TZWorks® Volume Shadow Snapshot Enumerator (*vssenum*) Users Guide

Abstract

vssenum is a standalone, command-line tool that lists the Volume Shadows that are currently available from a live Windows system. The output is tailored so that it can be easily to be used in scripts. ***vssenum*** only runs on Vista and later Windows operating systems.

www.tzworks.com

Contact Info: info@tzworks.com

Document applies to v0.34 of ***vssenum***

Updated: Aug 24, 2022

Table of Contents

TZWorks® Volume Shadow Enumerator (*vssenum*) Users Guide

Webpage: http://www.tzworks.com/prototype_page.php?proto_id=30
Contact Information: info@tzworks.com

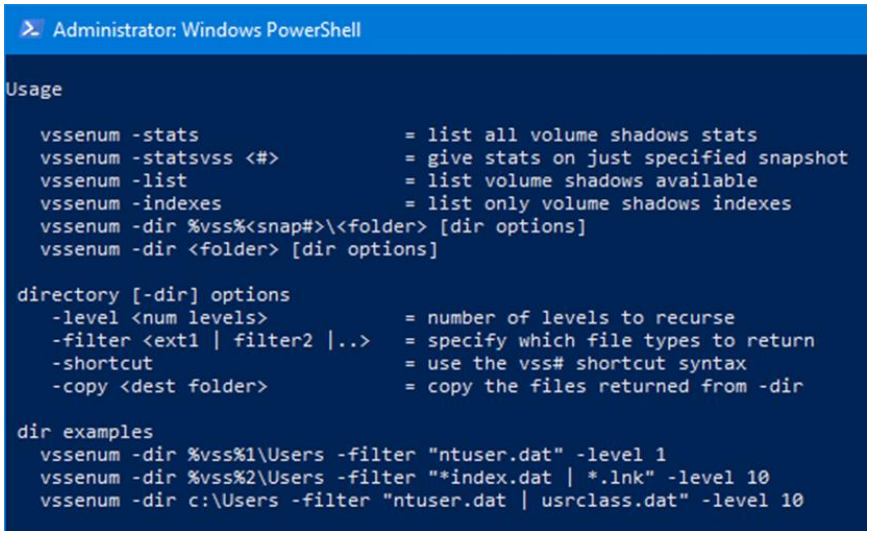
1 Introduction

vssenum is a command line tool that only works on Windows and its purpose is to enumerate the Volume Shadows on the host machine.

The purpose of this tool was not to recreate the built in ***vssadmin*** utility that is part of the Window OS, but to have something that could assist in testing out the other tools that were Volume Shadow aware.

2 How to Use *vssenum*

There are 4 options for Volume Shadow enumeration: (a) display volume shadow statistics, (b) display only volume shadow symbolic links, (c) display volume shadow snapshot indexes, and (d) enumerate a directory given a folder and snapshot index. Below is the command line menu:



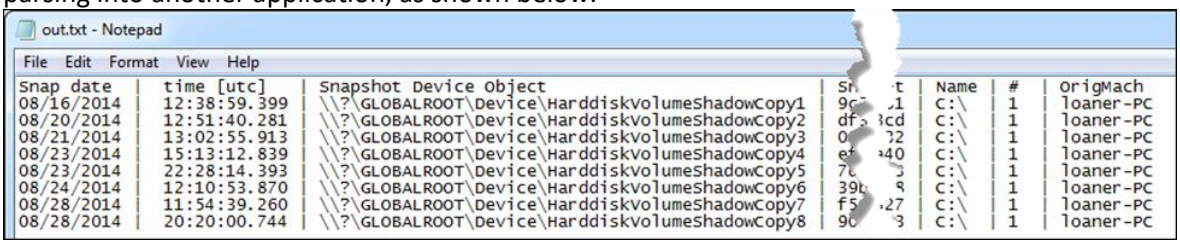
To run ***vssenum*** successfully, two things must happen: (a) the version of ***vssenum*** needs to be the same version of the operating system architecture. Specifically, a 32-bit version of ***vssenum*** only works on a 32-bit version of Window. The same is true for the 64-bit version. The second (b), is that the tool must be run with administrative privileges.

Normally, the architecture constraint is not an issue with the 32-bit versions of other TZWorks tools (eg. One can normally use a 32-bit version of a binary for a 64-bit machine), it is a constraint on ***vssenum*** due to the library dependency limitations that ***vssenum*** uses on some built in Microsoft libraries to allow it to enumerate the volume shadows.

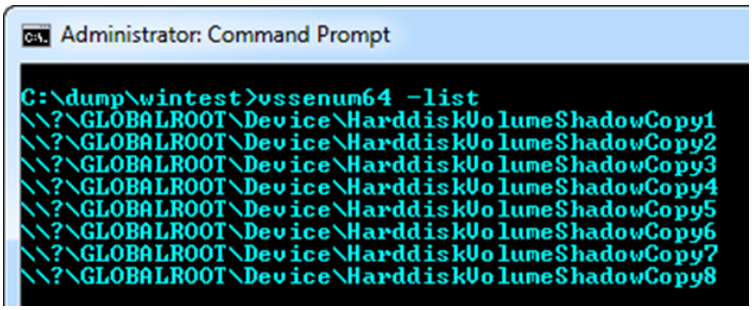
2.1 Understanding the Output

Presently, ***vssenum*** offers 3 types of output.

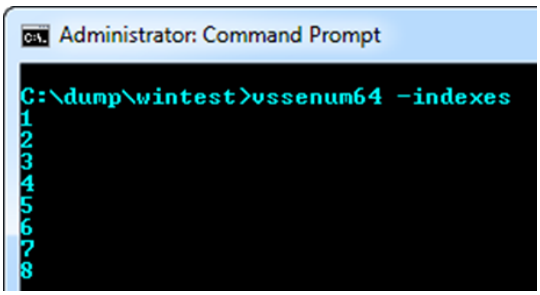
The first uses the ***-stats*** option and separates all the volume shadows statistics of each of the snapshots found on the system into fields. Each field is separated by the pipe character "|" to allow for easy parsing into another application, as shown below.



The second uses the ***-list*** option and just lists the volume shadow device objects for each snapshot as shown below:



The third uses the ***-indexes*** option and it will just list the snapshot number as shown below:



The fourth uses the ***–dir <starting folder>*** option and will list the files in the folder and subfolders in such a way that they can be piped into another tool. There are a number of sub-options, to include: (a) ***-level***

<depth> to specify how many directories to recursively traverse, where the default is 1 folder, (b) ***-filter <filemask>*** to specify a type of file to return, and (c) ***-shortcut*** to specify whether the volume shadow shortcut syntax should be used.

Below are 2 examples using different options to pull the ntuser.dat and usrclass.dat hives from volume shadow snapshot #2. The ***-filter*** option allows one to add multiple filters to the directory enumeration, which in this case, are the user hives. The first example uses the ***-shortcut*** option, while the second does not. If used in conjunction with another tool one can easily script the output of ***vssenum*** to give one the control to process the desired file(s).

```
C:\dump\wintest>vssenum64 -dir %vss%2\Users -filter "ntuser.dat | usrclass.dat" -level 10 -shortcut
%vss%2\users\default\ntuser.dat
%vss%2\users\acct1\appdata\local\microsoft\windows\usrclass.dat
%vss%2\users\acct1\ntuser.dat
%vss%2\users\acct2\appdata\local\microsoft\windows\usrclass.dat
%vss%2\users\acct2\ntuser.dat

C:\dump\wintest>vssenum64 -dir %vss%2\Users -filter "ntuser.dat | usrclass.dat" -level 10
\\?\globalroot\device\harddiskvolumeshadowcopy2\users\default\ntuser.dat
\\?\globalroot\device\harddiskvolumeshadowcopy2\users\acct1\appdata\local\microsoft\windows\usrclass.dat
\\?\globalroot\device\harddiskvolumeshadowcopy2\users\acct1\ntuser.dat
\\?\globalroot\device\harddiskvolumeshadowcopy2\users\acct2\appdata\local\microsoft\windows\usrclass.dat
\\?\globalroot\device\harddiskvolumeshadowcopy2\users\acct2\ntuser.dat
```

The ***-dir*** option is also smart enough to enumerate mounted volumes that are not volume shadows. Just substitute the ***%vss%<snap#>*** with the drive letter.

The latter three options ***–list***, ***-indexes*** and ***-dir*** are useful in automation when applied to making a script to parse certain artifact from all the snapshots on a system.

2.2 Using *vssenum* in a Script

One problem with pulling artifacts from volume shadows is finding which shadow copies are available on the system in question. Once this is known, one can read the desired volume shadow using the device object name of the volume shadow. Encapsulating this enumeration within a script and pulling the requisite data can cause some convoluted scripting. ***vssenum*** makes scripting of the enumeration of shadow copies much easier.

For example, using the ***-indexes*** option, one can take the output of ***vssenum*** and feed it into another tool to parse some artifact. Below is a script that does this and is tailored to work for a number of *TZWorks* tools that are volume shadow aware.

```
rem  ..................................................
rem   vsswrap64.bat for 64 bit Windows
rem   Copyright TZWorks LLC, All Rights Reserved
rem  ..................................................
rem

@echo off
set toolname=%1

if "NULL%toolname%"=="NULL" goto arg_error

for /f "delims=" %%i in ('vssenum64.exe -indexes') do (
  @echo Processing %%i 1>&2
  call %toolname% -vss %%i %2 %3 %4 %5 %6 %7
  @echo   result = %errorlevel% 1>&2
)

goto fin

:arg_error

@echo syntax:  %0 ^<toolname^> ^<arg1^> ^<arg2^> ^<arg3^> ^<arg4^> ^<arg5^> ^<arg6^> 1>&2

:fin
```

For the above example, the script is named vsswrap64.bat. Below are examples of using this script to parse a certain artifact from all the volume shadows on a system.

For ***sbag***, one can do the following:

```
vsswrap64 sbag64 -csv > sbag.results.csv
```

For ***jp***, one can do the following:

```
vsswrap64 jp64 -csv > jp.results.csv
```

For ***lp***, one can do the following:

```
vsswrap64 lp64 -csv > lp.results.csv
```

For ***jmp***, one can do the following:

```
vsswrap64 jmp64 -csv > jmp.results.csv
```

For ***usp***, one can do the following:

```
vsswrap64 usp64 -csv > usp.results.csv
```

2.3 Piping *vssenum*'s Output into another *TZWorks* Tool

Using the *-dir* option allows one to take the output of ***vssenum*** and pipe it into any another *TZWorks* tool that exports the *-pipe* option, such as ***cafae***, ***id***, ***sbag***, etc. Going broader, essentially any tool that can take standard input as the mechanism to identify which file to parse, ***vssenum*** can be used. Furthermore, because of the flexibility of the *-filter* option, one can be very exact on a set of possible conditions to have the file of choice returned. Below are some examples:

For ***id***, one can pipe in all the *index.dat* files from all the user accounts for a specified volume shadow

```
vssenum64 –dir %vss%1 \Users -level 99 -filter  “index.dat” | id64 -pipe > out.csv
```

For ***cafae***, one can pull the system, software and security hives with a one liner:

```
vssenum64 -dir %vss%1\windows\system32\config -filter "system | software | security" | cafae64 -pipe > out.txt
```

Each of the requested hives will be processed, as shown in ***cafae's*** output:

```
analyzing hive: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\windows\system32\config\SECURITY
analyzing hive: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\windows\system32\config\SOFTWARE
analyzing hive: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\windows\system32\config\SYSTEM
```

To pass in any user hive into cafae, one can use the following filter:

```
vssenum64 -dir  %vss%1\Users  -filter “ntuser.dat | usrclass.dat”  -level 10  | cafae64 –pipe > out.txt
```

Any other *TZWorks* tool that has a ***-pipe*** option can be a recipient to the ***vssenum*** tool.

2.4 Using the *vssenum* to do simple copying of files

If one is in a pinch and needs a way to copy files from a Volume Shadow or any mounted volume, one can use the -copy option to copy files from a directory specified by -dir. The -copy option takes one argument to tell vssenum where you want the files to be copied to. For example, if you wanted to copy all the LNK files from a Volume Shadow, one could use the following command:

```
vssenum64  -dir %vss%1\Users  -filter “*.lnk”  -level 9  -copy c:\dump\lnkfiles
```

One needs to keep in mind that copying files this way could cause collisions between files that have the same name in different source directories, since this option only copies the file without preserving the folder structure of the source.

3 Available Options

Option	Description
-stats	Shows the stats about the Volume Shadows
-statsvss	Shows the stat about a specified Volume Shadow snapshot. Syntax is ***-statsvss <index of volume snapshot>***
-list	Shows the symbolic names for the Volume Shadows
-indexes	Shows the indexes of the Volume Shadows
-dir	Enumerate one or more directories, given a starting folder. The available

	sub-options include: (a) ***-snap <index>*** to specify which snapshot to target, (b) ***-level <depth>*** to specify how many directories to recursively traverse, where the default is 1 folder, (c) ***-mask <filemask>*** to specify a type of file to return, (d) ***-shortcut*** to specify whether the volume shadow shortcut syntax should be used, and (e) ***-copy <destination folder>*** to copy files from one location to another. There are 2 required options: ***-dir <starting folder>*** and ***-snap <index>***. The rest are optional.
-no_whitespace	Only available for the ***-stats*** option. Used in conjunction with ***-csv*** option to remove any whitespace between the field value and the CSV separator.
-csv_separator	Only available for the ***-stats*** option. Used to change the CSV separator from the default pipe to something else. Syntax is ***-csv_separator ","*** to change the CSV separator to the comma character.
-dateformat	Output the date using the specified format. Default behavior is *-dateformat "yyyy-mm-dd"*. Using this option allows one to adjust the format to mm/dd/yy, dd/mm/yy, etc. The restriction with this option is the forward slash (/) or dash (-) symbol needs to separate month, day and year and the month is in digit (1-12) form versus abbreviated name form.
-timeformat	Only available for the ***-stats*** option. Output the time using the specified format. Default behavior is ***-timeformat "hh:mm:ss.xxx"*** The restrictions with this option is a the colon (:) symbol needs to separate hours, minutes and seconds, a period (.) symbol needs to separate the seconds and fractional seconds, and the repeating symbol 'x' to represent number of fractional seconds.
-utf8_bom	All output is in Unicode UTF-8 format. If desired, one can prefix an UTF-8 *byte order mark* to the CSV output using this option.

4 Authentication and the License File

This tool has authentication built into the binary. The primary authentication mechanism is the digital X509 code signing certificate embedded into the binary.

The other mechanism is the runtime authentication, which ensures that the tool has a valid license. The license needs to be in the same directory of the tool for it to authenticate. Furthermore, any modification to the license, either to its name or contents, will invalidate the license.

4.1 *Limited* versus *Demo* versus *Full* in the tool's Output Banner

The tools from *TZWorks* will output header information about the tool's version and whether it is running in *limited, demo* or *full* mode. This is directly related to what version of a license the tool authenticates with. The *limited* and *demo* keywords indicates some functionality of the tool is not available, and the *full* keyword indicates all the functionality is available. The lacking functionality in the *limited* or *demo* versions may mean one or all of the following: (a) certain options may not be available, (b) certain data may not be outputted in the parsed results, and (c) the license has a finite lifetime before expiring.

5 References

1. Microsoft's Windows 7 Software Development Kit (SDK)