IgnorePoolChecks, PreCheck, ForceCacheSize, CacheRaidType
- **Set-NcAggr**: Plex, Force
- **Remove-NcAggr**: Plex
- **Set-NcCifsSecurity**: IsSigningRequired, IsPasswordComplexityRequired
- **Set-NcCifsServer**: AdministrativeStatus, Domain, AdminUsername, AdminPassword, AdminCredential, ForceAccountOverwrite
- **Add-NcCifsServer**: AdministrativeStatus

---

<table>
<thead>
<tr>
<th>FeatureName</th>
<th>Notes</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDMI</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>CIFS</td>
<td></td>
<td>available</td>
</tr>
<tr>
<td>Compression</td>
<td>-</td>
<td>on</td>
</tr>
<tr>
<td>Dedupe</td>
<td>-</td>
<td>on</td>
</tr>
<tr>
<td>FCP</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>FlexCache</td>
<td>-</td>
<td>off</td>
</tr>
<tr>
<td>FlexClone</td>
<td>-</td>
<td>available</td>
</tr>
<tr>
<td>iSCSI</td>
<td>-</td>
<td>on</td>
</tr>
<tr>
<td>Multistore</td>
<td>-</td>
<td>on</td>
</tr>
<tr>
<td>NDMP</td>
<td>-</td>
<td>on</td>
</tr>
<tr>
<td>NFS</td>
<td>-</td>
<td>available</td>
</tr>
<tr>
<td>snapdrive_unix</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>snapdrive_windows</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>SnapLock</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>Snaplock_Enterprise</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>snapmanager_exchange</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>snapmanager_hyper</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>snapmanager_oracle</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>snapmanager_sap</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>snapmanager_sharepoint</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>snapmanager_sql</td>
<td>Feature not licensed</td>
<td>not_available</td>
</tr>
<tr>
<td>Snapmirror</td>
<td>-</td>
<td>on</td>
</tr>
<tr>
<td>Snaprestore</td>
<td>-</td>
<td>available</td>
</tr>
<tr>
<td>Snapvault_Primary</td>
<td>-</td>
<td>on</td>
</tr>
<tr>
<td>Snapvault_Secondary</td>
<td>-</td>
<td>on</td>
</tr>
<tr>
<td>Syncmirror</td>
<td>-</td>
<td>on</td>
</tr>
</tbody>
</table>
- Set-NcCifsShare: OfflineFilesMode
- Add-NcCifsShare: OfflineFilesMode
- Ping-NcHost: UseSourcePort
- Remove-NcLun: DestroyFencedLun
- New-NcLun: QosPolicyGroup, Class
- New-NcNetInterface: NetmaskLength, ListenForDnsQuery
- Set-NcNetInterface: NetmaskLength, ListenForDnsQuery
- Set-NcQosPolicyReadAhead: Name, UseHistogram
- Set-NcQosWorkload: Name, ReadAhead
- Remove-NcQosWorkload: Name
- Enable-NcSis: InfiniteVolume
- Set-NcSis: InfiniteVolume, EnableIdd, QuickCheckFileSize
- Start-NcSis: InfiniteVolume, ScanAll, BuildMetadata, QosPolicy
- Stop-NcSis: InfiniteVolume
- Disable-NcSis: InfiniteVolume
- New-NcSisPolicy: QosPolicy
- Update-NcSnapmirror: Policy
- Remove-NcSnapmirror: InfiniteVolume
- Get-NcSnapmirror: VserverContext
- New-NcSnapmirror: Policy
- Set-NcSnapmirror: Policy
- Invoke-NcSnapmirrorAbort: InfiniteVolume
- Invoke-NcSnapmirrorBreak: InfiniteVolume
- Invoke-NcSnapmirrorInitialize: TransferPriority
- Invoke-NcSnapmirrorQuiesce: Query, ContinueOnFailure, MaxFailureCount
- Invoke-NcSnapmirrorResume: Query, ContinueOnFailure, MaxFailureCount
- Invoke-NcSnapmirrorResync: SourceSnapshot, Preserve
- Invoke-NcSnapmirrorUpdate: SourceSnapshot, TransferPriority
- Remove-NcSnapshot: InfiniteVolume
- New-NcSnapshot: SnapmirrorLabel, InfiniteVolume
- New-NcSnapshotPolicy: SnapmirrorLabel
- Add-NcSnapshotPolicySchedule: SnapmirrorLabel
- Set-NcSnapshotPolicySchedule: SnapmirrorLabel
- Restore-NcSnapshotVolume: InfiniteVolume
- Update-NcVol: InfiniteVolume
- Set-NcVol: InfiniteVolume
- New-NcVol: QosPolicyGroup, ConstituentRole, StorageService, InfiniteVolume, NamespaceAggregate, MaxNamespaceConstituentSize, MaxDataConstituentSize, NamespaceMirrorAggregate, DataAggregate, EnableSnapdiff, Language
- Rename-NcVol: InfiniteVolume
- Remove-NcVol: InfiniteVolume
- Set-NcVolAutosize: Reset, Mode, MinimumSize, GrowThresholdPercent, ShrinkThresholdPercent
- New-NcVolClone: VolumeType, QosPolicyGroup
- Get-NcVolLimit: NamespaceAggregate, MaxNamespaceConstituentSize, MaxDataConstituentSize, DataAggregate, EnableSnapdiff, NamespaceMirrorAggregate
- Set-NcVolSize: InfiniteVolume
- Set-NcVserver: QosPolicyGroup, Language
Other Enhancements

Many of the host-side cmdlets use the ‘Na’ prefix even when they are able to operate on either a 7-Mode or clustered ONTAP controller. In order to prevent confusion, all of these cmdlets have been aliased with the ‘Nc’ prefix.

```
PS C:\Toolkit\2.4> Get-Alias | where { $_.ModuleName -eq 'DataONTAP' }
```

```
Capability    Name                                  ModuleName
------------- ------                             --------
Cmdlet        Add-NcHostIscsiConnection          DataONTAP
Cmdlet        Connect-NcHostIscsiTarget          DataONTAP
Cmdlet        ConvertTo-NcLun                  DataONTAP
Cmdlet        ConvertTo-NcVhd                   DataONTAP
Cmdlet        ConvertTo-NcVhdx                  DataONTAP
Cmdlet        Copy-NcHostFile                   DataONTAP
Cmdlet        Disconnect-NcHostIscsiTarget      DataONTAP
Cmdlet        Dismount-NcHostVolume            DataONTAP
Cmdlet        Dismount-NcVirtualDisk           DataONTAP
Cmdlet        Format-NcHostVolume              DataONTAP
Cmdlet        Get-NcHostDisk                    DataONTAP
Cmdlet        Get-NcHostFCApapter               DataONTAP
Cmdlet        Get-NcHostFCConnectivity          DataONTAP
Cmdlet        Get-NcHostIscsiAdapter            DataONTAP
Cmdlet        Get-NcHostIscsiSession            DataONTAP
Cmdlet        Get-NcHostIscsiTarget             DataONTAP
Cmdlet        Get-NcHostVolume                  DataONTAP
Cmdlet        Get-NcHyperV                      DataONTAP
Cmdlet        Get-NcVirtualDiskAlignment        DataONTAP
Cmdlet        Get-NcVssProvider                 DataONTAP
Cmdlet        Get-NcVssSnapshot                 DataONTAP
Cmdlet        Get-NcVssWriter                   DataONTAP
Cmdlet        Initialize-NcHostDisk             DataONTAP
Cmdlet        Invoke-NcHostVolumeSpaceReclaim   DataONTAP
Cmdlet        Invoke-NcVirtualDiskSpaceReclaim  DataONTAP
Cmdlet        Invoke-NcVmsBackup                DataONTAP
Cmdlet        Mount-NcHostVolume                DataONTAP
Cmdlet        Mount-NcVirtualDisk               DataONTAP
Cmdlet        New-NcHostVolume                  DataONTAP
Cmdlet        New-NcVirtualDisk                 DataONTAP
Cmdlet        Remove-NcHostFile                 DataONTAP
Cmdlet        Remove-NcHostIscsiConnection      DataONTAP
Cmdlet        Remove-NcHostVolume               DataONTAP
Cmdlet        Repair-NcVirtualDiskAlignment     DataONTAP
Cmdlet        Restore-NcHostFile                DataONTAP
Cmdlet        Set-NcHostDisk                    DataONTAP
Cmdlet        Set-NcHostVolumeSize              DataONTAP
Cmdlet        Set-NcVirtualDiskSize             DataONTAP
Cmdlet        Start-NcHostDiskRescan            DataONTAP
Cmdlet        Wait-NcHostDisk                   DataONTAP
```

Toolkit 2.4 supports the Update-Help cmdlet in PowerShell 3.0. Since the Toolkit module is a PowerShell 2.0 module, the HelpInfoURI property is not present by default. In order to use Update-Help, you must first add the HelpInfoURI property to the Toolkit module using `Add-NaHelpInfoUri`. Update-Help will be available the next time the Toolkit module is loaded.

```
PS C:\Toolkit\2.4> Get-Module DataONTAP | select HelpInfoUri
HelpInfoUri
-----------
```

```
PS C:\Toolkit\2.4> Add-NaHelpInfoUri
WARNING: HelpInfoURI will be available the next time the DataONTAP module is loaded.
PS C:\Toolkit\2.4> Remove-Module DataONTAP
PS C:\Toolkit\2.4> Import-Module DataONTAP
PS C:\Toolkit\2.4> Get-Module DataONTAP | select HelpInfoUri
HelpInfoUri
-----------
http://powershell.netapp.com/help/DataONTAP/
```
New Cmdlets
The following are new cmdlets, not including the Data ONTAP 8.2 API set:

- Invoke-NaVirtualDiskSpaceReclaim

The following 7-Mode categories contain new cmdlets:

- aggr (1 cmdlet)
- feature (1 cmdlet)
- volume (2 cmdlets)

The following clustered ONTAP categories contain new cmdlets:

- aggr (8 cmdlets)
- cifs (31 cmdlets)
- cluster (2 cmdlets)
- feature (1 cmdlet)
- fpolicy (22 cmdlets)
- group mapping (5 cmdlets)
- license (3 cmdlets)
- lun (3 cmdlets)
- net (6 cmdlets)
- qos (10 cmdlets)
- qtree (1 cmdlet)
- snapmirror (7 cmdlets)
- snapmirror policy (7 cmdlets)
- snapshot (4 cmdlets)
- system (1 cmdlet)
- volume (5 cmdlets)
- vserver (1 cmdlet)
- vserver peer (7 cmdlets)
- vserver peer transition (4 cmdlets)

Issues Fixed
- Get-NaCommand and Get-NcCommand were case-sensitive.
- Get-NaCifsShare would not fill out IsAccessBasedEnum value.
- Invoke-NaHostVolumeSpaceReclaim would emit "Target volume is not hosted by Data ONTAP" error on Windows Server 2003.

Data ONTAP PowerShell Toolkit 2.3

VSS Requester
The Microsoft Volume Shadow Copy Service (VSS) enables administrators to create consistent backups of volumes without degrading performance of running applications or services. This requires cooperation between the backup application (VSS requester), the host applications (VSS writers), and the storage system (VSS provider). Toolkit 2.3 provides basic VSS requester functionality which enables the creation of volume level shadow copies. It is recommended to have the Data ONTAP VSS Hardware Provider installed. This is available with SnapDrive for Windows.
Toolkit 2.3 introduces four new cmdlets used to interact with VSS: `Get-NAvssProvider`, `Get-NAvssSnapshot`, `Get-NAvssWriter`, and `Invoke-NAvssBackup`.

**Get-NAvssProvider** retrieves information about the VSS providers on the host system.

```
PS C:\Toolkit\2.3.0> Get-NAvssProvider

ProviderId : b5946137-7b9f-4925-af80-51abd60b20d5
ProviderName : Microsoft Software Shadow Copy provider 1.0
ProviderType : SYSTEM
ProviderVersion : 1.0.0.7
ProviderVersionId : 00000001-0000-0007-000000000001
ClassId : 65e1dbfa-8ff4-4a58-ac1c-3470eea2f376a

ProviderId : ddd3d232-a96f-4ac5-8f7b-250fd91fd102
ProviderName : Data ONTAP VSS Hardware Provider
ProviderType : HARDWARE
ProviderVersion : 6.4.1.5319
ProviderVersionId : 1a0989d8-b25b-439a-988a-a0f173f22046
ClassId : 819e04d3-809b-4657-ae00-778ca24470da
```

**Get-NAvssWriter** retrieves information about writers on the host system.

```
PS C:\Toolkit\2.3.0> Get-NAvssWriter

Name             : Task Scheduler Writer
WriterId         : d61d61c8-d73a-4eee-8cdd-f6f9786b7124
WriterInstanceId : 1bddd48e-5052-49db-9b07-b96f96727e6b
State            : STABLE
LastError        : 0

Name             : VSS Metadata Store Writer
WriterId         : 75dfb225-e2e4-4d39-9ac9-ffaff65ddf06
WriterInstanceId : 088e7a7d-09a8-4cc6-a609-ad90e75ddc93
State            : STABLE
LastError        : 0

Name             : Performance Counters Writer
WriterId         : 0bad1de-01a9-4625-8278-69e735f39dd2
WriterInstanceId : f008dda-9eefc-47c5-9eb6-a944c3d09381
State            : STABLE
LastError        : 0

Name             : Microsoft Hyper-V VSS Writer
WriterId         : 6641c0d4-6ded-4f4b-8f17-fd23f8ddc3de
WriterInstanceId : 2cd89efe-758b-4bc3-9af9-0c490e16adff
State            : STABLE
LastError        : 0

Name             : ASR Writer
WriterId         : be000cbe-11fe-4426-9c58-531a6355f04
WriterInstanceId : e2873497-f970-4e2f-b44c-11a1c3fd17ae
State            : STABLE
LastError        : 0

Name             : IIS Config Writer
WriterId         : 2a40fd15-dfca-4a8-a565-1f8c654603f6
WriterInstanceId : 68e9f7fe-12a3-47e6-b047-8223f31415c8
State            : STABLE
LastError        : 0

Name             : Shadow Copy Optimization Writer
WriterId         : 4dc3b74-ab48-4d07-adb0-3bee2926f7d7
WriterInstanceId : eae5c0c2-54e4-4f4a-b4a1-ff46cd14d666
State            : STABLE
LastError        : 0

Name             : Registry Writer
WriterId         : afbab4a2-367d-4d15-a586-71dab8f8485
WriterInstanceId : d22a492a-3cbb-40a8-b0bb-1e6e64dd61a2
State            : STABLE
LastError        : 0
```
Get-NaVssSnapshot gets information about the shadow copies present on the current host system.

Invoke-NaVssBackup offers a basic mechanism for interacting with VSS as a requester. Invoke-NaVssBackup operates on full volumes—it will not operate in component mode. The parameters IncludeWriter and ExcludeWriter provide a mechanism for explicitly selecting writers to participate in the backup processing.

Installer

Toolkit 2.3 is the first version of the Data ONTAP PowerShell Toolkit made available with a proper Windows installer. You can now install the Toolkit using the installer (no more manually copying files) and uninstall the Toolkit using Add/Remove Programs in Windows. Be sure to remove any previous versions of the Toolkit before running the installer.

SCSI XCOPY

Whenever possible, the PowerShell Toolkit uses WAFL block cloning to copy data from one file to another. When the WAFL blocks are not aligned, as is the case with Repair-NaVirtualDiskAlignment and ConvertTo-NaVhd for dynamic to fixed VHD conversion, data must be copied. In some cases this means utilizing a host-based copy, which can be painfully slow.

Toolkit 2.3 will attempt to use SCSI XCOPY for Repair-NaVirtualDiskAlignment and ConvertTo-NaVhd (for dynamic to fixed VHD conversion) once it has been determined a host-based copy must be used. If the SCSI XCOPY operation fails, the cmdlets will then fall back to using the slower host-based copy.

The following table compares running the two cmdlets using SCSI XCOPY and a host-based copy. The VHD disk0.vhd contains the boot drive of a Windows 2003 virtual machine that is misaligned. The disk is first converted to a fixed VHD using ConvertTo-NaVhd, and then aligned using Repair-NaVirtualDiskAlignment.
### Command

<table>
<thead>
<tr>
<th>Command</th>
<th>SCSI XCOPY</th>
<th>Host-Based Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConvertTo-Navhd -SourceVhd E:\VHD\Disk0.vhd – DestinationVhd E:\VHD\Disk0_fixed.vhd</code></td>
<td>19.188 seconds</td>
<td>1 minute, 10 seconds</td>
</tr>
<tr>
<td><code>Repair-NaVirtualDiskAlignment –VirtualDisk E:\VHD\Disk0_fixed.vhd</code></td>
<td>2 minutes, 25 seconds</td>
<td>12 minutes, 15 seconds</td>
</tr>
</tbody>
</table>

This functionality is available for Data ONTAP 8.1.1 7-Mode and will be available for clustered ONTAP in a future release.

### Private Key Authentication for SSH

Passwordless operation of SSH is important for authenticating SSH sessions without interactive password entry. The PowerShell Toolkit achieves passwordless operation using the credentials cache. Toolkit 2.3 introduces passwordless operation using private key authentication.

A new parameter on `Invoke-NaSsh` and `Invoke-NcSsh`, `PrivateKeyFile`, allows you to specify the private key file to use to authenticate the SSH connection.

```powershell
PS C:\Toolkit\2.3.0> Invoke-NaSsh -Name root@10.61.169.29 -Command "vol status" -PrivateKeyFile C:\ssh\privatekey
```

To further simplify private key authentication, the variable `$DataONTAP_SSHPrivateKeyFile` can be set instead of specifying the `PrivateKeyFile` parameter for each invocation of `Invoke-NaSsh` and `Invoke-NcSsh`. `Invoke-NaSsh` and `Invoke-NcSsh` will read this variable and use the private key file specified to authenticate the SSH connection.

```powershell
PS C:\Toolkit\2.3.0> $DataONTAP_SSHPrivateKeyFile = "C:\ssh\privatekey"
PS C:\Toolkit\2.3.0> Invoke-NaSsh -Name root@10.61.169.29 -Command "vol status"
Volume State Status Options
voll online raid_dp, flex sis
```

### Simplified Queries with PowerShell 3

PowerShell 3 introduces a new mechanism for instantiating and hydrating objects from hashtables. If an object defines an empty constructor and has public settable properties, a hashtable defining the value of the properties can be used to construct the object in PowerShell. For example, we can construct a VolumeAttributes object:

```powershell
PS C:\Toolkit\2.3.0> {DataONTAP.C.Types.Volume.VolumeAttributes}@{Name="test"}
```

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>TotalSize</th>
<th>Used</th>
<th>Available</th>
<th>Dedupe</th>
<th>Aggregate</th>
<th>Vserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>-----</td>
<td>----------</td>
<td>----</td>
<td>---------</td>
<td>-------</td>
<td>-----------</td>
<td>---------</td>
</tr>
</tbody>
</table>

```powershell
Name                      State       TotalSize  Used  Available Dedupe Aggregatename Vserver
----                      ----       ---------  ----  ---------   ---- -----------        -------
test
```
How does this affect the Toolkit? Consider this method for querying clustered ONTAP for volumes with a size greater than 100GB:

```
PS C:\Toolkit\2.3.0> $volume = Get-NcVol -Template
PS C:\Toolkit\2.3.0> Initialize-NcObjectProperty -Object $volume -Name VolumeSpaceAttributes
PS C:\Toolkit\2.3.0> $volume.VolumeSpaceAttributes.Size = ">$100GB"
PS C:\Toolkit\2.3.0> Get-NcVol -Query $volume
```

With PowerShell 3 and Toolkit 2.3, this can be condensed into a single line:

```
PS C:\Toolkit\2.3.0> Get-NcVol -Query @(TotalSize=">$100GB")
```

Or, you can look for something more specific still using a single line:

```
PS C:\Toolkit\2.3.0> Get-NcVol -Query @(TotalSize=">$100GB"; State=online; Aggregate="aggr1")
```

This functionality is enabled in Toolkit 2.3 with PowerShell 3.

**Other Enhancements**

*Get-NaHelp* and *Get-NcHelp* now include a cmdlet synopsis.

```
PS C:\Toolkit\2.3.0 > Get-NaHelp -Category aggr | ft Name, Category, Synopsis -AutoSize
```

```
Name                   Category Family    Synopsis
--------               -------- -----------
Add-NaAggr             aggr     {cluster} Add disks to the specified aggregate.
Confirm-NaAggrSpareLow aggr     {cluster} Return true if there is no suitable spare disk available for any filesystem (parity ...)
Get-NaAggr             aggr     Gets the aggregates for the current controller.
Get-NaAggrFilerInfo   aggr     Gets information on what possibilities and parameters exist for aggregates on a given ...
Get-NaAggrMediaScrub  aggr     Get the status of media scrubbing on the named aggregate, plex, or RAID group.
Get-NaAggrOption      aggr     Get the options that have been set for the specified aggregate.
Get-NaAggrScrub       aggr     Get the status of parity scrubbing on the named aggregate, plex, or RAID group.
Get-NaAggrSpace       aggr     Show the space usage of the aggregate on a per flexible volume basis.
Get-NaAggrVerify      aggr     Get the status of RAID mirror verification on the named aggregate.
New-NaAggr            aggr     Create a new aggregate.
New-NaAggrMirror      aggr     Turns an unmirrored aggregate into a mirrored aggregate by adding a plex to it.
Remove-NaAggr         aggr     Destroys the specified aggregate or plex.
Rename-NaAggr         aggr     Rename the specified aggregate.
Resume-NaAggrScrub    aggr     Resume parity scrubbing on the named aggregate, plex, or RAID group.
Resume-NaAggrVerify   aggr     Resume RAID mirror verification on the named aggregate.
Set-NaAggr            aggr     Set the specified aggregate or plex online, offline, or restricted.
Set-NaAggrOption      aggr     Set the specified option for the given aggregate.
Split-NaAggrMirror    aggr     Remove the specified plex from a mirrored aggregate and create a new unmirrored aggregate.
Start-NaAggrScrub     aggr     Start parity scrubbing on the named aggregate, plex, or RAID group.
Start-NaAggrVerify    aggr     Start RAID mirror verification on the named aggregate.
Stop-NaAggrScrub      aggr     Stop parity scrubbing on the named aggregate, plex, or RAID group.
Stop-NaAggrVerify     aggr     Stop RAID mirror verification on the named aggregate.
Suspend-NaAggrScrub   aggr     Suspend parity scrubbing on the named aggregate, plex, or RAID group.
Suspend-NaAggrVerify  aggr     Suspend RAID mirror verification on the named aggregate.
```

```
PS C:\Toolkit\2.3.0> Get-NcHelp -Category aggr | ft Name, Category, Family, Synopsis -AutoSize
```

```
Name                   Category Family    Synopsis
--------               -------- -----------
Add-NcAggr             aggr     {cluster} Add disks to the specified aggregate.
Confirm-NcAggrSpareLow aggr     {cluster} Return true if there is no suitable spare disk available for any filesystem ...
Get-NcAggr             aggr     {cluster} Gets the aggregates for the current controller.
Get-NcAggrNodeInfo    aggr     {cluster} Get information on what possibilities and parameters exist for aggregates ...
Get-NcAggrOption      aggr     {cluster} Get the options that have been set for the specified aggregate.
New-NcAggr            aggr     {cluster} Create a new aggregate with the given name.
Remove-NcAggr         aggr     {cluster} Destroys the specified aggregate.
Rename-NcAggr         aggr     {cluster} Rename the specified aggregate.
Set-NcAggr            aggr     {cluster} Set the specified aggregate online, offline, or restricted.
```
Two new cmdlets, **Get-NaCommand** and **Get-NcCommand**, are now available to help improve cmdlet discoverability. **Get-NaCommand** allows 7-Mode cmdlets to be searched by Name, Api, Category, Verb, and/or Noun, in any combination—including wildcards. **Get-NcCommand** allows clustered ONTAP cmdlets to be searched by Name, Api, Family, Category, Verb, and/or Noun, in any combination—including wildcards. Both **Get-NaCommand** and **Get-NcCommand** contain the switch ListAvailable which restricts the results to only cmdlets available in the current (or specified) controller context. The ListUnavailable switch is available to restrict the results to only cmdlets unavailable in the current (or specified) controller context.

For example, the following command gets all of the Get commands in the aggr and disk categories that are unsupported in the current controller context:

```
PS C:\Toolkit\2.3.0> Get-NaCommand -Verb Get -Category aggr, disk -ListUnavailable
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Family</th>
<th>Api</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-NcAggr</td>
<td>aggr</td>
<td>{cluster}</td>
<td>{aggr-get-iter}</td>
</tr>
<tr>
<td>Get-NcAggrNodeInfo</td>
<td>aggr</td>
<td>{cluster}</td>
<td>{aggr-get-filer-info}</td>
</tr>
<tr>
<td>Get-NcAggrOption</td>
<td>aggr</td>
<td>{cluster}</td>
<td>{aggr-options-list-info}</td>
</tr>
<tr>
<td>Get-NcDisk</td>
<td>disk</td>
<td>{cluster}</td>
<td>{storage-disk-get-iter}</td>
</tr>
<tr>
<td>Get-NcDiskOwner</td>
<td>disk</td>
<td>{cluster}</td>
<td>{disk-sanown-list-info}</td>
</tr>
<tr>
<td>Get-NcDiskOwnerFiler</td>
<td>disk</td>
<td>{cluster}</td>
<td>{disk-sanown-filer-list-info}</td>
</tr>
</tbody>
</table>

**Set-NaUserPassword** and **Set-NcUserPassword** will update the current controller connection credentials automatically if they are affected by the password change.

Cmdlets that use the vfiler-setup API (**Set-NaVfilerAddress**, **Set-NaVfilerAdminHost**, **Set-NaVfilerDns**, **Set-NaVfilerNis**, and **Set-NaVfilerPassword**) have been updated to include PreserveEtcHosts and PreserveEtcExports switches. The vfiler-setup API rewrites the /etc/hosts and /etc/exports files. Sometimes this behavior is not desirable. The PreserveEtcHosts and PreserveEtcExports switches will automatically restore the /etc/hosts and /etc/exports files on the vfiler to their pre-vfiler-setup state.

**Issues Fixed**

- Running **Set-NaVirtualDiskSize** with a UNC path did not fail elegantly.
- VHDX files created with **New-NaVirtualDisk** without the –Dynamic switch appeared as dynamic to Hyper-V.
- AccessTimeDT property of **Get-NaSnapshot** output was a UTC date with a Local DateTimeKind.
- **Remove-NaCredential** and **Remove-NcCredential** silently failed if the credential did not exist in the credential cache.
- **Get-NaSystemLog** may not properly parse dates from the previous year.

**Data ONTAP PowerShell Toolkit 2.2**

**VHDX Support**

The release of Windows Server 2012 brought with it the release of the VHDX file format. Toolkit 2.2 adds support for the VHDX file format by adding the ability to create VHDX files and convert to and from VHDX files. The cmdlet **New-NaVirtualDisk** is capable of rapidly creating fixed-size VHDX files that initially consume very little space.

| Create a 200GB fixed VHDX using New-Vhd | 33 minutes, 3 seconds |
| Create a 200GB fixed VHDX using New-NaVirtualDisk | 15 seconds |
The **New-NaVirtualDisk** cmdlet contains a new Switch, `-VHDX`, to enable the creation of VHDX virtual disks. The optional `-Dynamic` switch allows the creation of dynamic VHDX files, which now are aligned and will stay aligned as the file grows!

```
PS C:\Toolkit\2.2.0> New-NaVirtualDisk -FullName C:\ClusterStorage\Volume2\VirtualDisks\disk2.vhdx -Vhdx -Size 200GB
```

Mode LastWriteTime Length Name
---- ------------- ------ -----
-a-- 10/17/2012 10:04 AM 214752559104 disk2.vhdx

A new cmdlet, **ConvertTo-NaVhdx**, enables the conversion to VHDX virtual disks from VHDs, VMDKs, or LUNs.

```
PS C:\Toolkit\2.2.0> ConvertTo-NaVhdx -SourceVhd C:\ClusterStorage\Volume2\VirtualDisks\disk2.vhd
```

Mode LastWriteTime Length Name
---- ------------- ------ -----
-a-- 10/18/2012 9:17 AM 214752559104 disk2.vhdx

```
PS C:\Toolkit\2.2.0> ConvertTo-NaVhdx -SourceVmdk C:\ClusterStorage\Volume2\VirtualDisks\disk2.vmdk
```

Mode LastWriteTime Length Name
---- ------------- ------ -----
-a-- 10/18/2012 9:23 AM 53691285504 disk2.vhdx

```
PS C:\Toolkit\2.2.0> ConvertTo-NaVhdx -SourceLun /vol/clusterdisks/disk1 -DestinationVhdx C:\ClusterStorage\Volume2\VirtualDisks\disk1.vhdx
```

Mode LastWriteTime Length Name
---- ------------- ------ -----
-a-- 10/18/2012 9:26 AM 53724839936 disk1.vhdx

The **ConvertTo-NaVhd** cmdlet has been updated to convert VHDX to VHD.

```
PS C:\Toolkit\2.2.0> ConvertTo-NaVhd -SourceVhdx C:\ClusterStorage\Volume2\VirtualDisks\disk2.vhdx
```

Mode LastWriteTime Length Name
---- ------------- ------ -----
-a-- 10/17/2012 4:29 PM 214748365312 disk2.vhd

The **ConvertTo-NaVmdk** cmdlet has been updated to convert VHDX to VMDK.

```
PS C:\Toolkit\2.2.0> ConvertTo-NaVmdk -SourceVhdx C:\ClusterStorage\Volume2\VirtualDisks\disk2.vhdx
```

Mode LastWriteTime Length Name
---- ------------- ------ -----
-a-- 10/17/2012 4:12 PM 334 disk2.vmdk
-a-- 10/17/2012 4:12 PM 214748364800 disk2-flat.vmdk

The **ConvertTo-NaLun** cmdlet has been updated to convert VHDX to LUN.

```
PS C:\Toolkit\2.2.0> ConvertTo-NaLun -SourceVhdx C:\ClusterStorage\Volume2\VirtualDisks\disk1.vhdx -DestinationLun /vol/clusterdisks/disk1_vhdx
```

Path Size Protocol Online Mapped Thin Vserver
----- ------ ------- ---- ------ ----
/vol/clusterdisks/disk1_vhdx 50.0 GB windows_2008 True False True beam01

**IPv6 Support**

The toolkit is now ready for IPv6! **Connect-NaController** and **Connect-NcController** have been updated to enable connections to Data ONTAP controllers via IPv6.
The NDMP copy cmdlets, **Start-NaNdmpCopy** and **Invoke-NaNdmpCopy**, have been updated with a new **-UseIpv6** switch which will establish a connection between the source and destination controllers using IPv6. Both source and destination controllers must support IPv6.

```powershell
PS C:\Toolkit\2.2.0> Invoke-NaNdmpCopy -SrcController fe80::20c:29ff:fe9c:d223%26 -DstController fe80::20c:29ff:fe2d:4e78%26 -SrcPath /vol/vmstorage -DstPath /vol/vmstorage -SrcAuthType md5 -DstAuthType md5 -UseIpv6
```

The parameters **-Address**, **-StartAddress** and **-EndAddress** of **Find-NaController** and **Find-NcController** have been updated to accept IPv6 addresses.

```powershell
PS C:\Toolkit\2.2.0> Find-NaController -Address fe80::20c:29ff:fe9c:d223%26
```

**Storage Efficiency Calculations for Cluster-Mode**

A new cmdlet, **Get-NcEfficiency**, embeds the NetApp standard efficiency calculations for Cluster-Mode. The cmdlet will appear familiar to those using **Get-NaEfficiency**.

```powershell
PS C:\Toolkit\2.2.0> Get-NcEfficiency
```

A script posted on the Data ONTAP PowerShell communities takes the output from **Get-NcEfficiency** and exports the data to Microsoft Excel—including a few graphical representations of the data.
The release of Toolkit 2.1 contained stable output types which enabled users to safely work with strongly-typed code in projects that bundle the toolkit. With the release of Toolkit 2.2 we are introducing the Data ONTAP PowerShell Toolkit Proxy. The Proxy is a library that handles the interaction with PowerShell when interfacing with the PowerShell Toolkit in C# solutions. This allows you, the developer, to focus more on your solution and less on PowerShell.

Consider this example which starts a new PowerShell session, loads the Data ONTAP PowerShell module, connects to a controller, and then creates a new 50GB volume named 'testvol' on aggregate 'aggr1'.

```csharp
// open the PowerShell runspace
Runspace runspace = RunspaceFactory.CreateRunspace();
runtime.Open();

// import the Data ONTAP module
PowerShell s = PowerShell.Create();
s.AddCommand("Import-Module");
s.Runspace = runspace;
s.Invoke();

// connect to a controller
PowerShell connect = PowerShell.Create();
connect.Runspace = runspace;
connect.AddCommand("Connect-NaController");
connect.AddParameter("Name", "dunn");
connect.Invoke();
```
// invoke New-NaVol
PowerShell newVolume = PowerShell.Create();
newVolume.Runspace = this.runspace;
newVolume.AddCommand("New-NaVol");
newVolume.AddParameter("Name", "testvol");
newVolume.AddParameter("Aggregate", "aggr1");
newVolume.AddParameter("SpaceReserve", "none");
newVolume.AddParameter("Size", "50GB");
Collection<PSObject> results = newVolume.Invoke();

This is 20 lines of code to accomplish four fairly straightforward tasks. Now consider this example which accomplishes the same task using the Proxy.

// instantiate a Proxy object
Proxy ontapProxy = new Proxy();

// connect to the controller
ontapProxy.NaController.Connect("dunn");

// create the volume

What originally took 20 lines takes 3 when using the Proxy. See the post on the Data ONTAP PowerShell communities site for more information on how to get started.

Other Enhancements

Initialize-NaController performs the initial configuration of a Data ONTAP controller. Toolkit 2.2 provides an update to this cmdlet so that it is now compatible with Data ONTAP 8.1 7-Mode.

PS C:\Toolkit\2.2.0> Initialize-NaController 192.168.85.10 -Hostname rex -PrimaryInterface e0a -Gateway 255.255.255.0 -PrimaryInterfaceAddress 192.168.85.139 -Password p@ssword

Initialize controller
Are you sure you want to initialize controller 192.168.85.10/rex?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
WARNING: Nis is no longer supported in this cmdlet as of ONTAP 8.1, use Get-NaOption nis.* and Set-NaOption instead

Reboot Controller
Controller 192.168.85.139/rex must be rebooted before it is ready to use, would you like to do that now?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):

Two new cmdlets, Get-NaToolkitConfiguration and Set-NaToolkitConfiguration, allow you to easily disable and enable debug logging.

PS C:\Toolkit\2.2.0> Get-NaToolkitConfiguration

DebugLevel
--------
OFF

PS C:\Toolkit\2.2.0> Set-NaToolkitConfiguration -DebugLevel DEBUG
WARNING: New debug level will take effect after PowerShell is restarted.

DebugLevel
--------
DEBUG
New Cmdlets
Fourteen new cmdlets have been added to Toolkit 2.2, bringing the total number of cmdlets up to 1172!

These are new host-side cmdlets included in Toolkit 2.2:

- Add-NaHostIscsiConnection
- ConvertTo-NaVhdx
- Get-NaHostFcConnectivity
- Get-NaToolkitConfiguration
- Get-NcHostFcConnectivity
- Remove-NaHostIscsiConnection
- Set-NaToolkitConfiguration

These are new 7-Mode cmdlets included in Toolkit 2.2:

- Add-NalscsiInterfaceAccess
- Get-NaCifsBranchCacheHashStatistics
- Remove-NalscsiInterfaceAccess
- Set-NaCifsBranchCacheKey

These are new Cluster-Mode cmdlets included in Toolkit 2.2:

- Get-NcEfficiency
- Get-NcTime
- Set-NcTime

Issues Fixed

- **Restore-NaHostFile** would sometimes fail when a GPT disk came online as read-only.
- **Restore-NaHostFile** would fail if the temporary LUN’s disk volumes were not discovered immediately.
- **Get-NaVolAutosize** would return values 1024 times too large.
- **Set-NaQuota** would overwrite existing quota settings with default values if not specified.

Data ONTAP PowerShell Toolkit 2.1

Stable output types

Toolkit 2.1 at first glance appears to be a light release, but this is not the case. We’ve been hard at work making a major architectural change to the toolkit: stable output types.

In earlier versions of the toolkit, output types from cmdlets were subject to change, demonstrated by this change that occurred from Toolkit 1.6 to Toolkit 1.7:

```
PS C:\Toolkit\1.6.0> Get-NaLun | gm

    TypeName: NetApp.Ontapi.Filer.Lun73.LunInfo

PS C:\Toolkit\1.7.0> Get-NaLun | gm

    TypeName: NetApp.Ontapi.Filer.Lun810.LunInfo
```
As newer versions of APIs were released, it made sense to update the return types of the cmdlets. We were careful to
retain backwards compatibility when updating to a newer version, which resulted in this change occurring more or less
unnoticed by the average PowerShell user. As more developers show interest in bundling the toolkit in .Net projects,
this approach becomes less appealing. It is needlessly difficult (and somewhat dangerous) to write strongly-typed code
when these types are not guaranteed to stay the same. Toolkit 2.1 stabilizes these output types; they will no longer
change between releases:

```
PS C:\Toolkit\2.1.0> Get-NaLun | gm
TypeName: DataONTAP.Types.Lun.LunInfo
```

This change allows us to automatically update our return types whenever updated APIs are added to the toolkit ensuring
that you have the latest and greatest with each release of the toolkit. Users bundling the toolkit can safely work with
strongly-typed code without fear of a newer version of the toolkit changing established return types.

**Sub-LUN cloning a Windows file out of a snapshot**
The release of Data ONTAP 8.1 introduced the ability to SIS-clone from a snapshot. A new cmdlet, `Restore-NaHostFile`,
leverages this feature to allow a file to be copied from a LUN in a snapshot to the LUN in the active filesystem. With this
cmdlet, you can restore individual Windows files from Data ONTAP snapshots. If regular snapshots are scheduled, file
restores are possible from these incremental backups. The cmdlet works in both 7-Mode and Cluster-Mode, and
requires Data ONTAP 8.1 or later.

```
PS C:\Toolkit\2.1.0> Restore-NaHostFile K:\virtual.vhd -Snapshot weekly.2012-07-15_0015
```

<table>
<thead>
<tr>
<th>Mode</th>
<th>LastWriteTime</th>
<th>Length</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>-------------</td>
<td>------</td>
<td>-----------------</td>
</tr>
<tr>
<td>----</td>
<td>7/25/2012</td>
<td>11:09 AM</td>
<td>327547840 virtual.vhd</td>
</tr>
</tbody>
</table>

**Complete Data ONTAP 8.1 coverage**
With Toolkit 2.0, we froze our Cluster-Mode architecture with 84% API coverage for Data ONTAP 8.1. Now, with the
release of Toolkit 2.1, we are releasing full Data ONTAP 8.1 API coverage. There are 559 Cluster-Mode cmdlets in Toolkit
2.1 and 1158 total cmdlets!

**Other enhancements**

Find-NcController will only display Cluster LIFs unless the –All switch is present. This enables easier identification of
Data ONTAP clusters in the discovery process.

```
PS C:\Toolkit\2.1.0> Find-NcController 10.60.189.0 -Mask 255.255.255.0
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vxterp</td>
<td>10.60.189.26</td>
<td></td>
<td>NetApp Release RollingRock__8.1.1 Cluster-Mode: Tue O</td>
</tr>
<tr>
<td>dot-vsac12</td>
<td>10.60.189.130</td>
<td></td>
<td>NetApp Release RollingRock__8.1.0 Cluster-Mode: Wed M</td>
</tr>
<tr>
<td>vvol-vsim</td>
<td>10.60.189.157</td>
<td></td>
<td>NetApp Release RollingRock__8.1.0 Cluster-Mode: Thu J</td>
</tr>
</tbody>
</table>

With the –All switch, all the SNMP devices on the given subnet are returned.

```
PS C:\Toolkit\2.1.0> Find-NcController 10.61.172.0 -Mask 255.255.255.0 -All
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>brocade200e-11</td>
<td>10.61.172.11</td>
<td></td>
<td>Fibre Channel Switch.</td>
</tr>
<tr>
<td>spec01node01</td>
<td>10.61.172.12</td>
<td></td>
<td>NetApp Release RollingRock__8.1.0 Cluster-Mode: Mon M</td>
</tr>
<tr>
<td>spec01node02</td>
<td>10.61.172.14</td>
<td></td>
<td>NetApp Release RollingRock__8.1.0 Cluster-Mode: Mon M</td>
</tr>
<tr>
<td>brocade200e-19</td>
<td>10.61.172.19</td>
<td></td>
<td>Fibre Channel Switch.</td>
</tr>
<tr>
<td>refex-2050-4</td>
<td>10.61.172.26</td>
<td>Noncritical</td>
<td>NetApp Release 7.3.4: Thu May 27 17:52:48 PD</td>
</tr>
<tr>
<td>TESCO-17</td>
<td>10.61.172.83</td>
<td></td>
<td>NetApp Release 8.1.1RC1 Cluster-Mode: Fri Ju</td>
</tr>
</tbody>
</table>
The NDMP copy cmdlets now support incremental levels 0-9 and “Incremental Forever” using the switch – IncrementalForever.

PS C:\Toolkit\2.1.0> Invoke-NaNdmpCopy -SrcController 10.61.167.60 -SrcPath /vol/source/ -DstController 10.61.169.29 -DstPath /vol/destination/ -SrcAuthType md5 -DstAuthType md5 -Level 0

After an initial Level 0 transfer, the –IncrementalForever switch can be used.

PS C:\Toolkit\2.1.0> Invoke-NaNdmpCopy -SrcController 10.61.167.60 -SrcPath /vol/source/ -DstController 10.61.169.29 -DstPath /vol/destination/ -SrcAuthType md5 -DstAuthType md5 -IncrementalForever

Set-NaUser now supports setting the password minimum and maximum age.

PS C:\\ Toolkit\2.1.0> Set-NaUser steven -PasswordMinAge 0 -PasswordMaxAge 100

New-NcVserver contains the ability to create repository Vservers with the new –RepositoryVserver switch (new to Data ONTAP 8.1.1).

PS C:\Toolkit\2.1.0> New-NcVserver -Name repo -RootVolume repo_root -NameServerSwitch file -RootVolumeSecurityStyle unix -RootVolumeAggregate aggr1 -RepositoryVserver

New cmdlets
These are new cmdlets in Data ONTAP PowerShell Toolkit 2.1, not including the Cluster-Mode set:

- Restore-NaHostFile
These are categories with new Cluster-Mode cmdlets in Toolkit 2.1:

- Antivirus (11 cmdlets)
- Config backup (11 cmdlets)
- Dashboard (3 cmdlets)
- Diagnosis (11 cmdlets)
- Fcport (1 cmdlet)
- Flash (2 cmdlets)
- Flexcache (9 cmdlets)
- Gpo (6 cmdlets)
- Iscsi (7 cmdlets)
- Kerberos (5 cmdlets)
- Name mapping (5 cmdlets)
- Nis (4 cmdlets)
- Storage initiator (7 cmdlets)
- Volume (3 cmdlets)

**Issues fixed in Toolkit 2.1**

- `Get-NaSnapvaultSecStatus` | `Start-NaSnapvaultSecTransfer` would not bind parameters correctly.
- `Invoke-NcClusterHaTakeover` could fail when "immediate" is not possible even when `-Immediate` is not present.
- `Get-NaSystemLog -AuditLog` would sometimes fail to set the TimeStampDT field.
- Minor documentation tweaks.
Data ONTAP PowerShell Toolkit 2.0

Data ONTAP 8.1 Cluster-Mode

We labeled this release “2.0” not because of a generational change, but because we’ve frozen the architecture for the Toolkit’s Cluster-Mode support. This release adds 86 cmdlets for a Cluster-Mode total of 461, representing 84% API coverage for Data ONTAP 8.1 Cluster-Mode.

We’ve provided plenty of examples so far about how to use the Toolkit with Cluster-Mode. Here’s an example of a serious force multiplier courtesy of the Toolkit’s support for multiple simultaneous connections to Cluster-Mode controllers. In just four lines of PowerShell, we connect to four data Vservers, create a volume and LUN on each Vserver, and map the new LUNs to the igroups known to each Vserver. Note the –Add switch on Connect-NcController:

```powershell
PS C:\Toolkit\2.0.0> "Sea3210-SP", "Sea3210-XD", "Sea3210-SME", "Sea3210-SQL" | %{Connect-NcController Sea3240 -Vserver $_ -Add}
```

---

### Name | Address | Vserver | Version
--- | ---- | ---- | ----
Sea3240 | 10.58.96.30 | Sea3210-SP | NetApp Release RollingRock_8.1.0 Cluster-Mode: Thu Nov ...
Sea3240 | 10.58.96.30 | Sea3210-XD | NetApp Release RollingRock_8.1.0 Cluster-Mode: Thu Nov ...
Sea3240 | 10.58.96.30 | Sea3210-SME | NetApp Release RollingRock_8.1.0 Cluster-Mode: Thu Nov ...
Sea3240 | 10.58.96.30 | Sea3210-SQL | NetApp Release RollingRock_8.1.0 Cluster-Mode: Thu Nov ...

---

### Name | State | TotalSize | Used | Available | Dedupe | Aggregate | Vserver
--- | ---- | -------- | ---- | -------- | ------ | --------- | ----
vmsrorage | online | 10.0 TB | 36% | 6.3 TB | False | aggr1_Sea3240_01 | Sea3210-SP
vmsrorage | online | 10.0 TB | 37% | 6.3 TB | False | aggr1_Sea3240_01 | Sea3210-XD
vmsrorage | online | 10.0 TB | 37% | 6.2 TB | False | aggr1_Sea3240_01 | Sea3210-SME
vmsrorage | online | 10.0 TB | 38% | 6.2 TB | False | aggr1_Sea3240_01 | Sea3210-SQL

---

### Path | Size | Protocol | Online Mapped | Thin | Vserver | Comment
--- | ---- | -------- | -------- | ---- | ---- | ----
/vol/vmsrorage/csv1 | 10.0 TB | hyper_v | True | False | Sea3210-SP | 
/vol/vmsrorage/csv1 | 10.0 TB | hyper_v | True | False | Sea3210-XD | 
/vol/vmsrorage/csv1 | 10.0 TB | hyper_v | True | False | Sea3210-SME | 
/vol/vmsrorage/csv1 | 10.0 TB | hyper_v | True | False | Sea3210-SQL |

---

Dynamic VHD file to thin, fixed VHD file conversion

The benefits of running Hyper-V VMs on dynamic VHD files are obvious: quick creation time and minimal storage consumption. But a side effect of the dynamic VHD file format is that most I/O to a dynamic VHD file on a SAN is guaranteed to be misaligned regardless of how the VHD is formatted with data partitions. That isn’t a big deal with a couple of VMs, but with dozens of VMs running on a controller, performance can be significantly impaired.
An earlier Toolkit release introduced **New-NaVirtualDisk**, which creates thin, fixed VHD files without the wait or the up-front storage spend. But for everyone who already has dynamic VHD files on Data ONTAP, Toolkit 2.0 updates **ConvertTo-NaVhd** to convert dynamic VHDs to thin, fixed ones. The conversion runs as quickly as possible using a combination of copy offload and block cloning.

```powershell
PS C:\Toolkit\2.0.0> ConvertTo-NaVhd -SourceVhd M:\Dynamic.vhd -DestinationVhd M:\Fixed.vhd
```

<table>
<thead>
<tr>
<th>Mode</th>
<th>LastWriteTime</th>
<th>Length</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a---</td>
<td>4/19/2012 2:25 PM</td>
<td>536870912512</td>
<td>Fixed.vhd</td>
</tr>
</tbody>
</table>

**LUN to VHD file conversion**

A physical-to-virtual migration typically requires using a tool that copies VM data from a LUN to a VHD file. In Toolkit 2.0, **ConvertTo-NaVhd** can convert an entire LUN to a VHD file in another LUN. The cmdlet accepts a LUN path or a host disk index of a disk backed by a Data ONTAP LUN.

```powershell
PS C:\Toolkit\2.0.0> ConvertTo-NaVhd -SourceDiskIndex 1 -DestinationVhd E:\hostdisk1.vhd
```

<table>
<thead>
<tr>
<th>Mode</th>
<th>LastWriteTime</th>
<th>Length</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a---</td>
<td>4/30/2012 4:13 PM</td>
<td>53694628352</td>
<td>hostdisk1.vhd</td>
</tr>
</tbody>
</table>

**VHD file to LUN conversion**

A virtual-to-physical migration may be less common, but there are times it makes sense, such as moving an overloaded VM to a physical server on a temporary basis. Toolkit 2.0 adds **ConvertTo-NaLun** to rapidly create a LUN from a VHD file.

```powershell
PS C:\Toolkit\2.0.0> ConvertTo-NaLun -SourceVhd F:\win2k8.vhd -DestinationLun /vol/vol1/testwin2k8vm_vhd
```

<table>
<thead>
<tr>
<th>Path</th>
<th>TotalSize</th>
<th>SizeUsed</th>
<th>Protocol</th>
<th>Online</th>
<th>Mapped</th>
<th>Thin</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vol/vol1/testwin2k8vm_vhd</td>
<td>40.0 GB</td>
<td>40.1 GB</td>
<td>windows_2008</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

**NDMP copy client**

Sometimes it is necessary to copy data from one controller to another. Copying data through a host consumes both time and resources as data has to travel from the source to the host, then from the host to the destination. Toolkit 2.0 adds the capability to perform an NDMP copy from PowerShell in a synchronous process using **Invoke-NaNdmpCopy**, or an asynchronous process using **Start-NaNdmpCopy**, **Stop-NaNdmpCopy**, and **Get-NaNdmpCopy**. Please note that the NDMP copy operation will fail if the PowerShell session is closed before completion.

```powershell
PS C:\Toolkit\2.0.0> Start-NaNdmpCopy -SrcController 10.61.169.28 -SrcPath /vol/vol1/virtual.vhd -DstController 10.61.167.60 -DstPath /vol/vol4/ -SrcAuthType md5 -DstAuthType md5
```

WARNING: PowerShell session must remain open until the NDMP copy has completed or the operation will fail.

<table>
<thead>
<tr>
<th>Id</th>
<th>State</th>
<th>SrcPath</th>
<th>DstPath</th>
<th>BackupBytesProcessed</th>
<th>BackupBytesRemain</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>RUNNING</td>
<td>/vol/vol1/virtual.vhd</td>
<td>/vol/vol4/</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Current running NDMP copy operations can be queried using **Get-NaNdmpCopy**.

```powershell
PS C:\Toolkit\2.0.0> Get-NaNdmpCopy
```

<table>
<thead>
<tr>
<th>Id</th>
<th>State</th>
<th>SrcPath</th>
<th>DstPath</th>
<th>BackupBytesProcessed</th>
<th>BackupBytesRemain</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>SUCCESS</td>
<td>/vol/vol1/virtual.vhd</td>
<td>/vol/vol4/</td>
<td>20 MB</td>
<td>0</td>
</tr>
</tbody>
</table>

The NDMP copy cmdlets work for controllers operating in 7-Mode and Cluster-Mode, including between controllers operating in 7-Mode and Cluster-Mode.
Toolkit 2.0 adds Enable-NaNdmp, Disable-NaNdmp, Enable-NcNdmp, and Disable-NcNdmp to enable and disable NDMP on controllers operating in 7-mode or Cluster-mode.

**More complete support for CIFS shares**
Toolkit 2.0 continues building in support for Windows Server 2012, which will support Hyper-V over SMB. **NewNaVirtualDisk** now supports rapid, thin VHD file creation in CIFS shares, and **Remove-NaHostFile** now seamlessly removes files in CIFS shares.

**Installation scripts**
The beauty of PowerShell module installation is that no registration is necessary; the files merely have to be copied into one of a few potential locations. But there are still issues that can arise, most notably the Toolkit failing to load because the zone identifier security flag was not cleared after the DataONTAP.zip file was downloaded from the Internet. The Install.ps1 and Uninstall.ps1 scripts that we ship alongside the Toolkit have been improved for simpler operation, and the installer script clears the zone identifier so that the Toolkit always loads correctly. In most cases, it should be possible to accept all the defaults and install the Toolkit in seconds. You just need to have these scripts in the same directory as the DataONTAP.zip file, and you may have to be running with elevated privileges to access the system-level PowerShell directory.

```
PS C:\Toolkit\2.0.0> .\Install.ps1
Choose an installation destination:
  1: C:\Users\cknight\Documents\WindowsPowerShell\Modules
  2: C:\Windows\system32\WindowsPowerShell\v1.0\Modules

Install to directory: [2]:
Installed successfully; run "Import-Module DataONTAP" to make the cmdlets available.
```

Uninstallation is just as simple. Note that the script shows which versions are installed.

```
PS C:\Toolkit\2.0.0> .\Uninstall.ps1
Before continuing, please close all PowerShell instances that have the DataONTAP module loaded.

Choose an installation to uninstall:
  1: C:\Windows\system32\WindowsPowerShell\v1.0\Modules\DataONTAP (installed: 2.0.0.109)

Uninstall from directory: [1]:
The uninstallation completed successfully.
```

**Other enhancements**
A new cmdlet, Add-NcHostIscsiTargetPortal, handles iSCSI initiator target portal configuration for Cluster-Mode Vservers.

```
PS C:\Toolkit\2.0.0> Connect-NcController costea01
PS C:\Toolkit\2.0.0> Add-NcHostIscsiTargetPortal
TargetName                      InitiatorName               IsConnected
----------                      -------------               ---------
iqn.1992-08.com.netapp:sn.8e332132edd... Root\ISCSIPRT\0000_0   True
```

A timeout parameter is now available on Invoke-NaSsh and Invoke-NcSsh. This can speed script execution where commands are sent to multiple systems, any of which might not be reachable.

```
PS C:\Toolkit\2.0.0> Invoke-NaSsh -Name 10.10.10.10 -Credential root -Command "aggr status" -Timeout 500
Invoke-NaSsh : System.TimeoutException: Socket connection timed out
```
Get-NaEfficiency now includes additional fields to determine how far a volume or aggregate is overprovisioned. Use the –Recurse switch to cause the cmdlet to calculate this value.

```
PS C:\Toolkit\2.0.0> Get-NaEfficiency -Recurse -Volume vol2 | select Name, OverProvisioningPercent
Name    OverProvisioningPercent
------    -----------------------
vol2   135
```

```
PS C:\Toolkit\2.0.0> Get-NaEfficiency -Recurse -Aggregate * | select Name, OverProvisioningPercent
Name    OverProvisioningPercent
------    -----------------------
aggr0   88
aggr1   85
aggr2   182
```

RPC error messages could sometimes be too generic. Toolkit 2.0 should return better descriptions of any errors returned by Windows’ RPC subsystem.

```
PS C:\Toolkit\2.0.0> Connect-NaController dunn -RPC
Connect-NaController : RPC Error: The security context is invalid. Code 0xC0020032.
```

Data ONTAP 8.1 creates 64-bit aggregates by default. New-NaAggr adds a –Use32Bit switch to force creation of a 32-bit aggregate.

A bug in some versions of Data ONTAP 8 can lead to a controller panic if an igroup is created without a type specified. As a workaround, New-NaIgroup and New-NcIgroup now use a default igroup type of “windows” if no other value is set.

Get-NaClone adds more parameter sets to more closely match the behavior of the underlying API.

**New cmdlets**

These are the new cmdlets in Data ONTAP PowerShell Toolkit 2.0, not including the Cluster-Mode set:

- Invoke-NaEmsAutosupportLog
- Rename-NaFile
- Get-NaLunAttribute
- Set-NaLunAttribute
- ConvertTo-NaLun
- Disable-NaNdmp
- Enable-NaNdmp
- Invoke-NaNdmpCopy
- Start-NaNdmpCopy
- Stop-NaNdmpCopy
- Get-NaNdmpCopy

These are the categories of new cmdlets in the Cluster-Mode set:

- Audit (2 cmdlets)
- Autosupport (11 cmdlets)
- Cf (6 cmdlets)
- Clock (1 cmdlet)
- Ems (19 cmdlets)
- Host (1 cmdlet)
- Name mapping (10 cmdlets)
- Ndmp (2 cmdlets)
- Perf (4 cmdlets)
- Qos (18 cmdlets)
- Ses (3 cmdlets)
- System (7 cmdlets)
- Wafi (2 cmdlets)

**Issues fixed in Toolkit 2.0**

- Some fields could be missing from the output of `Get-NaHostDisk`.
- Some cloning operations (`Repair-NaVirtualDiskAlignment`, `Copy-NaHostFile`) could fail due to an arithmetic overflow in a Data ONTAP API.
- `Get-NaVol` could return EONTAPI_ENOMEM in some cases.
Data ONTAP PowerShell Toolkit 1.7

V2V utilities
There are already various tools available to convert disk images between the VHD and VMDK formats. But performance can be vastly improved if the data blocks are cloned in-place on the storage controller rather than copied via a Windows host. Toolkit 1.7 adds two cmdlets, `ConvertTo-NaVhd` and `ConvertTo-NaVmdk`, that can convert between these formats in a matter of seconds. These measurements were taken on the same Windows 2008 R2 server using the same source VMDK file under identical conditions:

<table>
<thead>
<tr>
<th>Task</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert a 32 GB flat VMDK file to fixed VHD using a popular commercial tool</td>
<td>46 minutes</td>
</tr>
<tr>
<td>Convert a 32 GB flat VMDK file to fixed VHD using ConvertTo-NaVhd</td>
<td>63 seconds</td>
</tr>
</tbody>
</table>

Note that a bootable VMDK file may not yield a bootable VHD file, and vice versa. Some OS- and hypervisor-specific modifications may be necessary to ensure the correct storage drivers are present and configured in the image before it will boot. Some operating systems may be able to recover from this automatically, possibly by running a repair task upon first boot after a file format conversion.

VHD file alignment
Misaligned I/O is one of the most common causes of poor disk performance in virtualized environments. Toolkit 1.7 adds two cmdlets for detecting and correcting alignment problems in VHD files.

`Get-NaVirtualDiskAlignment` reads the first sector of a fixed VHD file to determine if any of the partitions on the VHD are misaligned. The cmdlet only supports MBR-style disk files, as alignment problems on GPT disks are uncommon. The partition type is unimportant, and extended boot records are also supported.

```powershell
PS C:\Toolkit\1.7.0> Get-NaVirtualDiskAlignment M:\linux.vhd

VirtualDisk: M:\linux.vhd

<table>
<thead>
<tr>
<th>IsBootable</th>
<th>AbsoluteStartingLba</th>
<th>Size</th>
<th>IsExtendedBootRecord</th>
<th>IsAligned</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>64</td>
<td>102 MB</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>208848</td>
<td>15 GB</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>
```

`Repair-NaVirtualDiskAlignment` will correct misalignments in fixed VHD files that are formatted as MBR-style disks. Unlike some other Toolkit cmdlets such as `Copy-NaHostFile` or `ConvertTo-NaVhd`, correcting alignment issues is not possible using WAFL block cloning (because the cloned blocks would also be misaligned), so a data copy is required. For data in LUNs, Data ONTAP 7.3.5 and higher versions of 7.3.x perform this operation rapidly by offloading the copy operation to the storage controller. Data ONTAP 8 does not yet provide this capability, so this cmdlet falls back to a slower host-based copy as needed. For CIFS shares, copy offload is used by this cmdlet for all versions of Data ONTAP 7.3.5+ and 8.0.1+.

Data ONTAP 8.1 (Cluster-Mode)
Toolkit 1.7 adds 227 new cmdlets that implement Cluster-Mode APIs, meaning that users have a critical mass of 375 cmdlets for managing Cluster-Mode.
While there are many similarities between Data ONTAP (Cluster-Mode) and Data ONTAP (7-Mode), there are also several key differences that impact PowerShell users. Most importantly, Cluster-Mode has no equivalent to "vfiler0", which corresponds to a physical controller running 7-Mode. Cluster-Mode has a single "admin" Vserver for managing physical resources in the cluster (disks, nodes, aggregates, network ports, etc.) and one or more data Vservers for serving data and managing most virtual components related to serving data (volumes, qtrees, LUN, protocols, snapshots, etc.).

It is critical to note that some APIs (and therefore some cmdlets) are only handled by the cluster admin Vserver, while others are only handled by the data Vservers. The Toolkit user is responsible for directing cmdlets to the proper Vserver. The Toolkit provides several methods of directing cmdlets, and it does so automatically in some cases.

For example, to connect to a cluster admin Vserver:

```
PS C:\Toolkit\1.7.0> Connect-NcController cluster1
```

While connected to a cluster admin Vserver, you can direct cmdlets to a data Vserver in several ways:

1. Set the data Vserver while connecting to the cluster Vserver. All subsequent cmdlets are directed to the specified Vserver.

```
PS C:\Toolkit\1.7.0> Connect-NcController cluster1 -Vserver vserver1
```

2. Set the data Vserver anytime after connecting to the cluster Vserver:

```
PS C:\Toolkit\1.7.0> Connect-NcController cluster1
PS C:\Toolkit\1.7.0> $global:CurrentNcController.Vserver = "vserver1"
```

3. Specify the data Vserver destination as a cmdlet argument:

```
PS C:\Toolkit\1.7.0> Set-NcLunSize /vol/unitTestLuns/georgeTestFcp01 +10% -VserverContext vserver1
```

4. Use getters from either cluster or data Vserver, and pipe their output into other cmdlets. The Toolkit includes enough context on objects written to the pipeline to determine where to direct subsequent cmdlets that operate on those objects:

```
PS C:\Toolkit\1.7.0> Connect-NcController cluster1
PS C:\Toolkit\1.7.0> Get-NcVol | Set-NcLunSize -NewSize +10%
```

You can also connect directly to a data Vserver management interface. While connected directly to a data Vserver, cmdlets may not be issued to the cluster admin Vserver or any other data Vserver.

```
PS C:\Toolkit\1.7.0> Connect-NcController vserver1
```

Many getter cmdlets work in either the cluster admin or data Vserver contexts. To determine where cmdlets may be directed, use Get-NcHelp and check the Family field:

```
PS C:\Toolkit\1.7.0> Get-NcHelp -Category volume
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Family</th>
<th>Api</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dismount-NcVol</td>
<td>volume</td>
<td>{vserver}</td>
<td>{volume-unmount}</td>
</tr>
<tr>
<td>Get-NcVol</td>
<td>volume</td>
<td>{cluster, vserver}</td>
<td>{volume-get-iter}</td>
</tr>
<tr>
<td>Get-NcVolAutosize</td>
<td>volume</td>
<td>{vserver}</td>
<td>{volume-autosize-get}</td>
</tr>
<tr>
<td>Get-NcVolClone</td>
<td>volume</td>
<td>{cluster, vserver}</td>
<td>{volume-clone-get-iter}</td>
</tr>
<tr>
<td>Get-NcVolCloneSplit</td>
<td>volume</td>
<td>{vserver}</td>
<td>{volume-clone-split-status}</td>
</tr>
</tbody>
</table>
File cloning for CIFS

*Copy-NaHostFile* transparently adds support for files in CIFS shares. So regardless of whether a file to be cloned is on a LUN or in a share on Data ONTAP, the cmdlet figures out where it resides and takes the appropriate steps to rapidly clone the file. Controller credentials are retrieved from the Toolkit’s cache if they are not provided explicitly. The exception to this rule is Cluster-Mode CIFS, where there is no means for the cmdlet to discern the Vserver management LIF from the share info; the –Controller parameter (or current connection context) must be valid in that case.

Space reclamation for Cluster Shared Volumes

*Invoke-NaHostVolumeSpaceReclaim* adds support for CSVs in clustered Hyper-V environments. The cmdlet uses a different algorithm for CSVs due to their lack of support for certain NTFS APIs. During the reclamation process, the CSV will become 98% full for a brief time; this should not be a cause for concern as most I/O to CSV is within VHD files, which should be fixed-size VHDs in accordance with NetApp best practices. Furthermore, the temporary space consumption is only apparent from Windows’ perspective, as the associated WAFL blocks are freed continuously during the process, so the cmdlet should work even in thin-provisioned environments.

Other enhancements

LUN objects now include alignment info (requires Data ONTAP 8.0.1+):

PS C:\Toolkit\1.7.0> Get-NaLun | select Path, Alignment

<table>
<thead>
<tr>
<th>Path</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vol/beamdev2k8/lun</td>
<td>misaligned</td>
</tr>
<tr>
<td>/vol/beamdev2k8/vmdk_lun</td>
<td>aligned</td>
</tr>
<tr>
<td>/vol/DPM/x3550rxe8</td>
<td>aligned</td>
</tr>
<tr>
<td>/vol/knighttest/testlun</td>
<td>indeterminate</td>
</tr>
<tr>
<td>/vol/knighttest/testlun3</td>
<td>aligned</td>
</tr>
<tr>
<td>/vol/luns/deleteMe</td>
<td>aligned</td>
</tr>
<tr>
<td>/vol/luns/fcptest.lun</td>
<td>aligned</td>
</tr>
</tbody>
</table>

SIS objects and cmdlets now support volume compression (requires Data ONTAP 8.0.1+):

PS C:\Toolkit\1.7.0> Get-NcSis | select Path, IsCompressionEnabled

<table>
<thead>
<tr>
<th>Path</th>
<th>IsCompressionEnabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vol/costea01_root_vol</td>
<td>False</td>
</tr>
<tr>
<td>/vol/grupoMode</td>
<td>False</td>
</tr>
<tr>
<td>/vol/pp_ds</td>
<td>False</td>
</tr>
<tr>
<td>/vol/ppds_nfs</td>
<td>False</td>
</tr>
<tr>
<td>/vol/unitTestLuns</td>
<td>True</td>
</tr>
</tbody>
</table>

Controller passwords are now held as SecureString values for greater security.

*Get-NaHostDisk* now reports several more target-side fields:

PS C:\Toolkit\1.7.0> Get-NaHostDisk 1 | Format-List *
DiskSize                 : 9346023751680
DiskWmiPath              : \\x3550\r\root\cimv2\Win32_DiskDrive.DeviceID="\\\\PHYSICALDRIVE1"
ControllerName           : TS60808
ControllerIgroup         : bigigroup
ControllerIgroupType     : Windows
ControllerAddresses      : {10.61.175.81}
ControllerLunPath        : /vol/bigvol/biglun
ControllerVolumeName     : bigvol
ControllerLunOsType      : hyper_v
ControllerSnapshot       :
ControllerOntapRevision  : 8.0.2 7-Mode
ControllerModel          : FAS6080
IscsiTarget              : 10.61.175.81:3260
FcpTarget                :

**Wait-NaHostDisk** can now detect disk additions by serial number:

PS C:\Toolkit\1.7.0> Wait-NaHostDisk -SerialNumber HnaMP4g4okEC

<table>
<thead>
<tr>
<th>HostDrivePath</th>
<th>Disk Size</th>
<th>ControllerPath</th>
</tr>
</thead>
<tbody>
<tr>
<td>M:\</td>
<td>8.5 TB</td>
<td>TS60808:/vol/bigvol/biglun</td>
</tr>
</tbody>
</table>

CIFS share cmdlets now include parameters for both access-based enumeration and namespace caching.

**New-NaSnapshotMulti** adds an option to output newly created snapshots:

PS C:\Toolkit\1.7.0> New-NaSnapshotMulti vol1,vol2 -SnapName multiSnap -GetSnapshots

<table>
<thead>
<tr>
<th>Name</th>
<th>Created</th>
<th>Total</th>
<th>Cumulative</th>
<th>Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiSnap</td>
<td>1/30/2012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiSnap</td>
<td>1/30/2012</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A new cmdlet, **Get-NaControllerError**, lists all error codes and descriptions for Data ONTAP. The codes are common to 7-Mode and Cluster-Mode:

PS C:\Toolkit\1.7.0> Get-NaControllerError

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EONTAPI_EPERM</td>
<td>Operation not permitted</td>
</tr>
<tr>
<td>2</td>
<td>EONTAPI_ENOENT</td>
<td>No such file or directory</td>
</tr>
<tr>
<td>3</td>
<td>EONTAPI_ESRCH</td>
<td>No such process</td>
</tr>
<tr>
<td>4</td>
<td>EONTAPI_EINTR</td>
<td>Interrupted system call</td>
</tr>
<tr>
<td>5</td>
<td>EONTAPI_EIO</td>
<td>Input/Output error</td>
</tr>
<tr>
<td>6</td>
<td>EONTAPI_ENXIO</td>
<td>Device not configured</td>
</tr>
<tr>
<td>7</td>
<td>EONTAPI_E2BIG</td>
<td>Argument list too long</td>
</tr>
<tr>
<td>8</td>
<td>EONTAPI_ENOEXEC</td>
<td>Exec format error</td>
</tr>
<tr>
<td>9</td>
<td>EONTAPI_EBADF</td>
<td>Bad File descriptor</td>
</tr>
<tr>
<td>10</td>
<td>EONTAPI_ECHILD</td>
<td>No child process</td>
</tr>
<tr>
<td>11</td>
<td>EONTAPI_EDeadlk</td>
<td>Resource deadlock avoided</td>
</tr>
<tr>
<td>12</td>
<td>EONTAPI_ENOMEM</td>
<td>Cannot allocate memory</td>
</tr>
<tr>
<td>13</td>
<td>EONTAPI_EACCES</td>
<td>Permission denied</td>
</tr>
</tbody>
</table>

Under certain conditions, previous releases could consume extra system memory. Toolkit 1.7 offers improved memory handling and significantly better performance with repeated cmdlet invocations.

**Set-NaVirtualDiskSize** no longer clobbers a partition of unknown type during a shrink operation, and it offers a -Preserve switch to leave known partition types unchanged during grow or shrink operations.
Toolkit 1.7 adds persistent storage of controller connections. Both NaController and NcController types, as emitted by `Connect-NaController` and `Connect-NcController`, are serializable using two new cmdlets, `ConvertTo-SerializedString` and `ConvertFrom-SerializedString`. This allows a Toolkit connection context to be saved and used by other processes on the same host. Any .NET objects that implement the interface `System.Runtime.Serialization.ISerializable` should be supported by these new cmdlets.

**New cmdlets**

These are the new cmdlets in Data ONTAP PowerShell Toolkit 1.7, not including the Cluster-Mode set:

- ConvertTo-NaVhd
- ConvertTo-NaVmdk
- Get-NaVirtualDiskAlignment
- Repair-NaVirtualDiskAlignment
- Enable-NaStorageAdapter
- Get-NaStorageAdapter
- Get-NaStorageAdapterInfo
- Get-NaControllerError
- ConvertTo-SerializedString
- ConvertFrom-SerializedString

These are the categories of new cmdlets in the Cluster-Mode set:

- Cifs (32 cmdlets)
- Clone (1 cmdlet)
- Cluster peer (6 cmdlets)
- Disk (10 cmdlets)
- Exports (9 cmdlets)
- Fc (4 cmdlets)
- Fcp (20 cmdlets)
- File (9 cmdlets)
- Igroup (10 cmdlets)
- Iscsi (30 cmdlets)
- Net (27 cmdlets)
- Nfs (13 cmdlets)
- Portset (5 cmdlets)
- Quota (9 cmdlets)
- Security (13 cmdlets)
- Sis (10 cmdlets)
- Snapmirror (16 cmdlets)
- Storage adapter (3 cmdlets)

**Issues fixed in Toolkit 1.7**

- Fixed null handling in 7-Mode cmdlet wildcard patterns.
- Fixed error handling in `Get-NaVol`.
- Fixed `Invoke-NaHostVolumeSpaceReclaim` to support LUNs larger than 4 TB.
Data ONTAP PowerShell Toolkit 1.6

VHD file utilities

Virtual hard disk (VHD) files are central to multiple hypervisors, including Microsoft Hyper-V and Citrix XenServer. A common pain point is the time it takes to create a fixed VHD file, since it is fully zeroed during creation. And because it is zeroed, the VHD consumes its full size on its underlying storage medium. A common alternative is a dynamic VHD file, but that isn't any better because I/O to a dynamic VHD will always be misaligned.

A new cmdlet, **New-NaVirtualDisk**, creates a VHD file of any size instantly and then rapidly frees the associated blocks on Data ONTAP so no space is consumed initially. The VHD file must reside on a disk that is a Data ONTAP LUN, but no controller credentials are required. A simple experiment shows the improvement:

<table>
<thead>
<tr>
<th>Description</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a 500 GB fixed VHD using Hyper-V Manager in Windows Server 2008 R2</td>
<td>1 hour, 41 minutes</td>
</tr>
<tr>
<td>Create a 500 GB fixed VHD using New-NaVirtualDisk</td>
<td>21 seconds</td>
</tr>
</tbody>
</table>

To use this cmdlet, just provide a fully-qualified path for the new VHD file and a size:

```
PS C:\Toolkit\1.6.0> New-NaVirtualDisk M:\VM1.vhd 500g
```

```
Mode LastWriteTime Length Name
---- ------------- ------ -----
-a--- 10/27/2011  4:34 PM 536870912512 VM1.vhd
```

Another cmdlet, **Set-NaVirtualDiskSize**, can rapidly grow or shrink a fixed VHD file. Any space added to or removed from the file is reclaimed on the storage controller. If one or more NTFS data partitions are present inside the VHD file, the partition nearest the end of the file is automatically grown or shrunk using NTFS APIs. The VHD file must not be in use during this operation, such as by a running virtual machine, because the VHD is temporarily mounted on the local host for inspection and adjustment of the partitions within. Note that once a partition is present in a VHD file, the file may not be shrinkable beyond a certain point due to how NTFS manages the partition contents.

```
PS C:\Toolkit\1.6.0> Set-NaVirtualDiskSize M:\VM1.vhd 1t
Grow VHD
Are you sure you want to grow the VHD?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y") : Y
```

```
Mode LastWriteTime Length Name
---- ------------- ------ -----
-a--- 10/27/2011  4:37 PM 1099511628288 VM1.vhd
```

**Mount-NaVirtualDisk** can mount any VHD file as a disk on the local host, while **Dismount-NaVirtualDisk** dismounts a VHD-based disk so that it may be used elsewhere, such as in a virtual machine configuration. These cmdlets are most useful when combined with the new disk and volume management cmdlets in Toolkit 1.6.
**Space reclamation**

When a file is deleted from an NTFS-formatted volume hosted by a Data ONTAP LUN, the associated blocks on the storage controller are not automatically freed but remain allocated until NTFS writes new data to the same location. Over time, this effect can appear to reduce the space savings offered by thin provisioned LUNs. Two cmdlets in Toolkit 1.6 help alleviate this issue.

**Remove-NaHostFile** frees the associated WAFL blocks from the LUN before deleting the file. A secondary benefit of this cmdlet is a measure of security. There are disk utilities that can recover the contents of deleted files, but once the blocks are freed, they always read as zeroed blocks, so there is no possibility the data previously contained in the deleted file could ever be retrieved. Note that this cmdlet, like most Toolkit cmdlets that operate on the local file system, requires a fully-qualified file path.

**Invoke-NaHostVolumeSpaceReclaim** scans an entire NTFS volume and frees all WAFL blocks that are not associated with existing files. It is safe to run this cmdlet during normal I/O operation. A common use case might be to schedule a script to run this cmdlet against all ONTAP-hosted disks during periods of lower activity, perhaps on a weekly basis, to keep the LUNs running at peak space efficiency. Note that **Invoke-NaHostVolumeSpaceReclaim** does not (yet) support cluster-shared volumes (CSVs).

Performance of the space reclamation commands can vary depending on several factors, but this result from reclaiming a one-terabyte LUN suggests the potential of this cmdlet:

```powershell
PS C:\Toolkit\1.6.0> Measure-Command { Invoke-NaHostVolumeSpaceReclaim R:\ } | select TotalSeconds
TotalSeconds
---------
 92.9664409
```

To reclaim all unused space on all LUNs hosted by Data ONTAP and mounted on the local host, try this:

```powershell
PS C:\Toolkit\1.6.0> Get-NaHostDisk -DataOntap | Invoke-NaHostVolumeSpaceReclaim -Confirm:$false
```

**Windows disk and volume management**

Previous Toolkit releases have provided the capability to initiate iSCSI connections to Data ONTAP, create new LUNs and igroups, and map LUNs to Windows hosts. But that isn’t enough, as one must still instruct Windows to scan for the new disk, initialize it, and create one or more formatted volumes on it. Toolkit 1.6 completes this picture by adding host-side disk and volume cmdlets.

- Start-NaHostDiskRescan
- Initialize-NaHostDisk
- Set-NaHostDisk
- Wait-NaHostDisk
- Get-NaHostVolume
- New-NaHostVolume
- Format-NaHostVolume
- Mount-NaHostVolume
- Dismount-NaHostVolume
- Set-NaHostVolumeSize
- Remove-NaHostVolume
A few of these (New-NaHostVolume, Format-NaHostVolume, Remove-NaHostVolume) include optional Data ONTAP space reclamation.

**New-NaHostVolume** may be used as a shortcut to initialize a new disk and automatically create and mount a new NTFS partition within it. This example shows how to format and mount all newly discovered disks:

```powershell
PS C:\Toolkit\1.6.0> Get-NaHostDisk -Uninitialized | New-NaHostVolume

MountPoints Disk Size GuidPaths
--------- ---- ---------------
{E:\} 4 10.0 GB {\Volume{adada439-fabb-11e0-8fb4-00155da1ce00}}
{F:\} 5 10.0 GB {\Volume{adada441-fabb-11e0-8fb4-00155da1ce00}}
{G:\} 6 10.0 GB {\Volume{adada449-fabb-11e0-8fb4-00155da1ce00}}
{H:\} 7 10.0 GB {\Volume{adada451-fabb-11e0-8fb4-00155da1ce00}}
{I:\} 8 10.0 GB {\Volume{adada459-fabb-11e0-8fb4-00155da1ce00}}
```

If more control over the various steps is needed, one may use Initialize-NaHostDisk, New-NaHostVolume, Format-NaHostVolume, and Mount-NaHostVolume separately.

```powershell

MountPoints Disk Size GuidPaths
--------- ---- ---------------
{I:\} 12 49.9 GB {\Volume{4d6e4e9d-a449-4de3-be8c-bc9df182c5b}}
```

The new VHD and volume cmdlets may be combined to create and format new VHD files at the same time. **New-NaHostVolume** always creates partitions that are aligned to a 1MB boundary, so this is a great way to create VHD files that will not suffer from misaligned I/O (assuming they are placed in a LUN of the proper OS type) regardless of the operating system or data set placed into the new partition.

```powershell
PS C:\Toolkit\1.6.0> New-NaVirtualDisk I:\VM1.vhd 20g | Mount-NaVirtualDisk | New-NaHostVolume | Dismount-NaVirtualDisk -Confirm:$false

Mode LastWriteTime Length Name
---- --------------- ----- ----
-a--- 10/25/2011 2:22 PM 21474836992 VM1.vhd
```

Building on our provisioning script from Toolkit 1.5, we can now provision storage to the point where it is formatted and mounted on the local host. The new cmdlets are shown in red.

```
Connect-NaController TS60808
New-NaAggr bigaggr -Use64Bit -DiskCount 32
New-NaVol bigvol -Aggregate bigaggr 4t
Set-NaSnapshotReserve bigvol 0
New-NaLun -Path /vol/bigvol/biglun -Size 3.5t -Type hyper_v
Add-NaHostIscsiTargetPortal
$session = Get-NaHostIscsiSession -TargetName (Get-NaIscsiNodeName)
if ($session -eq $null) { Get-NaHostIscsiTarget (Get-NaIscsiNodeName) | Connect-NaHostIscsiTarget }
New-NaIgroup biggroup iscsi windows
Get-NaHostIscsiAdapter | Add-NaIgroupInitiator biggroup
Add-NaLunMap /vol/bigvol/biglun biggroup
Start-NaHostDiskRescan
Wait-NaHostDisk /vol/bigvol/biglun TS60808 | New-NaHostVolume

HostDrivePath Disk Size ControllerPath
--------- ---- -------------------
M:\ 14 3.5 TB TS60808:/vol/bigvol/biglun
```
**File cloning**

*Copy-NaHostFile* can clone individual files within (or between) NTFS-formatted Data ONTAP LUNs. The LUN(s) must be mounted on the local host and reside on the same Data ONTAP volume. The cmdlet uses Data ONTAP’s SIS-clone feature to clone the files rapidly; no data is copied via the Windows host and the block sharing on Data ONTAP creates highly space-efficient clones. This cmdlet can support a number of interesting use cases, such as duplicating sysprepped VHD files in virtual machine cloning workflows.

```powershell
PS C:\\Toolkit\\1.6.0> Copy-NaHostFile M:\SQL_VM_Master.vhd M:\VM1.vhd
```

<table>
<thead>
<tr>
<th>Mode</th>
<th>LastWriteTime</th>
<th>Length</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>10/27/2011</td>
<td>5:48 PM</td>
<td>VM1.vhd</td>
</tr>
</tbody>
</table>

**SSH setup**

A common request has been configuration of the SSH server within Data ONTAP. There is no API for this, and “Invoke-NaSsh secureadmin setup” won’t work until this step is completed: a classic chicken-and-egg problem. A new cmdlet, *Initialize-NaSecureAdminSsh*, invokes the equivalent of “secureadmin setup”.

**Vfiler migration**

Four new cmdlets in the 'vfiler' category support the vfiler migration feature of Data ONTAP.

**More complete API coverage**

Toolkit 1.6 adds 31 cmdlets across several categories to complete our planned coverage of Data ONTAP (7-Mode) APIs. Many of the new cmdlets require Data ONTAP 8.0 or later.

**Data ONTAP 8.1 (Cluster-Mode)**

Coinciding with the release of Data ONTAP 8.1 (Cluster-Mode) RC1, Toolkit 1.6 includes a “preview” set of cmdlets for managing cluster mode. While Cluster-Mode unquestionably represents the future of Data ONTAP, it is sufficiently different from 7-Mode that it is not practical to graft support for it into the PowerShell Toolkit’s existing 7-Mode cmdlets. We have taken the opportunity to improve the architecture of the Cluster-Mode cmdlets, with support for multiple controller connections, seamless Vserver handling, Cluster-Mode’s advanced queries, and more. Below are a few examples; note the -Nc cmdlet prefix denoting Cluster-Mode.

**Connect to a cluster LIF:**

```powershell
PS C:\\Toolkit\\1.6.0> Connect-NcController cerebus
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Vserver</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>cerebus</td>
<td>10.58.94.234</td>
<td></td>
<td>NetApp Release 8.1RC1 Cluster-Mode</td>
</tr>
</tbody>
</table>

**List nodes:**

```powershell
PS C:\\Toolkit\\1.6.0> Get-NcNode
```

<table>
<thead>
<tr>
<th>Name</th>
<th>NodeModel</th>
<th>NodeLocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cerebus-03</td>
<td>FAS3070</td>
<td>SeaLab-A7</td>
</tr>
<tr>
<td>Cerebus-04</td>
<td>FAS3040</td>
<td>SeaLab-A7</td>
</tr>
<tr>
<td>Cerebus-05</td>
<td>FAS3040</td>
<td>SeaLab-A7</td>
</tr>
<tr>
<td>Cerebus-06</td>
<td>FAS3040</td>
<td>SeaLab-A7</td>
</tr>
</tbody>
</table>
List volumes on multiple Vservers using a wildcard. It is important to note that the wildcards in this example, as well as in nearly all the Cluster-Mode cmdlets, are offloaded to Data ONTAP so as to limit how much information is transmitted to and filtered by the Windows host.

PS C:\Toolkit\1.6.0> Get-NcVol -Vserver Parker*

Use a Cluster-Mode advanced query to find aggregates greater than one terabyte, offloading the data filtering to Data ONTAP and transferring only the objects of interest to the host:

PS C:\Toolkit\1.6.0> $q = Get-NcAggr -Template -Fill
PS C:\Toolkit\1.6.0> $q.TotalSize = ">1tb"

Remove weekly snapshots from a set of volumes on a single Vserver. Note that the snapshot removal would work, even though we connected to a cluster LIF which doesn’t support the snapshot-delete API.

PS C:\Toolkit\1.6.0> Get-NcVol -Vserver Lionetti01 -Volume VMSet* | Get-NcSnapshot -SnapName weekly* | Remove-NcSnapshot -WhatIf

Connect to two different clusters simultaneously using the -Add switch and list their node info:

PS C:\Toolkit\1.6.0> Connect-NcController VxeRubble

PS C:\Toolkit\1.6.0> Connect-NcController cerebus -Add

PS C:\Toolkit\1.6.0> $global:CurrentNcController
Other enhancements

The Toolkit now offers more standardized data size inputs that may be specified as bytes or as string-formatted values with units. Cmdlets which change data size values all accept absolute size values, positive or negative size increments, and positive or negative percentages. For example:

Grow a LUN by 20 GB:

```
PS C:\Toolkit\1.6.0> Set-NaLunSize /vol/vol2clone/lun1 +20g
```

```
Path                TotalSize SizeUsed Protocol    Online Mapped Thin Comment
----                --------- ---------- -------- -------- ------ ------ ------ ------
/vol/vol2clone/lun1   40.0 GB   6.8 GB windows_gpt False  False  True
```

Grow a LUN by 20 percent:

```
PS C:\Toolkit\1.6.0> Set-NaLunSize /vol/vol2clone/lun1 +20%
```

```
Path                TotalSize SizeUsed Protocol    Online Mapped Thin Comment
----                --------- ---------- -------- -------- ------ ------ ------ ------
/vol/vol2clone/lun1   24.0 GB   6.8 GB windows_gpt False  False  True
```

Get-NaSystemLog offers several improvements:

- Support for the SnapMirror log.
- Better performance when -StartTime parameter is specified.
- Correct -Follow switch behavior through a log rotate.
- The -EventType and -Severity parameters accept arrays for improved filtering.

Add-NaHostIscsiTargetPortal and Connect-NaHostIscsiTarget now include -RadiusGenerate and -RadiusVerify switches.

Set-NaVfilerAddress now allows IP addresses to be configured without requiring the user to create an IpbindingInfo object.

The output of Get-NaLunMap now contains the LUN path, enabling better pipeline support:

```
PS C:\Toolkit\1.6.0> Get-NaLunMap /vol/vol2clone/lun1 | Remove-NaLunMap
```

```
Path                TotalSize SizeUsed Protocol    Online Mapped Thin Comment
----                --------- ---------- -------- -------- ------ ------ ------ ------
/vol/vol2clone/lun1   24.0 GB   6.8 GB windows_gpt False  False  True
```

Toolkit 1.6 reports connection timeouts more clearly.
New cmdlets
These are the new cmdlets in Data ONTAP PowerShell Toolkit 1.6, not including the Cluster-Mode set:

- Disable-NaCopyOffload
- Enable-NaCopyOffload
- Test-NaCopyOffload
- Ping-NaFcp
- Ping-NaFcpInfo
- Test-NaFpolicy
- Copy-NaHostFile
- Dismount-NaHostVolume
- Dismount-NaVirtualDisk
- Format-NaHostVolume
- Get-NaHostVolume
- Initialize-NaHostDisk
- Invoke-NaHostVolumeSpaceReclaim
- Mount-NaHostVolume
- Mount-NaVirtualDisk
- New-NaHostVolume
- New-NaVirtualDisk
- Remove-NaHostFile
- Remove-NaHostVolume
- Set-NaHostDisk
- Set-NaHostVolumeSize
- Set-NaVirtualDiskSize
- Start-NaHostDiskRescan
- Wait-NaHostDisk
- Get-NaLunClone
- Get-NaLunTargetDeviceId
- ConvertTo-NaUnixUser
- ConvertTo-NaWindowsUser
- Get-NaUnixGroup
- Get-NaNetDcb
- Get-NaNetDcbPriority
- Add-NaRadiusServer
- Disable-NaRadius
- Enable-NaRadius
- Get-NaRadiusServer
- Get-NaRadiusStatistics
- Remove-NaRadiusServer
- Reset-NaRadiusStatistics
- Test-NaRadius
- Initialize-NaSecureAdminSsh
- Get-NaSnapmirrorConnection
- Remove-NaSnapmirrorConnection
- Set-NaSnapmirrorConnection
- Get-NaSnapshotRestoreStatus
- Complete-NaVfilerMigrate
- Get-NaVfilerMigrate
- Start-NaVfilerMigrate
- Stop-NaVfilerMigrate
Invoke-NaWafSync

**Issues fixed in Toolkit 1.6**
- Enabled monitoring by Invoke-NaSysstat of LUNs owned by a vfiler.
- Add-NaNetVlan now works with 8.0.2. The -Gvrp switch can be used to enable GVRP on created VLAN interfaces.
- Initialize-NaController supports Data ONTAP 8 (7-Mode).
Data ONTAP PowerShell Toolkit 1.5

Storage efficiency calculations

Part of Data ONTAP’s unparalleled value proposition centers around storage efficiency using technologies such as deduplication, thin provisioning, snapshots, cloning and compression. A new cmdlet, Get-NaEfficiency, embeds the NetApp standard calculations for storage efficiency. This is especially useful for monitoring and reporting scripts in thin-provisioned environments.

PS C:\Toolkit\1.5.0> Get-NaEfficiency vol*

<table>
<thead>
<tr>
<th>Name</th>
<th>Capacity</th>
<th>Used</th>
<th>Free</th>
<th>SnapUsage</th>
<th>Reserve</th>
<th>Returns</th>
<th>EfficiencyPercentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>vol0</td>
<td>101.4 GB</td>
<td>15.7 GB</td>
<td>85.7 GB</td>
<td>5.7 GB</td>
<td>0</td>
<td>16.2 GB</td>
<td>31%</td>
</tr>
<tr>
<td>vol1</td>
<td>16.0 GB</td>
<td>809.4 MB</td>
<td>15.2 GB</td>
<td>3.0 MB</td>
<td>0</td>
<td>0</td>
<td>9%</td>
</tr>
<tr>
<td>vol2</td>
<td>180.0 GB</td>
<td>95.3 GB</td>
<td>84.7 GB</td>
<td>62.3 GB</td>
<td>0</td>
<td>82.6 GB</td>
<td>98%</td>
</tr>
<tr>
<td>vol3</td>
<td>400.0 MB</td>
<td>404.0 KB</td>
<td>399.6 MB</td>
<td>596.0 KB</td>
<td>0</td>
<td>354.3 KB</td>
<td>0%</td>
</tr>
<tr>
<td>vol1clone</td>
<td>16.0 GB</td>
<td>809.5 MB</td>
<td>15.2 GB</td>
<td>1.8 MB</td>
<td>0</td>
<td>1.6 GB</td>
<td>14%</td>
</tr>
<tr>
<td>vol2clone</td>
<td>180.0 GB</td>
<td>82.7 GB</td>
<td>55.3 GB</td>
<td>53.5 GB</td>
<td>0</td>
<td>83.5 GB</td>
<td>92%</td>
</tr>
</tbody>
</table>

See the script posted on the Data ONTAP PowerShell communities that imports the data from Get-NaEfficiency into Microsoft Excel and displays it in tabular and graphical form:
Host-side iSCSI & FC utilities

Several new cmdlets in the 'host' category support configuration of iSCSI interfaces and connections in Windows.

- Add-NaHostIscsiTargetPortal
- Connect-NaHostIscsiTarget
- Disconnect-NaHostIscsiTarget
- Get-NaHostIscsiAdapter
- Get-NaHostIscsiSession
- Get-NaHostIscsiTarget

These cmdlets provide for enumerating host-side iSCSI interfaces, sessions and targets. Just having a simple way to get the IQN of the Windows host is useful.

PS C:\Toolkit\1.5.0> Get-NaHostIscsiAdapter

<table>
<thead>
<tr>
<th>InstanceName</th>
<th>SerialNumber</th>
<th>Iqn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root\ISCSIPRT\0000_0</td>
<td>MSFT-05-1991</td>
<td>iqn.1991-05.com.microsoft:x3550rre7.rtprre.testdomain</td>
</tr>
</tbody>
</table>

PS C:\Toolkit\1.5.0> Get-NaHostIscsiTarget
Get-NaHostIscsiSession includes information about local disks and remote targets.

These getter cmdlets all include WMI path fields, so the PowerShell WMI accelerator may be used to make WMI queries for related objects.
The new cmdlets also support setting up or tearing down iSCSI connections to Data ONTAP storage controllers. Here is a simple script that demonstrates how to set up a new aggregate, volume and LUN on a controller, establish a new iSCSI connection to the controller from the current Windows host, set up a new initiator group containing the host’s initiator, and map the new LUN to the host. The new cmdlets are shown in red.

Connect-NaController 10.61.175.81
New-NaAggr bigaggr -Use64Bit -DiskCount 32
New-NaVol bigvol -Aggregate bigaggr 4t
Set-NaSnapshotReserve bigvol 0
New-NaLun -Path /vol/bigvol/biglun -Size (3.5tb) -Type hyper_v
Add-NaHostIscsiTargetPortal
Get-NaHostIscsiTarget (Get-NaIscsiNodeName) | Connect-NaHostIscsiTarget
New-NaIgroup bigigroup iscsi windows
Get-NaHostIscsiAdapter | Add-NaIgroupInitiator bigigroup
Add-NaLunMap /vol/bigvol/biglun bigigroup

Multi-path connections are also supported:

$c = Connect-NaController apollo -Transient
/target | Connect-NaHostIscsiTarget -InitiatorIp 172.16.2.7 -EnableMultipath
/target | Connect-NaHostIscsiTarget -InitiatorIp 172.16.3.7

Toolkit 1.5 also adds a cmdlet to list fibre channel interfaces, Get-NaHostFcAdapter, which obviates the need to use vendor-specific tools to obtain this information.

PS C:\Toolkit\1.5.0> Get-NaHostFcAdapter

Model   ModelDescriptor                      DriverName DriverVersion NodeWwn
-------------- ------------------------------- --------- -------------- ----------
QLE2462 QLogic QLE2462 Fibre Channel Adapter ql2300.sys       9.1.8.6 20:00:00:e0:8b:89:2e:2b
QLE2462 QLogic QLE2462 Fibre Channel Adapter ql2300.sys       9.1.8.6 20:01:00:e0:8b:a9:2e:2b

ONTAP log parsing and monitoring

By popular request, Toolkit 1.5 includes a new cmdlet, Get-NaSystemLog, that parses and displays the contents of three Data ONTAP logs: messages, auditlog and ems.

PS C:\Toolkit\1.5.0> Get-NaSystemLog -MessageLog

TimeStampDT           Source  Severity EventType                           Message
----------- ------ ---------- ---------- ------------------------------- ---------
6/19/2011 12:04:49 AM TS60808 notice kern.log.rotate System TS60808 (ID 01...
6/19/2011 1:00:00 AM TS60808 info kern.uptime.filer 1:00am up 16 days, 10...
6/19/2011 1:00:02 AM TS60808 notice raid.rg.scrub.start /sv_backup_aggr8/plex...
6/19/2011 1:00:02 AM TS60808 notice raid.rg.scrub.start /aggr0/plex2/rg0: sta...
6/19/2011 1:00:02 AM TS60808 notice raid.rg.scrub.start /aggr0/plex5/rg0: sta...
6/19/2011 2:00:00 AM TS60808 info kern.uptime.filer 2:00am up 16 days, 11...
6/19/2011 2:10:01 AM TS60808 notice raid.rg.scrub.done /sv_backup_aggr8/plex...
6/19/2011 2:10:01 AM TS60808 notice raid.rg.scrub.summary.pl Scrub found 0 parity ...
Get-NaSystemLog can filter on several criteria, such as event severity, event source, and date ranges. This example lists errors from the past week.

PS C:\Toolkit\1.5.0> Get-NaSystemLog -MessageLog -Severity error -StartTime ([DateTime]::Today.AddDays(-7))

<table>
<thead>
<tr>
<th>TimeStampDT</th>
<th>Source</th>
<th>Severity</th>
<th>EventType</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/21/2011</td>
<td>auth.dc.trace.DCConnection.errorMsg</td>
<td>error</td>
<td>AUTH: Domain Controller error:...</td>
<td></td>
</tr>
</tbody>
</table>

Get-NaSystemLog can even monitor the logs for changes, similar to the unix 'tail -f' command. This example will detect and display error messages as Data ONTAP writes them to the messages log file.

PS C:\Toolkit\1.5.0> Get-NaSystemLog -MessageLog -Severity error -Follow

Disk signature manipulation
A cloned LUN in a Windows cluster may remain offline because its disk signature matches that of its parent, so it is critical to update the disk signature of the clone before mapping it to the cluster hosts.

A new cmdlet, Get-NaLunSignature, will display the signature of both MBR and GPT disks.

PS C:\Toolkit\1.5.0> Get-NaLunSignature /vol/vol2/lun2

Guid
----
d3bladee-4c19-455f-99b5-edd660ef9ad1

Another new cmdlet, Set-NaLunSignature, will generate and write a new disk signature of the correct type.

PS C:\Toolkit\1.5.0> Set-NaLunSignature /vol/vol2/lun2

Write LUN signature
Are you sure you wish to write the lun signature?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): Y

PS C:\Toolkit\1.5.0> Set-NaLunSignature /vol/vol2/lun2

Guid
----
31bfc207-4dfe-4855-bfa9-1b19485fcf5

Note that these two cmdlets are not Windows-specific and may be used for LUNs formatted as either MBR or GPT disks for other operating systems.

Faster host disk & VM discovery
Both Get-NaHostDisk and Get-NaHyperV perform significantly better in this release, especially in Windows clusters.

Using Toolkit 1.4, a Hyper-V cluster with 1000 virtual machines required 35 minutes to enumerate:

PS C:\Toolkit\1.4.0> Measure-Command { $vms = Get-NaHyperV }
TotalSeconds : 2134.7455097
TotalMilliseconds : 2134745.5097
PS C:\Toolkit\1.4.0> $vms.Count
1000

Toolkit 1.5 handles the same task in 25 seconds.

PS C:\Toolkit\1.5.0> Measure-Command { $vms = Get-NaHyperV }

Days : 0
Hours : 0
Minutes : 0
Seconds : 25
Milliseconds : 249
Ticks : 252491256
TotalDays : 0.00029223525
TotalHours : 0.007013646
TotalMinutes : 0.42081876
TotalSeconds : 25.2491256
TotalMilliseconds : 25249.1256

Vfiler DR
Five new cmdlets in the 'vfiler' category support the vfiler disaster recovery (DR) feature of Data ONTAP.

SnapLock
Toolkit 1.5 adds 11 cmdlets that support the SnapLock feature of Data ONTAP.

Secure password handling
Several cmdlets that accept cleartext passwords now also accept PSCredential objects. These include:

- Add-NaIscsiInitiatorAuth
- Get-NaCifsOrganizationalUnit
- Get-NaCifsSite
- New-NaCifsPasswordFile
- New-NaUser
- Set-NaCifs
- Set-NaIscsiInitiatorAuthChap
- Set-NaIscsiInitiatorDefaultAuth
- Set-NaUserPassword
- Set-NaVfilerPassword
- Test-NaCifsName

Other enhancements
In accordance with PowerShell best practices, Toolkit cmdlets no longer emit objects if the -WhatIf switch is specified. Similarly, Toolkit cmdlets no longer emit objects if an error occurs while processing an input record.

Remove-NaVol now includes a -Force switch, which is needed if a non-default vfiler has storage on the volume.

Invoke-NaSysstat now includes consistency point (CP) details in its WAFL output info. Note that different versions of Data ONTAP report different CP types, so Invoke-NaSysstat also reports the CP type labels in its output.

A few Toolkit users reported cmdlet timeout issues, so the default timeout for Toolkit cmdlets was increased to 60 seconds. The timeout may still be adjusted by modifying the relevant field on the NaController object returned by Connect-NaController.
New cmdlets
These are the new cmdlets in Data ONTAP PowerShell Toolkit 1.5:

- Add-NaHostIscsiTargetPortal
- Connect-NaHostIscsiTarget
- Disconnect-NaHostIscsiTarget
- Get-NaHostIscsiAdapter
- Get-NaHostIscsiSession
- Get-NaHostIscsiTarget
- Get-NaHostFcAdapter
- Get-NaLunSignature
- Set-NaLunSignature
- Get-NaSnaplockComplianceClock
- Get-NaSnaplockLog
- Get-NaSnaplockLogVolume
- Get-NaSnaplockOption
- Get-NaSnaplockRetentionTime
- New-NaSnaplockLog
- Remove-NaSnaplockFile
- Set-NaSnaplockComplianceClock
- Set-NaSnaplockLogVolume
- Set-NaSnaplockOption
- Set-NaSnaplockRetentionTime
- Get-NaSystemLog
- Get-NaEfficiency
- Get-NaVfilerDr
- Invoke-NaVfilerDr
- Invoke-NaVfilerDrResync
- New-NaVfilerDr
- Remove-NaVfilerDr

Issues fixed in Toolkit 1.5

- Set-NaSis, Start-NaSis, and Stop-NaSis will now bind the path parameter by property name.
- Set-NaVfilerPassword will now create the root user if it does not already exist.
- Get-NaHostDisk and Get-NaHyperV could fail if one or more host disks (LUNs) were offline.
Data ONTAP PowerShell Toolkit 1.4

Sysstat

A common Toolkit feature request has been a way to invoke Data ONTAP's sysstat CLI command. Toolkit 1.4 adds a cmdlet, `Invoke-NaSysstat`, which uses the performance APIs to return continuously updating metrics.

By default, this cmdlet collects data every 15 seconds and writes its output in object form to the PowerShell pipeline. The Toolkit’s data formatters present the performance data in tabular form.

```
PS C:\Toolkit\1.4.0> Invoke-NaSysstat

<table>
<thead>
<tr>
<th>CPU</th>
<th>NFS</th>
<th>CIFS</th>
<th>HTTP</th>
<th>FCP</th>
<th>iSCSI</th>
<th>RdOps</th>
<th>WrOps</th>
<th>TotOps</th>
<th>NetRcvd</th>
<th>NetSent</th>
<th>DiskRd</th>
<th>DiskWr</th>
<th>RdLat</th>
<th>WrLat</th>
<th>TotLat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>16 KB</td>
<td>887 B</td>
<td>34 KB</td>
<td>237 KB</td>
<td>25 KB</td>
<td>17 KB</td>
<td>245 KB</td>
<td>0.0</td>
</tr>
<tr>
<td>1%</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>43</td>
<td>30 KB</td>
<td>25 KB</td>
<td>17 KB</td>
<td>245 KB</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>1%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5 KB</td>
<td>2 KB</td>
<td>29 KB</td>
<td>316 KB</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>1%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>15 KB</td>
<td>751 B</td>
<td>7 KB</td>
<td>133 KB</td>
<td>0.0</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>1%</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>42</td>
<td>28 KB</td>
<td>25 KB</td>
<td>48 KB</td>
<td>390 KB</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

In the unformatted output, data sizes are presented in bytes, data rates are presented in bytes per second, and latencies are presented as TimeSpan objects. In the tabular views, data sizes and rates are presented with units, and latencies are presented in milliseconds.

`Invoke-NaSysstat` also monitors other objects such as volumes, disks and network adapters, as well as the various protocols. Furthermore, multiple objects of the same type may be monitored together. For example, monitor two volumes like this:

```
PS C:\Toolkit\1.4.0> Invoke-NaSysstat -Volume vol2,opalis

<table>
<thead>
<tr>
<th>Read</th>
<th>Written</th>
<th>RdOps</th>
<th>WrOps</th>
<th>TotOps</th>
<th>RdLat</th>
<th>WrLat</th>
<th>TotLat</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 MB</td>
<td>17 KB</td>
<td>49</td>
<td>2</td>
<td>53</td>
<td>3.1</td>
<td>0.1</td>
<td>2.9</td>
<td>opalis</td>
</tr>
<tr>
<td>166 KB</td>
<td>143 KB</td>
<td>5</td>
<td>4</td>
<td>13</td>
<td>3.5</td>
<td>0.5</td>
<td>1.7</td>
<td>vol2</td>
</tr>
<tr>
<td>3 MB</td>
<td>46 KB</td>
<td>170</td>
<td>7</td>
<td>178</td>
<td>60.4</td>
<td>0.1</td>
<td>58.0</td>
<td>opalis</td>
</tr>
<tr>
<td>0</td>
<td>819 B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>vol2</td>
</tr>
<tr>
<td>12 MB</td>
<td>346 KB</td>
<td>295</td>
<td>23</td>
<td>318</td>
<td>85.1</td>
<td>0.1</td>
<td>79.0</td>
<td>opalis</td>
</tr>
<tr>
<td>8 MB</td>
<td>102 B</td>
<td>144</td>
<td>0</td>
<td>145</td>
<td>0.3</td>
<td>0.1</td>
<td>0.3</td>
<td>vol2</td>
</tr>
<tr>
<td>4 MB</td>
<td>139 KB</td>
<td>98</td>
<td>22</td>
<td>120</td>
<td>46.5</td>
<td>0.1</td>
<td>37.0</td>
<td>opalis</td>
</tr>
<tr>
<td>3 MB</td>
<td>102 B</td>
<td>50</td>
<td>0</td>
<td>50</td>
<td>0.3</td>
<td>0.1</td>
<td>0.3</td>
<td>vol2</td>
</tr>
<tr>
<td>2 MB</td>
<td>201 KB</td>
<td>88</td>
<td>35</td>
<td>123</td>
<td>15.2</td>
<td>0.3</td>
<td>11.0</td>
<td>opalis</td>
</tr>
<tr>
<td>443 KB</td>
<td>2 MB</td>
<td>13</td>
<td>34</td>
<td>48</td>
<td>5.3</td>
<td>0.5</td>
<td>1.8</td>
<td>vol2</td>
</tr>
<tr>
<td>6 MB</td>
<td>471 KB</td>
<td>175</td>
<td>37</td>
<td>212</td>
<td>50.9</td>
<td>0.5</td>
<td>41.9</td>
<td>opalis</td>
</tr>
<tr>
<td>77 KB</td>
<td>164 KB</td>
<td>4</td>
<td>10</td>
<td>14</td>
<td>2.7</td>
<td>0.2</td>
<td>0.9</td>
<td>vol2</td>
</tr>
</tbody>
</table>

`Invoke-NaSysstat` also accepts wildcards. For example, monitor all drives on a loop like this:

```
PS C:\Toolkit\1.4.0> Invoke-NaSysstat -Disk 0a.*

<table>
<thead>
<tr>
<th>Busy</th>
<th>DiskRd</th>
<th>DiskWr</th>
<th>RdOps</th>
<th>WrOps</th>
<th>TotOps</th>
<th>QueueOps</th>
<th>PendOps</th>
<th>RdLat</th>
<th>WrLat</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>1 KB</td>
<td>20 KB</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>32.4</td>
<td>1.0</td>
<td>0a.16</td>
</tr>
<tr>
<td>2%</td>
<td>1 KB</td>
<td>21 KB</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>47.6</td>
<td>0.9</td>
<td>0a.17</td>
</tr>
<tr>
<td>0%</td>
<td>546 B</td>
<td>15 KB</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>11.5</td>
<td>1.5</td>
<td>0a.18</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0a.19</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0a.20</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0a.21</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0a.22</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0a.23</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0a.24</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0a.25</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0a.26</td>
</tr>
<tr>
<td>11%</td>
<td>989 KB</td>
<td>116 KB</td>
<td>18</td>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1.1</td>
<td>1.2</td>
<td>0a.27</td>
</tr>
<tr>
<td>1%</td>
<td>0</td>
<td>116 KB</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.8</td>
<td>0a.28</td>
</tr>
<tr>
<td>1%</td>
<td>0</td>
<td>118 KB</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>1.0</td>
<td>0a.29</td>
</tr>
</tbody>
</table>
```
Invoke-NaSysstat is optimized not only for continuous monitoring in tabular format, but also for use in scripts via hashtable output. In a script where continuous monitoring of the pipeline isn’t as useful, specify -Hashtable as well as a -Count value of 1. Then index the hashtable by object name:

```powershell
PS C:\Toolkit\1.4.0> $diskPerf = Invoke-NaSysstat -Disk 0a.* -Count 1 -Hashtable

PS C:\Toolkit\1.4.0> $diskPerf.Keys
0a.20
0a.21
0a.22
0a.23
0a.18
0a.19
0a.28
0a.29
0a.24
0a.16
0a.17
0a.25
0a.26
0a.27

PS C:\Toolkit\1.4.0> $diskPerf["0a.20"]

Busy  DiskRd  DiskWr  RdOps  WrOps  TotOps  QueueOps  PendOps  RdLat  WrLat  Disk
----  ------  ------  ------  ------  ------  --------  ------  -----  -----  ----
0%    819 B   0       0       0       0        0        0      1.7    0.0    0a.20

Use the -AllProtocols switch to collect more system-level performance data all at once:

```powershell
PS C:\Toolkit\1.4.0> $allPerf = Invoke-NaSysstat -AllProtocols -Count 1 -Hashtable

PS C:\Toolkit\1.4.0> $allPerf

Name                           Value
----                           ----- 
nfsv3                          DataONTAP.PowerShell.SDK.Cmdlets.Perf.Nfs3PerfInfo
wafl                           DataONTAP.PowerShell.SDK.Cmdlets.Perf.WaflPerfInfo
system                         DataONTAP.PowerShell.SDK.Cmdlets.Perf.SystemPerfInfo
cifs                           DataONTAP.PowerShell.SDK.Cmdlets.Perf.CifsPerfInfo
iscsi                          DataONTAP.PowerShell.SDK.Cmdlets.Perf.IscsiPerfInfo
fcp                             DataONTAP.PowerShell.SDK.Cmdlets.Perf.FcpPerfInfo

PS C:\Toolkit\1.4.0> $allPerf["system"]

CPU  NFS  CIFS  HTTP  FCP  ISCSI  RdOps  WrOps  TotOps  NetRcvd  NetSent  DiskRd  DiskWr  RdLat  WrLat  TotLat
---- ----- ----- ----- ----  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
1%    0     0     0     0     0      0      0      0      956 B   4 KB    5 KB    85 KB   0.0    0.0    0.1

PS C:\Toolkit\1.4.0> $allPerf["wafl"]

CpCount CpTime                CacheAge
------ ------                -------
1 00:00:00.1120000 02:36:28.1750000

PS C:\Toolkit\1.4.0> $allPerf["iscsi"]

Read  Written  TotOps  TotLat
------  ------  ------  -----
3 KB    0      0      0.0

Invoke-NaSysstat totals or averages data over the specified interval. This may be changed using the -SampleIntervalSeconds parameter.
LUN-to-host correlation

Toolkit 1.4 adds a cmdlet, `Get-NaHostDisk`, which enumerates all disks on the local host that correspond to Data ONTAP LUNs. Both host- and controller-side information are presented. The cmdlet supports Windows Server 2003 and later.

```
PS C:\Toolkit\1.4.0> Get-NaHostDisk
Disk       Size     ControllerPath
    ----     ------     ------------
        4   20.0 GB   dunn:/vol/vol2/lun2
        5   20.0 GB   dunn:/vol/vol2/lun1
        1   50.0 GB   benson:/vol/vol1/testlun
        7   20.0 GB   dunn:/vol/vol2/lun5
        9   34.0 GB   dunn:/vol/vol2/lun7b
       10   20.0 GB   dunn:/vol/vol2/lun6
        3   1.0 GB    benson:/vol/vol1/testlun
        6   49.9 GB   dunn:/vol/opalis/opalisVM
       27   20.0 GB   dunn:/vol/vol2/lun7b
        8   34.0 GB   dunn:/vol/opalis/w2k3OpalisVM
```

More information about the host disks may be obtained from Windows using the PowerShell WMI accelerator:

```
PS C:\Toolkit\1.4.0> $diskWmiPath = (Get-NaHostDisk 4).DiskWmiPath
PS C:\Toolkit\1.4.0> [wmi] $diskWmiPath
Partitions : 2
DeviceID   : \\.\PHYSICALDRIVE4
Model      : NETAPP LUN SCSI Disk Device
Size       : 21476206080
Caption    : NETAPP LUN SCSI Disk Device
```

This information is also useful as a starting point for subsequent WMI queries.
MinBlockSize : 
NumberOfMediaSupported : 63
SectorsPerTrack : 63
Size : 2147620680
TotalCylinders : 2611
TotalHeads : 255
TotalSectors : 41945715
TotalTracks : 665805
TracksPerCylinder : 255
__GENUS : 2
__CLASS : Win32_DiskDrive
__SUPERCLASS : CIM_DiskDrive
__DYNASTY : CIM_ManagedSystemElement
__PROPERTY_COUNT : 51
__DERIVATION : {CIM_DiskDrive, CIM_MediaAccessDevice, CIM_LogicalDevice, CIM_LogicalElement...}
__SERVER : X3550RRE7
__NAMESPACE : root\cimv2
__PROPERTY_COUNT : 51
__DERIVATION : {CIM_DiskDrive, CIM_MediaAccessDevice, CIM_LogicalDevice, CIM_LogicalElement...}
__SERVER : X3550RRE7
__NAMESPACE : root\cimv2
__PROPERTY_COUNT : 51
__DERIVATION : {CIM_DiskDrive, CIM_MediaAccessDevice, CIM_LogicalDevice, CIM_LogicalElement...}
__SERVER : X3550RRE7
__NAMESPACE : root\cimv2
Availability : True
Capabilities : {3, 4}
CapabilityDescriptions : {Random Access, Supports Writing}
Caption : NETAPP LUN SCSI Disk Device
CompressionMethod : 
CreationClassName : Win32_DiskDrive
Description : Disk drive
ErrorCleared : True
ErrorDescription : 
ErrorMethodology : 
FirmwareRevision : 7350
Manufacturer : (Standard disk drives)
MediaLoaded : True
MediaType : Fixed hard disk media
Model : NETAPP LUN SCSI Disk Device
Name : \\.
PNPDeviceID : \SCSI\DISK\VEN_NETAPP\PROD_LUN\1&1C121344&0&000100
PowerManagementCapabilities : 
PowerManagementSupported : True
SCSIBus : 0
SCSILogicalUnit : 0
SCSIPort : 3
SCSITargetId : 1
SerialNumber : HnSolJNfoTZK
Signature : 
SystemCreationClassName : Win32_ComputerSystem
SystemName : X3550RRE7
Scope : System.Management.ManagementScope
Path : \\.
Options : System.Management.ObjectGetOptions
ClassPath : \\.
Properties : {Availability, BytesPerSector, Capabilities, CapabilityDescriptions...}
SystemProperties : {__GENUS, __CLASS, __SUPERCLASS, __DYNASTY...}
Qualifiers : {dynamic, Locale, provider, UUID}
Site : 
Container : 

Get-NaHostDisk is also fully aware of Windows Server 2008 failover clusters and cluster shared volumes:

```
PS C:\Toolkit\1.4.0> Get-NaHostDisk | ft ClusterResource, ClusterGroup, ClusterNode, ControllerPath -auto
+----------------+----------------+-----------------+-----------------+
| ClusterResource | ClusterGroup    | ClusterNode     | ControllerPath  |
|----------------+----------------+-----------------+-----------------+
| Cluster Disk VM0| VM0            | FUJIRX200RRE3   | FAS3070-SITEB:/vol/WSFCRRE/VM0 |
| MCB - WSFCRRE3 | Available Storage| FUJIRX200RRE3   | FAS3070-SITEB:/vol/WSFCRRE/WSFCRRE2 |
| MCB - WSFCRRE2 | Available Storage| FUJIRX200RRE3   | FAS3070-SITEB:/vol/WSFCRRE/WSFCRRE3 |
| Cluster Disk 3  | Available Storage| FUJIRX200RRE3   | fas2040rel:/vol/luns/wsfcrre/vmdisk0 |
| fas2040rel - CSV_1 | 2a2edcb4-1f03-4e81-9a0e-77d09337d248 | FUJIRX200RRE3 | fas2040rel:/vol/luns/WSFCRRE/CSV_1 |
| CSV0            | 7a1c8935-2112-44d3-ba5-e-cf9cecd045e3 | FUJIRX200RRE3 | FAS3070-SITEB:/vol/WSFCRRE/CSV0 |
| Cluster Disk 2  | Cluster Group  | FUJIRX200RRE3   | indy:/vol/WSFCRRE/quorum2 |
```
```
Updated Data ONTAP support

Newer Data ONTAP features, such as large aggregates, volume move, and copy offload are all available in Toolkit 1.4. Some cmdlet output types, such as those representing volumes and aggregates, have been updated to include more information from newer Data ONTAP releases; this should be a non-breaking change.

New cmdlet categories

Toolkit 1.4 adds support for these API categories:

- copyoffload
- flash
- fpolicy
- lock
- rsh

Improved error handling

Most pipeline-terminating errors have been replaced with non-terminating errors. This enables better support of PowerShell's -ErrorAction and $ErrorActionPreference mechanisms, which only apply to non-terminating errors. For example, Toolkit 1.3 returned this with DataONTAP 8.0.1:

```
PS C:\Toolkit\1.3.0> Get-NaNetActiveConfig -ErrorAction SilentlyContinue
Get-NaNetActiveConfig : zapi_net_config_get_active is not implemented in this release
At line:1 char:22
+ Get-NaNetActiveConfig <<<< -ErrorAction SilentlyContinue
   + CategoryInfo          : InvalidOperation: (fas2040rre1:NaController) [Get-NaNetActiveConfig],
   + FullyQualifiedErrorId : ApiException,DataONTAP.PowerShell.SDK.Cmdlets.Net.GetNaNetActiveConfig

 Toolkit 1.4 is better behaved:

PS C:\Toolkit\1.4.0> Get-NaNetActiveConfig -ErrorAction SilentlyContinue
PS C:\Toolkit\1.4.0>
```

Simpler quota cmdlets

Additional parameter sets abstract some of the API parameters, making these cmdlets easier to use. For example, to find a tree quota, just specify a path:

```
PS C:\Toolkit\1.4.0> Get-NaQuota -Path /vol/vol1
Type               : tree
Target             : (*)
Volume             : vol1
Qtree              :
DiskLimit          : 5368709120
FileLimit          : -
PerformUserMapping : False
QuotaError         :
SoftDiskLimit      : -
SoftFileLimit      : -
Threshold          : -
```

These cmdlets include the new parameter sets:

- Add-NaQuota
- Get-NaQuota
- Remove-NaQuota
- Set-NaQuota
Improved snapshot cmdlets
Snapshot objects have two new fields, TargetName and TargetType. This allows for better pipeline support, such as:

PS C:\Toolkit\1.4.0> Get-NaSnapshot vol0 hourly.* | Remove-NaSnapshot

Delete snapshot
Are you sure you want to delete snapshot hourly.0 from volume vol0?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): a

PS C:\Toolkit\1.4.0>

Furthermore, more of the snapshot cmdlets now work with aggregate snapshots. These include:

- Get-NaSnapshot
- Get-NaSnapshotRes
- New-NaSnapshot
- Remove-NaSnapshot
- Rename-NaSnapshot
- Set-NaSnapshotRes

Improved cluster support in Get-NaHyperV
When run on any host in a Windows Server 2008 failover cluster, Get-NaHyperV now lists all virtual machines running in the cluster along with all of their storage resources supplied by Data ONTAP controllers.

PS C:\Toolkit\1.4.0> Get-NaHyperV
Name       Storage
---- --------
BEAMDEV2K8 {C:\ClusterStorage\Volume1\BEAMDEV2K8\BEAMDEV2K8.vhd}
VM0        {F:\VMO\VM0.vhd}
WSFCRRE2   {C:\ClusterStorage\Volume1\WSFCRRE2\Virtual Hard Disks\WSFCRRE2.vhd}
WSFCRRE3   {C:\ClusterStorage\Volume1\WSFCRRE3\Virtual Hard Disks\WSFCRRE3.vhd}
NICHOLSONDEV2K8 {C:\ClusterStorage\Volume1\NICHOLSONDEV2K8\NICHOLSONDEV2K8_2CB75BE7-534F-41D4-8062-07...}

PS C:\Toolkit\1.4.0> Get-NaHyperV | select -ExpandProperty Storage

Disk  Size ControllerPath
----  ----  ------------------
1      151.0 GB fas2040rre1:/vol/luns/wsfcrrs/csv_1
5      21.0 GB FAS3070-SITEB:/vol/WSFCRRE/VM0
1      151.0 GB fas2040rre1:/vol/luns/WSFCRRE/csv_1
1      151.0 GB fas2040rre1:/vol/luns/WSFCRRE/csv_1
1      151.0 GB fas2040rre1:/vol/luns/WSFCRRE/csv_1

PS C:\Toolkit\1.4.0> Get-NaHyperV BEAMDEV2K8 | select -ExpandProperty Storage | Format-List *
ControllerAddresses : {192.168.0.8, 10.61.169.28}
ControllerLunPath   : /vol/luns/2099A0D010
ControllerVolumeName: luns

API list cmdlets

Get-NaSystemApi lists the APIs on a controller.

PS C:\Toolkit\1.4.0> Get-NaSystemApi | select Api

<table>
<thead>
<tr>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggr-add</td>
</tr>
<tr>
<td>aggr-check</td>
</tr>
<tr>
<td>aggr-create</td>
</tr>
<tr>
<td>aggr-destroy</td>
</tr>
<tr>
<td>aggr-get-filer-info</td>
</tr>
<tr>
<td>aggr-get-root-name</td>
</tr>
<tr>
<td>aggr-list-info</td>
</tr>
<tr>
<td>aggr-mediascrub-list-info</td>
</tr>
<tr>
<td>aggr-mirror</td>
</tr>
<tr>
<td>aggr-offline</td>
</tr>
<tr>
<td>aggr-online</td>
</tr>
<tr>
<td>aggr-options-list-info</td>
</tr>
</tbody>
</table>

Get-NaHelpUnsupported correlates the supported API list with the Toolkit cmdlets and returns a list of Toolkit cmdlets that will not work on the current controller due to missing APIs. For example, a controller running Data ONTAP 7.3.5 will not have the newer volume move functions:

PS C:\Toolkit\1.4.0> Get-NaHelpUnsupported

<table>
<thead>
<tr>
<th>Name</th>
<th>Api</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-NaVolMove</td>
<td>{volume-move-status}</td>
<td>volume</td>
</tr>
<tr>
<td>Invoke-NaVolMoveCutover</td>
<td>{volume-move-cutover}</td>
<td>volume</td>
</tr>
<tr>
<td>Resume-NaVolMove</td>
<td>{volume-move-resume}</td>
<td>volume</td>
</tr>
<tr>
<td>Start-NaVolMove</td>
<td>{volume-move-start}</td>
<td>volume</td>
</tr>
<tr>
<td>Stop-NaVolMove</td>
<td>{volume-move-abort}</td>
<td>volume</td>
</tr>
<tr>
<td>Suspend-NaVolMove</td>
<td>{volume-move-pause}</td>
<td>volume</td>
</tr>
</tbody>
</table>

Raw API invocation

Toolkit 1.4 provides a cmdlet, Invoke-NaSystemApi, to provide low-level access to the Data ONTAP API interface. Input is either an XML-formatted string or a .NET XmlDocument object. The output is an XmlDocument that is easily traversed in PowerShell. More details and documentation are available in the NetApp Manageability SDK.

PS C:\Toolkit\1.4.0> $xml = Invoke-NaSystemApi "<system-get-version/>"
PS C:\Toolkit\1.4.0> $xml.results.version

NetApp Release 7.3.5: Mon Nov 22 21:32:44 PST 2010

Hashed options

For easier use in scripts, the options cmdlets (Get-NaAggrOption, Get-NaVolOption, Get-NaOption) optionally return their output in a hashtable.

PS C:\Toolkit\1.4.0> $options = Get-NaOption *http* -Hashtable
PS C:\Toolkit\1.4.0> $options

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>httpd.admin.top-page.authen...</td>
<td>on</td>
</tr>
<tr>
<td>httpd.admin.enable</td>
<td>on</td>
</tr>
<tr>
<td>httpd.rootdir</td>
<td>/vol/vol0/home/http</td>
</tr>
<tr>
<td>httpd.admin.ssl.enable</td>
<td>on</td>
</tr>
<tr>
<td>httpd.enable</td>
<td>off</td>
</tr>
<tr>
<td>httpd.ipv6.enable</td>
<td>on</td>
</tr>
</tbody>
</table>
Provider enhancements
In Toolkit 1.4, the DataONTAP PowerShell provider leverages the copy offload feature, if available, for copying or moving files. The provider also supports aggregate paths.

Table column header aliases
In Toolkit 1.4, the table column headers match up directly with fields of the output object. For example, with Toolkit 1.3, the aggregate table looks like this:

```
PS C:\Toolkit\1.3.0> Get-NaAggr aggr0
Name            State   TotalSize  Used  Available Disks RaidType RaidSize MirrorStatus     FilesUsed FilesTotal
----            ------   --------  ----  -------- ----  ------  -------  ---------------------- ---------- ----------
aggr0           online   2.8 TB   38%   1.7 TB  6  raid_dp  14  unmirrored              96        31k
```

Selecting just the Name, TotalSize, Available, and Disks fields leads to this:
```
PS C:\Toolkit\1.3.0> Get-NaAggr aggr0 | Select Name,TotalSize,Available,Disks
Name                          TotalSize                     Available                     Disks
----                          --------                      --------                      ----
aggr0                         3039371522048                 1945910370304                             6
```

TotalSize, Available, and Disks are not actually fields of the output object, so they appear empty when referred to by name. With Toolkit 1.4, the column headers (which are often shortened to fit in the limited space available) are aliased to their source fields:
```
PS C:\Toolkit\1.4.0> Get-NaAggr aggr0 | Select Name,TotalSize,Available,Disks
Name                          TotalSize                     Available                     Disks
----                          --------                      --------                      ----
aggr0                         3039371522048                 1945910370304                             6
```

New cmdlets
These are the new cmdlets in Data ONTAP PowerShell Toolkit 1.4:

- Get-NaCopyOffload
- Start-NaCopyOffload
- Stop-NaCopyOffload
- Get-NaFlashDevice
- Get-NaFlashProfile
- Get-NaFlashThreshold
- Add-NaFpolicyExtension
- Disable-NaFpolicy
- Enable-NaFpolicy
Issues fixed in Toolkit 1.4

- Added workaround for issue where system-cli API sent to a vfiler can cause Data ONTAP to panic.
- Fixed issue with wildcard patterns in cmdlets that accept arrays.
- Added workaround for incorrect date/time values reported by Data ONTAP.
Data ONTAP PowerShell Toolkit 1.3

SSH invocation

When writing scripts to manage Data ONTAP, there are sometimes cases where there is not an API (hence no cmdlet) to accomplish a certain task that is available via the controller’s command-line interface. Unfortunately there isn’t a publicly supported API for invoking a CLI command, but Data ONTAP fully supports CLI commands issued via SSH connections. So Toolkit 1.3 offers a cmdlet, **Invoke-NaSsh**, to fill this role. The input and output are just unformatted strings, like this:

```
Invoke-NaSsh aggr status
```

<table>
<thead>
<tr>
<th>Aggr State</th>
<th>Status</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggr0 online</td>
<td>raid_dp, aggr</td>
<td>root</td>
</tr>
<tr>
<td></td>
<td>redirect</td>
<td></td>
</tr>
<tr>
<td>aggr1 online</td>
<td>raid_dp, aggr</td>
<td>snapshot_autodelete=off</td>
</tr>
<tr>
<td>aggr2 online</td>
<td>raid_dp, aggr</td>
<td></td>
</tr>
</tbody>
</table>

**Invoke-NaSsh** declares the `ValueFromRemainingArguments` attribute, so the CLI command need not use quotation marks unless there are hyphens that can confuse the PowerShell parser:

```
Invoke-NaSsh "sysconfig -V"
```

```
Volume State       Status       Options
--------------     ----------     ----------
volume aggr0 (1 RAID group):
  group 0: 3 disks
volume aggr1 (1 RAID group):
  group 0: 8 disks
volume aggr2 (1 RAID group):
  group 0: 3 disks
```

**Invoke-NaSsh** uses SSH connections independent of the normal HTTP or RPC connections used by the Toolkit, so the cmdlet must have valid controller credentials. If you are using HTTP or HTTPS connections, the Toolkit already has credentials and uses those for SSH as well. If you are using RPC, credentials must be supplied independently, either explicitly using the `-Credential` parameter or implicitly using the new credentials cache mechanism.

You can use PowerShell’s alias feature to simplify **Invoke-NaSsh** even further:

```
New-Alias ssh Invoke-NaSsh
```

```
Volume State       Status       Options
--------------     ----------     ----------
testvol3 online    raid_dp, flex| create_ucode=on,
sis               | convert_ucode=on
SC_3_3_0 online   raid_dp, flex| create_ucode=on,
sis               | convert_ucode=on
vol10 online      raid_dp, flex| root, create_ucode=on,
sis               | convert_ucode=on
            redirect     | active_redirect
```

And although it isn’t something for which we can claim unbounded support, **Invoke-NaSsh** isn’t artificially limited to Data ONTAP connections. SSH commands may be directed elsewhere, such as a NetApp DataFabric Manager host, by specifying the `-Name` and `-Credential` parameters:

```
Invoke-NaSsh -Name 10.60.0.10 -Credential Administrator whoami
```

```
Credentials cache

Toolkit 1.3 adds a secure credentials cache, which is supported by Connect-NaController and Invoke-NaSsh. This feature allows HTTP/HTTPS users to use the Toolkit with nearly the same convenience as RPC users. To add a credential, just connect to a controller using HTTP or HTTPS and issue Add-NaCredential:

```
PS C:\Users\cknight> Connect-NaController dunn -Credential root
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Ontapi</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>dunn</td>
<td>10.61.167.60</td>
<td>1.12</td>
<td>NetApp Release 7.3.3: Thu Mar 11 22:29:52 PST 2010</td>
</tr>
</tbody>
</table>

```
PS C:\Users\cknight> Add-NaCredential
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Credential</th>
<th>HostUser</th>
</tr>
</thead>
<tbody>
<tr>
<td>dunn</td>
<td>System.Management.Automation.PSCredential</td>
<td>SYSTEM</td>
</tr>
</tbody>
</table>

Once in the cache, those credentials are always used and cause Connect-NaController to prefer HTTP/HTTPS unless explicitly overridden, such as with the -Credential or -RPC parameters.

The credentials cache uses Windows’ built-in data protection facility. By default, credentials are encrypted using the context of the current user, meaning that only the same user can retrieve those credentials. Multiple users may each save their own credentials on the same Windows host. By specifying the -SystemScope parameter to Add-NaCredential, the user may save credentials such that any user or process on the local Windows system can retrieve the credentials. Saving credentials with system scope is useful for scripts that run under a different user account, but that option should only be used on a Windows host with limited access since any logged-in user may access the credentials.

The credentials cache is designed to interoperate with the cmdlets shipped with ApplianceWatch 2.1 product. Note, however, that ApplianceWatch only supports storing credentials with system scope.

Regardless of whether credentials are saved using user or system scope, they may only be decrypted on the Windows host on which they were saved. It is therefore not possible to copy the credentials cache file for use on a different host.

If the cache contains entries for a given controller for both the current user and the system, as in the following example, the entry with local user scope is used preferentially.

```
PS C:\Users\cknight> Get-NaCredential | ft -AutoSize
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Credential</th>
<th>HostUser</th>
</tr>
</thead>
<tbody>
<tr>
<td>dunn</td>
<td>System.Management.Automation.PSCredential</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>benson</td>
<td>System.Management.Automation.PSCredential</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>indy</td>
<td>System.Management.Automation.PSCredential</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>indy</td>
<td>System.Management.Automation.PSCredential</td>
<td>cknigt-w\cknight</td>
</tr>
<tr>
<td>10.61.169.28</td>
<td>System.Management.Automation.PSCredential</td>
<td>cknigt-w\cknight</td>
</tr>
<tr>
<td>10.61.169.29</td>
<td>System.Management.Automation.PSCredential</td>
<td>cknigt-w\cknight</td>
</tr>
<tr>
<td>dunn</td>
<td>System.Management.Automation.PSCredential</td>
<td>cknigt-w\cknight</td>
</tr>
</tbody>
</table>

The credentials cache may be emptied easily:

```
PS C:\Users\cknight> Get-NaCredential | Remove-NaCredential
```

FlexShare

Toolkit 1.3 adds eight cmdlets that support FlexShare, Data ONTAP’s volume-based quality-of-service mechanism. The cmdlet category for these is `priority`. 
PowerShell Provider

PowerShell providers offer access to various resources, such as a file system, registry, or certificate store. Toolkit 1.3 contains an experimental feature, a PowerShell provider for Data ONTAP. Using the Data ONTAP provider, files and directories in the Data ONTAP file system may be accessed using the same semantics as a local Windows disk. The provider does not require any specific protocol, such as CIFS, to be licensed. The provider relies on the ManageONTAP SDK, so I/O operations are slower than CIFS access and are primarily suited for administrative tasks.

The provider is automatically loaded when the DataONTAP module is imported:

```powershell
PS C:\Users\cknight> Import-Module DataONTAP
PS C:\Users\cknight> Get-PSProvider
Name                      Capabilities                                      Drives
----                      ----------------------------                  ----
WSMan                     Credentials                               {WSMan}
Alias                     ShouldProcess                            {Alias}
Environment               ShouldProcess                            {Env}
FileSystem                 Filter, ShouldProcess                      {C, D, E, I...}
Function                  ShouldProcess                            {Function}
Registry                  ShouldProcess, Transactions                 {HKLM, HKCU}
Variable                  ShouldProcess                            {Variable}
Certificate               ShouldProcess                            {cert}
DataONTAP                 ShouldProcess                            {}
```

To use the provider, start by connecting to a Data ONTAP controller:

```powershell
PS C:\Users\cknight> Connect-NaController dunn
Name          Address          Ontapi    Version
----          --------          ------    -----      
dunn          10.61.167.60    1.12      NetApp Release 7.3.3: Thu Mar 11 22:29:52 PST 2010
```

Next, create a PSDrive object:

```powershell
PS C:\Users\cknight> Mount-NaController
Name          Used (GB)     Free (GB) Provider Root
----          --------          ----      ----
dunn          DataONTAP     
```

Then, change the current working directory to the new drive:

```powershell
PS C:\Users\cknight> cd dunn:
PS dunn:\>
```

Most of the various file listing and manipulation commands work as expected. The `/etc` directory is a shortcut to the etc directory on the root volume. Keep in mind that PowerShell provider drives are only accessible within PowerShell.

```powershell
PS dunn:\> dir
Name          Type            Size      Created     Modified Owner Group   Perm Empty
----          ----            --------      --------      -------- ----    ---- ----
.            directory       4 KB
etc          directory       12 KB      8/12/2008    1/31/2011     0     0   1070
vol          directory       4 KB
```

How often have you wished you could do this?

```powershell
PS dunn:\> cat /etc/rc
#Auto-generated by zapi_net_config Fri Oct  1 14:52:04 EDT 2010
hostname dunn
ifconfig e0a `hostname`-e0a netmask 255.255.255.0 broadcast 10.61.167.255 flowcontrol full partner e0a up
ifconfig e0a alias 10.61.167.98 netmask 255.255.255.0
route add net default 10.61.167.1 1
routed on
options dns.domainname rtpre.testdomain
options dns.enable on
options nis.enable off
savecore
```
When transferring binary files to Data ONTAP, use the -Encoding and -ReadCount parameters as follows:

```
PS dunn:\> Get-Content C:\file.bin -Encoding byte -ReadCount 128kb | Set-Content C:\file.bin -Encoding byte
```

When transferring binary files from Data ONTAP, use the -Encoding parameter as follows:

```
PS dunn:\> Get-Content file.bin -Encoding byte | Set-Content C:\file.bin -Encoding byte
```

The Data ONTAP PowerShell Provider supports a few dynamic parameters that are patterned after the FileSystem provider:

- **Encoding**
  - Unknown. The encoding type is unknown or invalid. The data can be treated as binary.
  - String. Uses the encoding type for a string.
  - Unicode. Encodes in UTF-16 format using the little-endian byte order.
  - Byte. Encodes a set of characters into a sequence of bytes.
  - BigEndianUnicode. Encodes in UTF-16 format using the big-endian byte order.
  - UTF8. Encodes in UTF-8 format.
  - UTF7. Encodes in UTF-7 format.
  - ASCII. Uses the encoding for the ASCII (7-bit) character set. ASCII is the default.
  - Cmdlets Supported: Add-Content, Get-Content, Set-Content

- **Delimiter**
  - Specifies the line delimiter to use when writing the file. The default is "\n" (end of line).
  - Cmdlets Supported: Add-Content, Set-Content

Although we consider the provider to be experimental, significant changes are not anticipated. We invite users to let us know how they are using the provider and how we can make it better.

**ShouldProcess**

PowerShell cmdlets use a built-in mechanism called ShouldProcess to handle confirmations and the -WhatIf parameter. Previous releases of the Toolkit included ShouldProcess support mostly for cmdlets whose nature required a user confirmation. Toolkit 1.3 extends ShouldProcess support for all non-getter cmdlets. This allows extensive use of the -WhatIf parameter:

```
PS C:\Users\cknight> Set-NaVolLanguage vol0 C.UTF-8 -WhatIf
What if: Setting language for volume vol0 to C.UTF-8.
```

Almost all of the affected cmdlets are set to ConfirmImpact.Low, whereas PowerShell’s default confirmation level is Medium, so this change should have little or no impact on existing scripts.

**ToString**

The ToString() method has been overridden for all object types that are returned by Toolkit cmdlets, improving readability of Toolkit output. What previously appeared like this:

```
PS C:\Software\Toolkit\1.2.0> Get-NaSnmp

Contact     : Bill Smith
IsTrapEnabled : True
Location    : Main lab
Traphosts   : {x3550re8.rtprre.testdomain}
```

...is now a little more readable:

```
PS C:\Software\Toolkit\1.3.0> Get-NaSnmp

...
The obvious caveat is that it isn’t obvious whether one is looking at a mere string or what may be a much larger object.

When inspecting Toolkit output, don’t hesitate to use `Get-Member` to better understand what is available:

```
PS C:\Software\Toolkit\1.3.0> Get-NaSnmp | Get-Member
```

```
TypeName: DataONTAP.PowerShell.SDK.Cmdlets.Snmp.SnmpStatusInfo
```

```
Name               MemberType       Definition
----               ---------         -----------------------
Equals             Method           bool Equals(System.Object obj)
GetHashCode        Method           int GetHashCode()
GetType            Method           type GetType()
ToString           Method           string ToString()
Communities        Property         NetApp.Ontapi.Filer.Snmp.CommunityInfo[] Communities {get;set;}
Contact            Property         System.String Contact {get;set;}
IsTrapEnabled      Property         System.Nullable`1[System.Boolean, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7dd8a72f9] IsTrapEnabled System.String Contact {get;set;}
Location           Property         System.String Contact {get;set;}
```

### Sorting

Many getter cmdlets now sort their output by a nominal field. For example, what was previously:

```
PS C:\Software\Toolkit\1.2.0> Get-NaLun */vol2/*
```

```
Path                                      TotalSize Protocol     Online Mapped  Thin  Comment
----                                      ---------         --------  ------  ------  ------
/vol/vol2/{6e8dbad4-894b-413a-b9cc-3c...    20.0 GB windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/lun2clone                      2.0 GB windows_gpt   True   False   True 数据卷为VM3
/vol/vol2/testgtree/testlun              10.0 GB image        True   False   True
/vol/vol2/lun101                         2.3 GB image        True   False   True
/vol/vol2/lun1                           20.0 GB windows_gpt   True   True   True
/vol/vol2/lun3b                          2.0 GB windows_gpt   True   False   True 数据卷为VM3
/vol/vol2/lun7b                          2.0 GB windows_gpt   True   False   True 数据卷为VM3
/vol/vol2/lun5                           20.0 GB windows_gpt   True   True   True
/vol/vol2/lun4                           20.1 GB windows_gpt   True   True   True
/vol/vol2/lun3                           20.0 GB windows_gpt   True   True   True
/vol/vol2/quorum                        1.0 GB windows_gpt   True   True   True  Cluster quorum LUN
/vol/vol2/lun2                          20.0 GB windows_gpt   True   True   True
/vol/vol2/lun1                          20.0 GB windows_gpt   True   True   True
```

...is now automatically sorted by path:

```
PS C:\Software\Toolkit\1.3.0> Get-NaLun */vol2/*
```

```
Path                                      TotalSize Protocol     Online Mapped  Thin  Comment
----                                      ---------         --------  ------  ------  ------
/vol/vol2/{6e8dbad4-894b-413a-b9cc-3c...    20.0 GB Windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/lun1                           20.0 GB Windows_gpt   True   True   True
/vol/vol2/lun101                         2.3 GB image        True   True   True  Virtual machine 1
/vol/vol2/lun2                           20.0 GB Windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/lun3                           20.0 GB Windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/lun4                           20.1 GB Windows_gpt   True   True   True
/vol/vol2/lun6                           20.0 GB Windows_gpt   True   True   True
/vol/vol2/lun7b                          2.0 GB Windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/lun3b                          2.0 GB Windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/lun5                           20.0 GB Windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/lun7                           34.0 GB Windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/metadat                        1.0 GB Windows_gpt   True   True   True  Virtual machine 1
/vol/vol2/quorum                        1.0 GB Windows_gpt   True   True   True  Cluster quorum LUN
/vol/vol2/test                          10.0 GB Windows_2008 False   False   True  test
/vol/vol2/testqtree/testlun             10.0 GB image        True   False   True  test
```
Connection enhancements

`Connect-NaController` adds three new switch parameters (-RPC, -HTTP, -HTTPS) that allow more explicit control over which protocol is used by Toolkit connections. RPC is still the default if no credentials are available, but with the addition of the credentials cache, it is no longer true that RPC is always used if the -Credential parameter is not specified. The previously available parameters, -ForceSecure and -ForceUnsecure, are deprecated but still work via parameter aliases.

`Connect-NaController` also adds a -Timeout parameter that affects all Toolkit cmdlets that use the resulting connection. The parameter is specified in milliseconds and only affects HTTP/HTTPS connections. The default value is 10 seconds.

There were a few reports of Windows HTTP proxy settings preventing Toolkit connections. Toolkit 1.3 ignores HTTP proxy settings.

Parameter aliases

PowerShell cmdlets often support parameter aliases, but it is not obvious how to discover those. Toolkit 1.3 adds a cmdlet, `Get-NaHelp Alias`, which displays parameter aliases. It works with all cmdlets, not just those of the Toolkit.

```
PS C:\Software\Toolkit\1.3.0> Get-NaHelpAlias New-NaVol

Name                  Aliases
--------              -------
Name                  {}
Aggregate             {Aggr}
SnapLock              {}
LanguageCode          {Language, Lang}
SpaceReserve          {Reserve}
Controller            {Filer, Server}
Size                  {}
SnapLockType          {}
Verbose               {vb}
Debug                 {db}
ErrorAction           {ea}
WarningAction         {wa}
ErrorVariable         {ev}
WarningVariable       {wv}
OutVariable           {ov}
OutBuffer             {ob}
WhatIf                {wi}
Confirm               {cf}
```

New cmdlets

These are the new cmdlets in Data ONTAP PowerShell Toolkit 1.3:

- Disable-NaPriority
- Enable-NaPriority
- Get-NaPriority
- Get-NaPriorityDefault
- Get-NaPriorityVolume
- Set-NaPriority
- Set-NaPriorityDefault
- Set-NaPriorityVolume
- Set-NaQtree
- Add-NaCredential
- Get-NaCredential
- Remove-NaCredential
- Mount-NaController
- Dismount-NaController
- Invoke-NaSsh
- Get-NaHelpAlias

**Issues fixed in Toolkit 1.3**

- Enabled selection of whole-volume qtrees in `Get-NaQtree`.
- Toolkit changed to not use proxy for HTTP connections.
- Source snapshot is now accepted by `Start-NaSnapvaultSecTransfer` & `Start-NaSnapvaultPriRestoreTransfer`. 
Data ONTAP PowerShell Toolkit 1.2

Data formatters

The data formatters included in Toolkit 1.2 have no impact on script operation. Rather, they are designed to make the Toolkit more useful as a CLI tool by displaying common object types in tabular format with only the most useful fields included.

For example, earlier Toolkit releases might have behaved like this:

```powershell
PS C:\Users\cknight> Get-NaVol

DedupeEnabled : True
ChecksumStyle  : block
CloneChildren  :
CloneParent    :
ContainingAggregate : aggr1
DiskCount      : 8
FilesTotal     : 622580
FilesUsed      : 155
IsChecksumEnabled : True
IsInconsistent : False
IsSnaplock     : False
IsUnrecoverable: False
MirrorStatus   : unmirrored
Name           : testvol
PercentageUsed : 0
PlexCount      : 1
QuotaInit      : 0
RaidSize       : 16
RaidStatus     : raid_dp,sis
Reserve        : 0
ReserveRequired: 0
ReserveUsed    : 0
Sis            : NetApp.Ontapi.Filer.Volume.Sis
SizeAvailable  : 17175863296
SizeTotal      : 17179869184
SizeUsed       : 4005888
SnaplockType   :
SpaceReserve   : volume
SpaceReserveEnabled : True
State          : online
Type           : flex
Uuid           : ac4b29ba-3293-11df-87b2-00a9806c227
IsUnrecoverableSpecified : True
PercentageUsedSpecified : True
SpaceReserveEnabledSpecified : True
```

Whereas Toolkit 1.2 works like this:

```powershell
PS C:\Users\cknight> Get-NaVol

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>TotalSize</th>
<th>Used</th>
<th>Available</th>
<th>Dedupe</th>
<th>FilesUsed</th>
<th>FilesTotal</th>
<th>Aggregate</th>
</tr>
</thead>
<tbody>
<tr>
<td>testvol3</td>
<td>online</td>
<td>16.0 GB</td>
<td>0%</td>
<td>16.0 GB</td>
<td>True</td>
<td>103</td>
<td>623k</td>
<td>aggr1</td>
</tr>
<tr>
<td>SC_3_3_0</td>
<td>online</td>
<td>4.0 GB</td>
<td>4%</td>
<td>3.9 GB</td>
<td>True</td>
<td>596</td>
<td>156k aggr1</td>
<td></td>
</tr>
<tr>
<td>vol0</td>
<td>online</td>
<td>96.1 GB</td>
<td>16%</td>
<td>80.9 GB</td>
<td>True</td>
<td>83k</td>
<td>4M aggr0</td>
<td></td>
</tr>
<tr>
<td>vol1</td>
<td>online</td>
<td>16.0 GB</td>
<td>5%</td>
<td>15.2 GB</td>
<td>True</td>
<td>142</td>
<td>623k aggr1</td>
<td></td>
</tr>
<tr>
<td>vol2</td>
<td>online</td>
<td>180.0 GB</td>
<td>50%</td>
<td>89.3 GB</td>
<td>True</td>
<td>208</td>
<td>6M aggr1</td>
<td></td>
</tr>
<tr>
<td>testvol4</td>
<td>online</td>
<td>8.0 GB</td>
<td>0%</td>
<td>8.0 GB</td>
<td>True</td>
<td>136</td>
<td>311k aggr1</td>
<td></td>
</tr>
<tr>
<td>clinton_backup</td>
<td>online</td>
<td>24.0 GB</td>
<td>88%</td>
<td>2.8 GB</td>
<td>True</td>
<td>61k</td>
<td>934k aggr1</td>
<td></td>
</tr>
<tr>
<td>opalis</td>
<td>online</td>
<td>118.0 GB</td>
<td>100%</td>
<td>0</td>
<td>True</td>
<td>116</td>
<td>4M aggr2</td>
<td></td>
</tr>
<tr>
<td>AS_5_12_1</td>
<td>online</td>
<td>2.4 GB</td>
<td>1%</td>
<td>2.4 GB</td>
<td>True</td>
<td>1k</td>
<td>93k aggr1</td>
<td></td>
</tr>
<tr>
<td>vol1clone</td>
<td>online</td>
<td>16.0 GB</td>
<td>5%</td>
<td>15.2 GB</td>
<td>True</td>
<td>142</td>
<td>623k aggr1</td>
<td></td>
</tr>
<tr>
<td>vol2clone</td>
<td>online</td>
<td>180.0 GB</td>
<td>46%</td>
<td>97.3 GB</td>
<td>True</td>
<td>214</td>
<td>6M aggr1</td>
<td></td>
</tr>
<tr>
<td>testvol2</td>
<td>online</td>
<td>16.0 GB</td>
<td>0%</td>
<td>16.0 GB</td>
<td>True</td>
<td>146</td>
<td>623k aggr1</td>
<td></td>
</tr>
<tr>
<td>testvol</td>
<td>online</td>
<td>16.0 GB</td>
<td>0%</td>
<td>16.0 GB</td>
<td>True</td>
<td>155</td>
<td>623k aggr1</td>
<td></td>
</tr>
</tbody>
</table>
```

The data formatters are optimized for a 120-character display width. You can set the default shell width by modifying your PowerShell profile like this:
You can also customize the data formatters as needed by modifying the file DataONTAP.Format.ps1xml in the Toolkit module installation directory, i.e. C:\Windows\System32\WindowsPowerShell\v1.0\Modules\DataONTAP. If you edit the file, make sure to keep a backup copy of your changes in a separate location, as future Toolkit updates will overwrite the file.

The formatters also streamline list and wide views for many common object types. To see the unformatted objects, try something like: Get-NaVol | Format-List *

**Microsoft Hyper-V integration**

Toolkit 1.2 includes a new cmdlet, Get-NaHyperV, that enumerates all Hyper-V virtual machines running on the local host that utilize Data ONTAP storage resources. For each such virtual machine, the cmdlet lists all of the pertinent host and controller information that might be useful for scripting in virtualized environments. The cmdlet supports both VHD files and passthrough disks, as well as cluster shared volumes. For example:

```powershell
Get-NaHyperV | ft -AutoSize
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Guid</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpalisVM</td>
<td>68073F14-5B9A-44F2-923F-766432BE9A68 (C:\OpalisVM.vhd, Disk 2 50.01 GB Bus 0 Lun 1 Target 2)</td>
<td></td>
</tr>
<tr>
<td>W2k3OpalisVM</td>
<td>F4043F56-79A1-4FA3-8F71-77E2CEBCF0E7 (F:\W2k3OpalisVM.vhd)</td>
<td></td>
</tr>
</tbody>
</table>

```powershell
(Get-NaHyperV OpalisVM).Storage
```

<table>
<thead>
<tr>
<th>VmDiskResourceName</th>
<th>HostDrivePath</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:\OpalisVM.vhd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VmDiskResourceType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VHD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HostDrivePath</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0:\PHYSICALDRIVE6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HostVolume</td>
<td>?\Volume\4044589-4fd9-4c90-8495-f430945e7439\</td>
<td></td>
</tr>
<tr>
<td>HostVolumeIsCsv</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>ControllerName</td>
<td>dunn</td>
<td></td>
</tr>
<tr>
<td>ControllerIgroup</td>
<td>viaRPC.1991-05.com.microsoft:x3550rre7.rtprre.testdomain</td>
<td></td>
</tr>
<tr>
<td>ControllerAddress</td>
<td>10.61.167.60</td>
<td></td>
</tr>
<tr>
<td>ControllerLunPath</td>
<td>/vol/opalis/opalisVM</td>
<td></td>
</tr>
<tr>
<td>ControllerVolumeName</td>
<td>opalis</td>
<td></td>
</tr>
<tr>
<td>VmDiskResourceName</td>
<td>Disk 2 50.01 GB Bus 0 Lun 1 Target 2</td>
<td></td>
</tr>
<tr>
<td>VmDiskResourceType</td>
<td>Passthrough</td>
<td></td>
</tr>
<tr>
<td>HostDrivePath</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HostDrivePath</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HostDrivePath</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HostDrivePath</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HostVolume</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HostVolumeIsCsv</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>ControllerName</td>
<td>benson</td>
<td></td>
</tr>
<tr>
<td>ControllerIgroup</td>
<td>viaRPC.1991-05.com.microsoft:x3550rre7.rtprre.testdomain</td>
<td></td>
</tr>
<tr>
<td>ControllerAddress</td>
<td>10.61.167.61</td>
<td></td>
</tr>
<tr>
<td>ControllerLunPath</td>
<td>/vol/voll/testlun2</td>
<td></td>
</tr>
<tr>
<td>ControllerVolumeName</td>
<td>voll</td>
<td></td>
</tr>
</tbody>
</table>

Get-NaHyperV requires no credentials but requires the Hyper-V role to be installed. The values returned in the ControllerName or ControllerAddress fields may be passed to Connect-NaController for further storage queries or manipulation.

**Network configuration**

By popular request, Toolkit 1.2 adds twenty cmdlets for configuring network interfaces, VLANs, VIFs, routes and ipspaces. By design, all cmdlets that change the network settings modify both the active configuration and the persistent configuration stored in /etc/rc. This ensures that all changes are maintained across reboots. Non-network
items in /etc/rc are preserved by Data ONTAP. Most of the new network cmdlets require Data ONTAP 7.3.3 or later and aren’t yet supported in the Data ONTAP 8.x releases.

New cmdlets
These are the new cmdlets in Data ONTAP PowerShell Toolkit 1.2. Many of these require APIs that are new in Data ONTAP 7.3.3 or later and aren’t yet available in the Data ONTAP 8.x releases.

- Get-NaCifsShareAcl
- Remove-NaCifsShareAcl
- Set-NaCifsShareAcl
- Clear-NaLunPersistentReservation
- Get-NaLunPersistentReservation
- Set-NaLunDeviceId
- Get-NaLunSelect
- Set-NaLunSelect
- Add-NaNetRoute
- Add-NaNetVlan
- Get-NaNetActiveConfig
- Get-NaNetInterface
- Get-NaNetIpspace
- Get-NaNetPersistentConfig
- Get-NaNetRoute
- Get-NaNetVif
- Get-NaNetVlan
- New-NaNetIpspace
- New-NaNetVif
- Remove-NaNetIpspace
- Remove-NaNetRoute
- Remove-NaNetVif
- Remove-NaNetVlan
- Set-NaNetInterface
- Set-NaNetIpspace
- Set-NaNetPersistentConfig
- Set-NaNetVif
- Set-NaNetVlan
- Get-NaShelf
- Get-NaShelfEnvironment
- Set-NaShelfLight
- Start-NaShelfUpdate
- Get-NaSystemAvailableReplicationTransfers
- ConvertTo-FormattedNumber
- Get-NaHyperV
- Get-NaToolkitVersion
- Get-NaVolCloneSplitEstimate

Issues fixed in Toolkit 1.2
- getattr-ops data missing from NfsTopInfo
- Missing timestamp field in perf InstanceData
- Snapvault max transfer rate incorrect