**Converged Perfstat Release Guide:**

* **Introduction:**

Perfstat Converged is a diagnostics data collection tool for data ONTAP running Cluster-Mode and data ONTAP running 7-mode, with several key properties:

* Captures all needed performance information with one command.
* Captures information from node(s).
* Captures information from local host and remote Host(s).
* Captures information from network switch(s).
* Captures all information simultaneously for cross correlation
* All captured data is returned as zip/tar file.

Converged Perfstat comes in exactly three flavors:

* Linux version (perfstat8)
* Windows version (perfstat8.exe)
* Apple Mac OS X version (perfstat8)
* **Converged Perfstat8 supported platforms.**
* **Supported ONTAP versions**
* ONTAP 8.0.x (C-mode)
* ONTAP 8.1+ (7-mode)
* ONTAP 8.1+ (C-mode)
* **Supported Client OS versions**
* Windows 2003
* Windows 2008
* Windows XP
* Windows Vista
* Windows 7
* Linux
* Apple Mac OS X Snow Leopard [ Intel x86-64]
* Apple Mac OS X Lion [ Intel x86-64]
* **Switch data collection:**

Converged perfstat is capable of collecting data from network switches. The only supported connectivity medium is SSH protocol. The pre-requisite to collect data from the switch is that the switch must support SSH and the same must have been enabled.

The currently supported switch vendors/models are:

* Cisco: cisco nexus 5000 series
* Brocade: brocade 4100 switch series
* Qlogic: SANbox 5602 FC switch
* **Local host data collection:**

Perfstat Converged collects data from local host by default. The following Operating systems are supported for local host.

* Linux
* Windows
* Windows 7
* Windows XP
* Windows Vista
* Windows 2003
* Windows 2008
* Apple Mac OS X
* Mac OS X Snow Leopard [ Intel x86-64 ]
* Mac OS X Lion [ Intel x86-64 ]
* **Remote host data collection:**

Converged Perfstat collects data from the remote host(s) if specified with command line. The prerequisite to collect data from remote host is that it should have SSH enabled. The only exception to this rule is the host running AIX where RSH is mandatorily configured. The following Operating systems are supported for remote host:

* Linux
* Solaris
* HP-UX
* FreeBSD
* ESX server 3.5
* AIX
* **Converged Perfstat for 7-Mode:**

Converged Perfstat in current mode executes by establishing an interactive session with the node. There is a limitation with 7-Mode, that only one SSH/Telnet interactive session could be active at any instance of time. Because of this constraint if any session is opened with the node, Converged Perfstat fails to execute in current mode for nodes running data ONTAP 7-mode. To overcome this limitation in 7-Mode, a non-interactive way of connectivity with nodes, is been added. This newly added capability is a called "Legacy Mode". With this capability, Converged Perfstat 7-mode executes alike perfstat7. In Legacy Mode, as there is no interactive session established with the node so systemshell and SKTrace data will not be collected in Legacy Mode. To achieve this Legacy Mode capability, a new CLI option [ LEGACY\_MODE ] is been added in Converged Perfstat.

To summarize, 7-Mode Converged Perfstat can be executed in:

* Current Mode
* LEGACY MODE.(with --LEGACY\_MODE at command Line)
* Starting with Current Mode and Later dynamically switching to LEGACY MODE due to an opened SSH/Telnet session.

**3.1 Prerequisite to run Converged Perfstat in 7-Mode:**

There are set of rules which need to be followed while using Converged Perfstat to collect data from the controllers running with Data ONTAP 7-Mode :

* Use puttygen.exe generated public/private SSH key pair for passwordless perfstat run on windows client .
* On Windows clients that are used to execute Perfstat in 7-Mode, the file plink.exe must be present in the perfstat run directory to run parallel commands if the SSH option is enabled at command line.
* The controller must have been added to the 'trusted known host list' on the client before running perfstat. For more details, Please see the FAQ section.
* While running Converged Perfstat 7-mode with 'RSH' [ i.e. default protocal ] either in current mode or Legacy Mode, the "/etc/hosts.equiv" file must have been appended already with the "<hostname> <username>". For the Apple Mac OS X hosts, rsh need to be enabled on the host first.For more details, Please see the FAQ section.
* The 'diag' user account must be unlocked on the controller to collect SystemShell data from it.
* While running Converged Perfstat 7-mode in Legacy mode on Windows2008/Windows7 systems, openSSH must have been setup already by following the procedure given in “openssh\_procedure.txt” bundled with the perfstat binary.
* **Converged Perfstat C-mode:**

Converged Perfstat C-mode executes only in current mode by establishing the interactive SSH session with the node. The Legacy mode is not supported in C-mode.

* **Pre-requisite to run Converged Perfstat in C-mode:**

There are set of rules which need to be followed while using Converged Perfstat to collect data from the controllers running with Data ONTAP C-mode.

* The 'diag' user account must be unlocked on the controller to collect SystemShell data.
* On Windows clients that are used to execute Perfstat in cluster-mode, the file plink.exe must be present in the perfstat run directory to download perfstat generated files from the controllers onto the host.
* **Caveats & Known Issues:**
* Perfstat8 may fail or behave unexpectedly if:
* A controller name cannot be resolved from the host running the perfstat utility.
* User does not have write permissions in the directory where perfstat is running.
* A system shell account for the "diag" user is necessary for perfstat to collect the system shell commands.
* If individual commands fail to execute they will be logged in the console log.
* If multiple perfstat instances are run simultaneously from within the same directory.
* Known Burts

|  |  |
| --- | --- |
| BURT | Description |
| [405679](http://burtweb.eng.netapp.com:8080/burt-bin/start?id=405679) | Multiple instances of perfstat running against the same cluster will cause one instance of perfstat to fail. |
| [316710](http://burtweb.eng.netapp.com:8080/burt-bin/start?id=316710) | Requests can time out if a controller is under severe load, resulting in all data not being collected. |
| [391050](http://burtweb.eng.netapp.com:8080/burt-bin/start?391050) | Linux Perfstat can fail on a 24 node cluster. |

* **Output :**

The output of the Converged Perfstat follows the below format:

* node (folder)
* host (folder)
* console.log (file)
* perfstat.preset (file)

if switch data collection is used at the command line, then the output also contains

* switch

Detailed view of files generated:

console.log

Perfstat.preset

node-| node-ip1 -- output.data

output.index

Logs (if rastrace (or) profile (or) sktrace (or) log is used).

node-ip2 --- output.data

output.index

Logs (if rastrace (or) profile (or) sktrace (or) log is used).

…

…

host Local-host -- output.data

output.index

remotehost-ip1 -- output.data

output.index

remotehost-ip2 -- output.data

output.index

…

…

Switch

switch-ip1 -- output.data

output.index

switch-ip2 -- output.data

output.index …

…

…

* **Details about files generated under node folder:**

The node folder contains “n” number of sub-folders, where “n” is the number of nodes specified at the command line (or) number of nodes in the cluster, if cluster IP is provided at the command line. Each sub-folder will be having the node IP as the name.

Each sub-folder of the node contains:

* output.data - contains the output that is collected from the node.
* output.index - contains the information about the commands that are

executed on the node with this perfstat run.

* Logs - if profile (or) rastrace (or) log (or) sktrace data collection is provided at the command line, then this folder contains those files.
* **Details about files generated under host folder:**

By default Converged Perfstat will collect local host data, unless –FILER\_ONLY or –z option is used at the command line. Along with local host data, the remote host data (if asked for) will also be collected in this folder. This folder contains sub-folders with the name of the remote host IPs and the local host name.

Each sub-folder in the host contains:

* output.data - output collected from the host.
* output.index - valid commands that run on the host.
* **console.log**

This is perfstat log file where all the debug messages are logged.

* **perfstat.preset**

This is the preset file containing all the commands which perfstat has executed during the run.

* **Switch**

The switch folder contains “n” number of sub-folders, where “n” is the number of switches specified at the command line. Each sub-folder will be having the switch IP as the name.

Each sub-folder of the switch contains:

* output.data - contains the output that is collected from the switch.
* output.index – valid commands that ran on the switch

**6. Converged Perfstat help manual:**

NAME

perfstat v8.1 (20121020\_1826517) Supported Converged ONTAP

SYNOPSIS

Perfstat is a performance diagnostic tool supplied and supported by NetApp.

Usage: perfstat [OPTIONS] <a node-management IP of a node in the cluster>

OPTIONS:

-h --help To see a complete detailed usage

-v --version Prints perfstat version, and

supported ONTAP versions.

-q --quiet Suppress all non-error messages

--verbose Increase verbosity

(beware: prints all command executions)

--debug Beware verbose output. Both 'verbose'

and 'debug' will add client-side timestamps.

--preset-file Use a different preset file.

--print-preset Prints the internal preset file.

-o --output-file Use a different output file name.

(default: "perfstat\_data\_yyyymmdd\_hhmmss.zip").

--sshprivatekey-file SSH private key filename for admin user.

--sample-interval Specifies statit/sysstat frequency.

--config-all captures config information in every iteration.

(default: CONFIG info is captured only during

the first iteration).

-t --time sample time in minutes, per iteration.

(default: 2 minutes).

-i --iteration=N[,M] repeat N times with M seconds between samples.

(default: N=1, M=0)..

-n --nodes=N1[,N2,N3,..] node names to collect from.

(defaults to all nodes).

--max-run-time Maximum Perfstat runtime in minutes.

--full-stutter-statit Split entire time (-t)

into all stutter statit collections.

--no-stutter-statit Disable 'stuttered' statit commands.

--enable-flag=PATTERN Enables data collection for specified flags.

(default: All flags are disabled.

Command with !<flag-name> flag value will be executed).

--unzip=FILE Unzips the specified file.

--include=PATTERN Includes \*only\* commands matching

PATTERN, and excludes everything else.

--exclude=PATTERN Excludes commands matching PATTERN,

and includes everything else.

-H --hosts=H1:username:passWord[,H2:userName:passWord,H3:passWord,..]Remote host names to collect from.

-m --mode=[7-mode/cluster-mode/cluster/c] Specify the mode for the target filers (default: 7-mode).

-z --FILER\_ONLY Specify the flag to capture information from filer(s) only (default: Host and Filer(s)

-s --SSH How the 7-mode parallel commands should be executed, using SSH or RSH. (default: RSH)

-r --MODULE\_INST =ID1[ID2,ID3..]Capture RAS trace for the instances of pre-configured modules(default: instance-id 1)

-p --PROFILES=[kahuna,flat,storage,(more)]Capture MP Domain profiles from filers

-l --login Filer login user in 7-mode [ default: root ]

-a --app\_name Name of the application for which data to be collected on host side.

-w --app\_args List of parameters for the application data collection on host side.

-S --switch Collect the data from target switches.The format should <SwitchIP>:<SwitchType>:<loginUser>:<loginPassword>

-k --SKTRACE\_POINT Collect the sktrace for the modules. The format should be <Module>:<debug level> (deafult: 'SK:7,WAFL:4'). The debug level should be in old data ONTAP7.x fashion

-L --log Collect back up log information

--LEGACY\_MODE Collects output in Legacy Mode. This option work only for 7-mode. In Legacy mode, Systemshell and SKTrace data will not be collected.

--diag-passwd diag (systemshell) password. Format should be <systemShellPassword> for C-Mode. <IP1>:<systemShellPassword1>,<IP2>:<systemShellPassword2> for 7-Mode

-g Collects data using OPENSSH

OPTION DESCRIPTION

1. --include=PATTERN/--exclude=PATTERN

The include/exclude patterns consist of semicolon separated parameter tags, which match those from the preset file.

The parameter tags themselves have comma separated values. Any definition lines in the preset file which match ANY

parameter tag and value will be included or excluded from execution. include/exclude option may not be used together.

Here are a few examples:

a. --exclude="SHELL=SYSTEMSHELL"

With this option perfstat will not execute any systemshell commands.

b. --include="TYPE=PERF"

With this option perfstat will only execute commands with TYPE as "PERF".

2. --enable-flag=PATTERN

The enable-flag pattern consist of comma separated flag names, which match those from the preset file.

Any definition lines in the preset file which match ANY flag value will be enabled and added for execution.

Here are a few examples:

a. --enable-flag="snapmirror,wafl" With this option perfstat will enable "snapmirror" and "wafl" flags. Command with "snappmirror" and

"wafl" flag will be executed. Command with "!snappmirror" and "!wafl" flag will be excluded.

3. --diag-password

With this option, user doesn't need to input the systemShell password during perfstat run.

As all the nodes in the cluster will be having same systemShell password, provide only the systemShell password of a single node in case of C-Mode

In case of 7-Mode, the node ip and its respective systemShell password should be provided.

Note: Patterns matching will not be case sensitive.

EXAMPLES FOR C-MODE

%perfstat8.exe 10.10.2.10 --mode="cluster-mode"

%perfstat8.exe 10.10.2.10 --mode="cluster"

%perfstat8.exe 10.10.2.10 --mode="c"

This command runs perfstat with 1 iterations. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

%perfstat8.exe -i 5,10 --verbose 10.10.2.10 --mode="cluster-mode"

This command runs perfstat with 5 iterations, waiting 10 seconds between each iteration.

Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

%perfstat8.exe 10.10.2.10 -a "oracle" -w "runas=someDbaUser,sysdba=true" --mode="cluster-mode"

This command runs perfstat with default preset file and also collects the host side data along with the oracle AWR report.

Here 'someDbaUser' is a pre-existing DBA user. The another option for oracle params is '-w "oracle\_login=testUser/testUser123" '

%perfstat8.exe -i 5,10 --verbose 10.10.2.10 --enable-flag=snapmirror --mode="cluster-mode"

This command enables "snapmirror" flag in preset file and runs perfstat with 5 iterations, waiting 10 seconds between each iteration.

Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

Example to provide systemShell password at command Line.

%perfstat8.exe -i 5,10 --verbose 10.10.2.10 --diag-passwd=abcxyz --mode="cluster-mode"

This command runs perfstat with 5 iterations. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

And "abcxyz" is the systemShell password.

%perfstat8.exe --verbose --nodes=10.10.2.10,10.10.2.11,10.10.2.12 --diag-passwd=abcxyz --mode="cluster-mode"

This command runs perfstat with 1 iteration. Here IP address "10.10.2.10", "10.10.2.11", "10.10.2.12" are the address of a nodes in the cluster.

"abcxyz" is the systemShell password.

EXAMPLES FOR 7-MODE

%perfstat8.exe --nodes=10.72.57.222 --mode="7-mode"

%perfstat8.exe --nodes=10.72.57.222

This command runs perfstat with 1 iterations. Here "10.72.57.222" is the management IP for a 7-mode filer.

%perfstat8.exe -i 5,10 --verbose --nodes=10.72.57.222 --mode="7-mode"

This command runs perfstat with 5 iterations, waiting 10 seconds between each iteration.

Here "10.72.57.222" is the management IP for a 7-mode filer.

%perfstat8.exe --verbose --nodes=10.72.57.222 --mode="7-mode" -a "oracle" -w "runas=someDbaUser,sysdba=true"

This command runs perfstat with default preset file and also collects the host side data along with the oracle AWR report.

Here 'someDbaUser' is a pre-existing DBA user. The another option for oracle params is '-w "oracle\_login=testUser/testUser123" '

%perfstat8.exe -i 5,10 --verbose --nodes=10.72.57.222 --mode="7-mode" --enable-flag=snapmirror

This command enables "snapmirror" flag in preset file and runs perfstat with 5 iterations, waiting 10 seconds between each iteration.

Example to provide systemShell password at command Line.

%perfstat8.exe --verbose --nodes=10.10.2.10,10.10.2.11,10.10.2.12 --mode="7-mode" --diag-passwd=10.10.2.10:abcxyz1,10.10.2.11:abcxyz2,10.10.2.12:abcxyz3

This command runs perfstat with 1 iteration. Here IP address "10.10.2.10", "10.10.2.11", "10.10.2.12" are the address of the respective nodes.

"abcxyz1", "abcxyz2", "abcxyz3" are the systemShell passwords of the respective nodes.

Example to show the usage of LEGACY MODE.

%perfstat8.exe --verbose --nodes=10.10.2.10 --mode="7-mode" --LEGACY\_MODE

This command runs perfstat in Legacy Mode with 1 iteration. Here IP address "10.10.2.10", is the address of the node.

Example to show the usage of -g option in LEGACY MODE.

%perfstat8.exe --verbose --nodes=10.10.2.10 --mode="7-mode" -g --sshprivatekey-file=<sshPrivateKeyFile> --LEGACY\_MODE

This command runs perfstat in Legacy Mode with 1 iteration. Here IP address "10.10.2.10", is the address of the node and sshPrivateKeyFile is the name of the sshprivate key.

Note: System shell account for "diag" user needs to be setup before running perfstat with system shell commands.

Note: Converged perfstat does not accept empty password

COPYRIGHT

Copyright 2009, NetApp All Rights Reserved.

* **Examples of Perfstat Commands:**
* This command runs perfstat with 1 iteration. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

Command:

prompt% <perfstat\_binary> <ip-address> (Format) <mode option>

Windows:

prompt% perfstat8.exe 10.10.2.10 --mode="c"

Linux:

prompt% perfstat8 10.10.2.10 --mode="c"

* This command runs perfstat with 1 iteration. Here IP address "10.10.2.10" is a management IP address for a 7-mode filer.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe --nodes=10.10.2.10 --mode="7-mode"

Linux:

prompt% perfstat8 --nodes=10.10.2.10 --mode="7-mode"

* This command enables "snapmirror" flag in preset file and runs perfstat with 5 iterations, waiting 10 seconds between each iteration.

Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

Command:

prompt% <perfstat\_binary> -i 5,10 <ip-address> (Format)

Windows:

prompt% perfstat8.exe -i 5,10 10.10.2.10 --enable-flag=snapmirror --mode="cluster"

Linux:

prompt% perfstat8 -i 5,10 10.10.2.10 --enable-flag=snapmirror --mode="cluster"

* This command collects data based on the preset file provided. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster and "user\_defined\_preset" is the user defined preset file.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe 10.10.2.10 --preset-file user\_defined\_preset --mode="cluster"

Linux:

prompt% perfstat8 10.10.2.10 --preset-file user\_defined\_preset --mode="cluster"

* This command collects oracle AWR report. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster and 'someDbaUser' is a pre-existing DBA user. Another option for oracle params is '-w "oracle\_login=testUser/testUser123"

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe 10.10.2.10 -a "oracle" -w "runas=someDbaUser,sysdba=true" --mode="cluster"

Linux:

prompt% perfstat8 10.10.2.10 -a "oracle" -w "runas=someDbaUser,sysdba=true" --mode="cluster"

* This command runs perfstat with 5 iterations, waiting 10 seconds between each iteration. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe 10.10.2.10 –i 5,10 --mode="cluster"

Linux:

prompt% perfstat8 10.10.2.10 –i 5,10 --mode="cluster"

* This command runs perfstat with 5 iterations, waiting 10 seconds between each iteration and with the system shell password mentioned in the command line.

**For c-mode:**

Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster. 10.10.2.11, 10.10.2.12 are the IP addresses of the nodes in the cluster. “abcxyz” is the systemshell password of the cluster.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe 10.10.2.10 –i 5,10 --diag-passwd=abcxyz --mode="cluster"

prompt% perfstat8.exe –-nodes=10.10.2.11, 10.10.2.12 –i 5,10 --diag-passwd=abcxyz

Linux:

prompt% perfstat8 10.10.2.10 –i 5,10 --diag-passwd=abcxyz --mode="cluster"

prompt% perfstat8 –-nodes=10.10.2.11, 10.10.2.12 –i 5,10 --diag-passwd=abcxyz

**For 7-mode:**

10.10.2.11, 10.10.2.12 are the IP addresses of the nodes in the cluster. “abcxyz1”, “abcxyz2” are the systemshell passwords of the nodes 10.10.2.11 and 10.10.2.12.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe –-nodes=10.10.2.11, 10.10.2.12 –i 5,10

--diag-passwd=10.10.2.11:abcxyz 1, 10.10.2.12:abcxyz2

Linux:

prompt% perfstat8 –-nodes=10.10.2.11, 10.10.2.12 –i 5,10

--diag-passwd=10.10.2.11:abcxyz 1, 10.10.2.12:abcxyz2

* This command runs perfstat with 1 iteration and collects the profile data. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster and “kahuna” is the profile domain.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe 10.10.2.10 –PROFILES=kahuna --mode="cluster"

Linux:

prompt% perfstat8 10.10.2.10 –PROFILES=kahuna --mode="cluster"

* This command runs perfstat with 1 iteration and collects the rastrace data. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster and “1” is the module id.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe 10.10.2.10 –MODULE\_INST=1 --mode="cluster"

Linux:

prompt% perfstat8 10.10.2.10 –MODULE\_INST=1 --mode="cluster"

* This command runs perfstat with 1 iteration and collects the SKTRACE data. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe 10.10.2.10 –SKTRACE\_POINTS=”SK:7,WAFL:4” --mode="cluster"

Linux:

prompt% perfstat8 10.10.2.10 –SKTRACE\_POINTS=”SK:7,WAFL:4” --mode="cluster"

* This command runs perfstat with 1 iteration and collects the log data. Here IP address "10.10.2.10" is a node-management IP address of a node in the cluster.

Command:

prompt% <perfstat\_binary> <ip-address> (Format)

Windows:

prompt% perfstat8.exe 10.10.2.10 –log --mode="cluster"

Linux:

prompt% perfstat8 10.10.2.10 –log --mode="cluster"

Note: System shell account for "diag" user needs to be setup before running perfstat with system shell commands.

Note: Perfstat for Linux Platfrom only supports empty passphrases.

COPYRIGHT

Copyright 2009, NetApp All Rights Reserved.

* **FAQs:**
* ***Where to get plink.exe and puttygen.exe to generate SSH public/private key pair?***

**Ans**: These executables can be downloaded from this URL:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

* ***What exactly is this LEGACY MODE?***

**Ans :** In 7-Mode, under normal case Converged Perfstat executes by establishing an interactive session with the node. The limitation with this interactive session is, when a SSH/Telnet session is already opened with the node, Converged Perfstat could not enter the node shell and hence it will fail to execute. In Legacy Mode, there is no interactive session with the node. So, Converged Perfstat executes even if there is an SSH/Telnet session already opened with the node. In Legacy Mode, Converged Perfstat will not collect Systemshell and SKTrace data.

* ***I have already opened an SSH session with the node and then am running Converged Perfstat in current mode (without --LEGACY\_MODE specified at command line). Will my Perfstat run collect data?***

**Ans :** Converged Perfstat will fail at the first try. Later, it will warn the user about the opened SSH session and suggest the user whether to try running Perfstat in LEGACY MODE. If the user accepts and says “y” or “yes”, Perfstat will run in LEGACY MODE and collects data except the Systemshell and SKTrace data. If user says “n” or “no”, Perfstat will warn the user to close the opened SSH session and retry using the current mode.

***7.4 How to setup the ssh key on the filer?***

**Ans**: Here are the steps to do sshkey generation and exchange for windows.

For Linux hosts, "ssh-keygen" command can be used to generate the public/private key pair.

Note: Ensure that you type a real value for the password and the password field is not left blank.Using SSH instead of RSH ensures security, but requires interactive authentication unless you first perform a host key exchange. Using the free, open source utilities putty, puttygen, and plink, you can control the filer from a Windows host and send non-interactive commands much like you would using RSH, but without the security risks.

Enable SSH Host Key Exchange

This document covers implementing SSH (using plink.exe, putty.exe, pscp.exe) as an alternative to RSH for security conscious environments.

* Set up SecureAdmin on the filer (if you have not already done so).
* Download putty.exe, plink.exe, puttygen.exe, pscp.exe and place in C:\windows or some directory in your system path.
* Run puttygen (Click Start, Run, type puttygen, click OK).
* When puttygen loads, click the Generate button. You will be prompted to move your mouse around the windows to generate randomness. Make sure the Parameters are set as shown below (SSH-2 RSA).
* At this point, we need to save our private key file somewhere on the host by clicking the Save Private Key button. In this scenario, we chose C:\windows\privkey.ppk as the path/filename for my private key. Not the most secure location, but easy to remember.

OPTIONAL - You may want to set an ACL on this file so that it can be accessed only by the users that will connect via plink (or VFM using plink) as shown below:

cacls c:\windows\privkey.ppk /g <YOURUSERNAME>:f <VFMUSERNAME>:f administrators:f In the above example, plink.exe is being used both for the RSH replacement script (srsh.bat) as well as for VFM . So, I want to grant full access to both of those users as well as the 'local administrators' group (and no-one else). Of course, this step is optional, but you really should do something to keep your private key file relatively secure.

* Next, save your public key. Cut and paste the highlighted text into WordPad and save the file as authorized\_keys.

Note: DO NOT click save public key as this creates a format that the filer does not understand. This file needs to be saved to: [\\HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E"<HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E"FILERNAMEHYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E">HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E"\c$\etc\sshd\HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E"<HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E"USERNAMEHYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E">](file://\\%25252525253cfilername%25252525253e\c$\etc\sshd\%2525252525253CUSERNAME%2525252525253E)\.ssh\authorized\_keys. However, the problem is, there is no such folder. If you are using CIFS to copy the file, this can be problematic as CIFS does not let you create folders with a "." at the beginning of their name.

To get around this, create a folder named <USERNAME> in the following location using Windows Explorer: [\\HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/"<HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/"FILERNAMEHYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/">HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/"\c$\etc\sshd\](file://\\%25252525253cfilername%25252525253e\c$\etc\sshd\)

e.g.- [\\filername\c$\etc\sshd\joeadmin](file://\\filername\c$\etc\sshd\joeadmin)

Navigate to the newly created <USERNAME> folder using Windows Explorer and create a subfolder called ssh.

FROM THE FILER CONSOLE, issue the following command:

mv /etc/sshd/<USERNAME>/ssh /etc/sshd/[USERNAME]/.ssh

Now you have the proper directory structure, so save the 'authorized\_keys' file into this folder:

[\\HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E/.ssh/authorized\_keys"<HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E/.ssh/authorized\_keys"FILERNAMEHYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E/.ssh/authorized\_keys">HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E/.ssh/authorized\_keys"\c$\etc\sshd\HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E/.ssh/authorized\_keys"<HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E/.ssh/authorized\_keys"USERNAMEHYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E/.ssh/authorized\_keys">HYPERLINK "file://%25252525253cfilername%25252525253e/c$/etc/sshd/%2525252525253CUSERNAME%2525252525253E/.ssh/authorized\_keys"\.ssh\authorized\_keys](file://\\%25252525253cfilername%25252525253e\c$\etc\sshd\%2525252525253CUSERNAME%2525252525253E\.ssh\authorized_keys)

We want to cache the filer's key in our registry, so issue the following command from the Windows system:

plink -ssh -l root <FILERNAME> -i c:\windows\privkey.ppk

You will be prompted to save the key to the local registry. Answer "yes" (this is required for VFM to work with plink.exe).

SSH access is now enabled and commands can be sent to the filer by simply preceding them with ssh.

* ***Generated a public SSH key with puttygen.exe and given it a passphrase but don’t know what to do with the saved key?***

**Ans**: Once the public/private key pair is generated by puttygen.exe, you should put the public key on the filer. To achieve the same, please create the root directory under “/etc/sshd” directory and further create “.ssh” directory under the newly created “root” directory. Once the “/etc/sshd/root/.ssh” directory is ready, please create the “authorized\_keys” file under this directory path by copying the content of the “public key” you just generated using puttygen.exe. So after it, effectively you would have a file called “/etc/sshd/root/.ssh/authorized\_keys” having the public key content.Now the private key from this public/private key pair should be used with the Perfstat command line.

* ***Copied the plink.exe to the same folder as that of perfstat8, but don’t know what else to do with it?***

**Ans**: Nothing is expected to be done explicitly after putting this exe. Perfstat would use it internally to execute the “parallel” commands, in case of 7-mode.

* ***The filer must have been added to the 'trusted known host list' on the client before running the perfstat. Where exactly do we need to add to this trusted host list?***

**Ans**: The meaning for the phrase “trusted known host list’ is that the server’s host “Keys” need to be cached in the system registry. This requirement is imposed because Perfstat executes plink in batch mode. However, you cannot respond to the request to update Window's registry.

So precisely, to achieve the above, please run plink directly from a command prompt, and answer "y" when plink asks to update the registry. Once the key is successfully cached, perfstat should be able to execute plink in batch mode

For example:

%plink.exe  -ssh –x –a  -pw  <your password for anyUser> -l <anyUser>   <your filer name/IP>  “version”

The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.

The server's rsa2 key fingerprint is:

ssh-rsa 768 67:4b:5a:0b:14:e9:5c:e0:65:b5:92:ef:48:26:92:38

If you trust this host, enter "y" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, enter "n".

If you do not trust this host, press Return to abandon the connection.

Store key in cache? (y/n) y

NetApp Release R8.1RCxN\_111122\_0214 7-Mode: Tue Nov 22 02:32:29 PST 2011 <1>

* ***How can we test whether the public and private key pair transfer has been successful or not?***

**Ans**: Assuming you wanted to run perfstat without “password”, so you need to modify the command line for the above run as follows:

%perfstat8.exe –i 5,60 –nodes=<node-ip> –mode=”7-mode” --sshprivatekey-file=”priv\_key.ppk”

Please note that the above perfstat would run with “root” user. If you want to run with some other user, you need to setup the key for that user and “-l” option need to be used at command line.

* ***I am trying to run perfstat against a 7mode system. When it asks for the nodeshell user, I give the systemshell diaguser name and password since there is no node shell in 7 mode that I know of?***

**Ans**: Nodeshell and Noderun shell are interchangeably used in 7-mode as default shell. Being the systemshell user, ‘diag’ is not exposed to Nodeshell user directly. So please run perfstat with any of the user having administrative privilages ( could be find by the command ‘useradmin user list’ on the filer).

Note: It is assumed that you are trying to run converged perfstat against the filer imaged with OS >= data ONTAP8.1 and you have followed all the prerequisites mentioned in the perfstat8 page on NOW .

* ***What is this --SSH option for?***

**Ans**: In 7-Mode, using the --SSH option makes:

* The parallel commands to run using “ssh”, for current mode.
* All the commands to run using “ssh”, for Legacy Mode.

prerequisite to use --SSH is:

For windows, keep the **plink.exe** in the same location as that of the perfstat binary.

For Linux, --SSH should be used in conjunction with –sshprivatekey-file.

* ***What setting should I do on node, to make Perfstat run in Legacy Mode using rsh?***

**Ans:** The prerequisite to run Perfstat in Legacy Mode using “rsh” is, the local host details should be available at “/etc/hosts.equiv” in the node.

One can do this by using:

***wrfile –a /etc/hosts.equiv “host\_ip login\_name”***

* ***What exactly is this –-time or –t option?***

**Ans**: --time option doesn’t mean the time for which the perfstat will run or the time for which an iteration will run. It specifies the sample time in minutes, per iteration. In other words, the time for which the parallel commands will run.

* ***Is there a way that I can exclude or include specific commands for perfstat run?***

**Ans**: Yes. Using the –exclude or –include options one can exclude or include specific commands for perfstat run.

For further information, please refer to the Converged Perfstat help manual.

* ***Why is this plink.exe required?***

**Ans**: In Current Mode, Converged perfstat tool opens a SSH connection and executes the PRESTATS and POSTSTATS commands over this channel. Due to the limitation of 'only single interactive SSH session allowed' for 7-mode, the parallel commands gets executed using "rsh.exe" on windows and "rsh binary on linux. There are few windows OS versions which does not have native RSH support for example windows2008/windows 7, to make parallel commands work on those windows clients an open source command line tool called "plink.exe" for windows and "ssh" binary for linux is been used.

In Legacy Mode when --SSH option is used, all the commands (PRESTATS, POSTSTATS and PARALLEL command) will be executed using “plink.exe”.

* ***Why is the option “flat” used for profiling?***

***Ans:*** profiling is triggered by means of –PROFILES or –g at command line. These options take the profile domains as the input. “flat” is the mixture of the other domains like kahuna and storage.

* ***How can I collect sktrace data?***

***Ans:***  SKTRACE can be triggered by Converged Perfstat by using –SKTRACE\_POINTS (or) –k. The format should be <Module>:<debug level> (deafult: 'SK:7,WAFL:4'). The debug level should be in old data ONTAP7.x fashion

* ***What and why is this –g option used for?***

**Ans:** There are few windows OS versions which does not have native RSH support for example windows2008/windows 7. On these machines/others, using plink.exe for executing commands using “ssh” hangs. So, “openssh” is used to execute the commands using –g option. This –g option is available only with --LEGACY\_MODE and --sshprivatekey-file.

Please refer to “openssh\_procedure.txt” file to know the steps to follow, so as to make –g option work.