

TZWorks® NTFS Directory Utility (*ntfsdir*) Users Guide



Abstract

ntfsdir is a standalone, command-line tool that can enumerate any directory on a NTFS volume. ***ntfsdir*** can operate on a live volume, an image of a volume or a VMWare volume. ***ntfsdir*** runs on Windows, Linux and macOS.

Copyright © TZWorks LLC

www.tzworks.com

Contact Info: info@tzworks.com

Document applies to v1.46 of ***ntfsdir***

Updated: Apr 25, 2025

Table of Contents

1	Introduction	2
2	How to Use <i>ntfsdir</i>	3
3	Examples	4
3.1	To view a directory within a live (mounted) partition	4
3.2	To view a directory within an unmounted partition that is in the form of a file that was generated by a 'dd' utility	4
3.3	To dump all the dates associated with files and directories to an XML or CSV file	5
3.4	To dump only selected records, one can use the built-in Windows 'find' command.	5
3.5	To dump directory from a VMWare image	5
3.6	Accessing Volume Shadow Copies	5
4	Known Issues	6
5	Available Options	6
6	Authentication and the License File	8
6.1	<i>Limited</i> versus <i>Demo</i> versus <i>Full</i> in the tool's Output Banner	8
7	References	9

TZWorks® NTFS Directory Enum (*ntfsdir*) Users Guide

Copyright © TZWorks LLC

Webpage: http://www.tzworks.com/prototype_page.php?proto_id=8

Contact Information: info@tzworks.com

1 Introduction

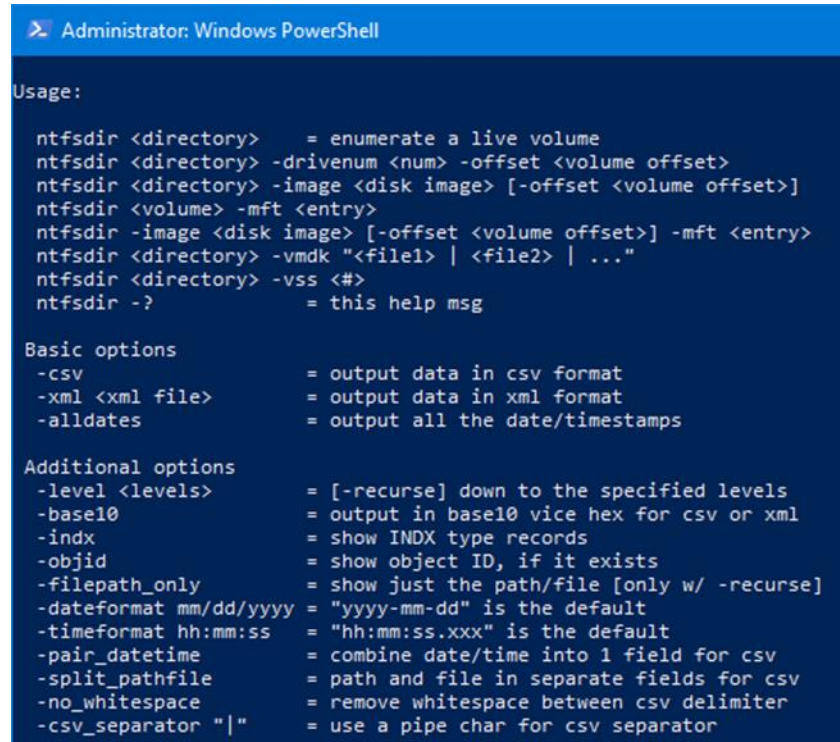
ntfsdir is a command line version of a tool that traverses files and directories of live NTFS partitions. This tool will list additional directory items that the built in windows directory command 'dir' will not.

The objective for this project was to have a tool that could show 'all' the directories and files on a live NTFS system. While there are currently other tools that do this sort of thing for the unmounted NTFS volume, there were not any free tools that could do this on a live system.

The engine of this tool is Windows API agnostic and therefore, was recompiled to run on Windows, Linux and macOS. To do this, the internals of ***ntfsdir*** traverses the volume of the logical drive specified reading the raw sectors to enumerate the desired path. While the drive is only opened in 'read' mode (vice read/write), Windows requires ***ntfsdir*** to run with 'administrator' privileges to examine a live NTFS partition.

2 How to Use *ntfsdir*

The screenshot below is the command line menu for this tool. The options allow for directory traversal for a number of situations: (a) live mounted NTFS volume, (b) *dd* image of a NTFS volume, and (c) VMWare NTFS volume.



```
Administrator: Windows PowerShell

Usage:

ntfsdir <directory>      = enumerate a live volume
ntfsdir <directory> -drivenum <num> -offset <volume offset>
ntfsdir <directory> -image <disk image> [-offset <volume offset>]
ntfsdir <volume> -mft <entry>
ntfsdir -image <disk image> [-offset <volume offset>] -mft <entry>
ntfsdir <directory> -vmdk "<file1> | <file2> | ..."
ntfsdir <directory> -vss <#>
ntfsdir -?               = this help msg

Basic options
-csv                     = output data in csv format
-xml <xml file>          = output data in xml format
-alldates               = output all the date/timestamps

Additional options
-level <levels>         = [-recurse] down to the specified levels
-base10                 = output in base10 vice hex for csv or xml
-indx                  = show INDX type records
-objid                  = show object ID, if it exists
-filepath_only          = show just the path/file [only w/ -recurse]
-dateformat mm/dd/yyyy  = "yyyy-mm-dd" is the default
-timeformat hh:mm:ss    = "hh:mm:ss.xxx" is the default
-pair_datetime          = combine date/time into 1 field for csv
-split_pathfile         = path and file in separate fields for csv
-no_whitespace          = remove whitespace between csv delimiter
-csv_separator "|"      = use a pipe char for csv separator
```

The output is similar to a normal directory traversal with added options to view INDX data, Object ID data, Alternate Data Streams or to display the date/time in another format. For those entries that have symbolic links, the link and target are displayed together.

cmdline: ntfsdir64 c:\ -timeformat hh:mm:ss.xxxxxxxx -dateformat yyyy/mm/dd -indx -objid									
last modified [UTC]		MFT & seq num		size	name				
2013/09/27 15:32:39.641898200		5	5		dir	c:\			
2013/09/27 15:32:39.641898200		5	5	12288	indx	c:\$I30			
2011/01/16 15:52:47.861369300		4	4	4096		c:\$AttrDef			
2011/01/16 15:52:47.861369300		8	8	0		c:\$BadClus			
2011/01/16 15:52:47.861369300		8	8	104857595904	ads	c:\$BadClus:\$Bad			
2011/01/16 15:52:47.861369300		6	6	3203072		c:\$Bitmap			
2011/01/16 15:52:47.861369300		7	7	8192		c:\$Boot			
2011/01/16 15:52:47.861369300		11	11		dir	c:\$Extend\			
2011/01/16 15:52:47.861369300		2	2	67108864		c:\$LogFile			
2011/01/16 15:52:47.861369300		0	1	375914496		c:\$MFT			
2011/01/16 15:52:47.861369300		1	1	4096		c:\$MFTMirr			
2011/01/16 13:35:17.960871100		36087	3		dir	c:\$Recycle.Bin\			
2011/01/16 15:52:47.861369300		9	9	0		c:\$Secure			
2011/01/16 15:52:47.861369300		9	9	327680	indx	c:\$Secure:\$SDH			
2011/01/16 15:52:47.861369300		9	9	393216	indx	c:\$Secure:\$SII			
2011/01/16 15:52:47.861369300		9	9	3055616	ads	c:\$Secure:\$SDS			
2011/01/16 15:52:47.861369300		10	10	131072		c:\$UpCase			
2011/01/16 15:52:47.861369300		3	3	0		c:\$Volume [objid c157352c-7385-4203-a			
2011/10/22 23:13:38.174378800		148638	1	4096		c:.\rnd			
2009/07/14 05:08:56.568037000		93929	1		dir	c:\Documents and Settings\ -> C:\Users			
2011/01/16 16:47:25.753567000		215	3		dir	c:\drvrtmp\			
2011/01/16 16:47:25.753567000		215	3	4096	indx	c:\drvrtmp:\$I30			
2013/09/24 01:59:16.934740800		25036	70		dir	c:\dump\ [objid 45313518-bb85-11e0-b14			
2013/09/24 01:59:16.934740800		25036	70	16384	indx	c:\dump:\$I30			
2013/09/30 23:16:45.445208700		52022	2	6434451456		c:\hiberfil.sys			
2011/01/16 15:57:20.825705100		2029	2		dir	c:\Intel\			
2011/01/16 19:00:05.888997900		17498	5		dir	c:\MSOCache\			
2013/08/10 02:35:37.581634800		79361	24		dir	c:\ntddk\			
2013/09/30 23:16:51.092418600		321105	59	8579272704		c:\pagefile.sys			
2009/07/14 03:20:08.555426400		36088	1		dir	c:\PerfLogs\			
2012/12/10 02:23:08.934349600		36090	1		dir	c:\Program Files\			
2012/12/10 02:23:08.934349600		36090	1	20480	indx	c:\Program Files:\$I30			
2013/03/05 15:58:06.688955800		36277	1		dir	c:\Program Files (x86)\			
2013/03/05 15:58:06.688955800		36277	1	20480	indx	c:\Program Files (x86):\$I30			
2013/05/17 21:19:32.059399600		36393	1		dir	c:\ProgramData\			
2013/05/17 21:19:32.059399600		36393	1	8192	indx	c:\ProgramData:\$I30			
2011/01/16 13:34:57.942009700		94416	1		dir	c:\Recovery\			
2013/09/11 12:40:42.254993600		94102	2		dir	c:\System Volume Information\			
2013/09/11 12:40:42.254993600		94102	2	16384	indx	c:\System Volume Information:\$I30			
2013/08/10 02:36:14.694499900		207486	2		dir	c:\Temp\ [objid 179470ae-ee4e-11e2-b29			
2013/08/10 02:36:14.694499900		207486	2	16384	indx	c:\Temp:\$I30			
2013/09/15 02:04:44.672576500		16483	19		dir	c:\tools\ [objid fa8bf78-21b7-11e0-b9			
2013/09/15 02:04:44.672576500		16483	19	12288	indx	c:\tools:\$I30			
2011/01/16 13:35:07.852053300		36487	1		dir	c:\Users\			
2011/01/16 13:35:07.852053300		36487	1	4096	indx	c:\Users:\$I30			
2013/09/11 13:21:23.752032100		36649	1		dir	c:\windows\			
2013/09/11 13:21:23.752032100		36649	1	16384	indx	c:\windows:\$I30			

3 Examples

3.1 To view a directory within a live (mounted) partition

Open a command prompt via "Run as administrator" to ensure the process has administrative privileges, and then invoke the following command to enumerate the root directory on the 'c' partition.

```
ntfsdir c:\
```

3.2 To view a directory within an unmounted partition that is in the form of a file that was generated by a 'dd' utility

```
ntfsdir \-image "dd_file_of_partition_c"
```

Note: the backward slash '\ ' to denote we want to enumerate the root directory on the file that is a 'dd' image of a NTFS volume.

3.3 To dump all the dates associated with files and directories to an XML or CSV file

```
ntfsdir c:\ -alldates -xml results.xml
```

```
ntfsdir c:\ -alldates -csv > results.csv
```

Note: the use of the [-alldates] switch and the [-xml] and [-csv] switches. The [-alldates] option will output the 4 timestamps for the standard information, 4 timestamps for the filename, and if the file or directory is greater than 8 characters, the default NTFS behavior is to have another filename attribute (one for the DOS short name and one for the long name). Each filename attribute will have another 4 timestamps. Thus, if both the short and long name attribute exist, a total of 12 timestamps will be present.

3.4 To dump only selected records, one can use the built-in Windows 'find' command.

```
ntfsdir c:\ -indx | find "dir" = extract the directories from the root volume
```

```
ntfsdir c:\ -indx | find "indx" = extract the indx type files from the root
```

3.5 To dump directory from a VMWare image

ntfsdir can handle multiple VMDK files to accommodate a snapshot, and its descendants, by separating multiple filenames with a pipe delimiter and enclosing the expression in double quotes. In this case, each filename represents a segment in the inheritance chain of VMDK files (eg. **-vmdk "<VMWare NTFS virtual disk-1> | .. | <VMWare NTFS virtual disk-x>"**). To aid the user in figuring out exactly the chain of descendant images, **ntfsdir** can take any VMDK file (presumably the VMDK of the snapshot one wishes to analyze) and determine what the descendant chain is. Finally, **ntfsdir** will suggest a chain to use.

3.6 Accessing Volume Shadow Copies

To dump directories from within a volume shadow copy, one uses the **-vss <index of Volume Shadow>** option and then specifies the directory to enumerate. For example, if we wanted to look at the Users directory recursively 2 levels deep at Volume Shadow Copy specified by index one, one would use the following:

```
ntfsdir -vss 1 \Users -level 2 > out.txt
```

To determine which indexes are available from the various Volume Shadows, one can use the Windows built-in utility **vssadmin**, as follows:

```
vssadmin list shadows
```


To filter some of the extraneous detail, type

vssadmin list shadows / find /i "volume"

While the amount of data can be voluminous, the keywords one needs to look for are names that look like this:

Shadow Copy Volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1

Shadow Copy Volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2

...

From the above, notice the number after the word *HarddiskVolumeShadowCopy*. It is this number that is passed as an argument to the previous options.

4 Known Issues

Use of backslashes and quotes. When passing in a directory and using quotes around the directory path, ensure you don't use a backslash at the end of the directory, since it will be interpreted as an escape sequence for the quote that follows:... eg.

ntfsdir "c:\windows" <== correct

ntfsdir "c:\windows\" <== incorrect, since the ending \" will be interpreted incorrectly.

5 Available Options

The options labeled as 'Extra' require a separate license for them to be unlocked.

Option	Description
<i>-drivenum</i>	Extract artifacts from a mounted disk specified by a drive number and volume offset. The syntax is <i>-drivenum <#> -offset <volume offset></i> .
<i>-image</i>	Extract artifacts from a volume specified by an image and volume offset. The syntax is <i>-image <filename> -offset <volume offset></i> .
<i>-vmdk</i>	Extract artifacts from a VMWare monolithic NTFS formatted volume. The syntax is <i>-vmdk <disk name></i> . For a collection of VMWare disks that include snapshots, one can use the following syntax: <i>-vmdk "disk1 disk2 ..."</i>
<i>-mft</i>	Process the MFT entry specified. The syntax is <i>-mft <entry></i> .

-vss	Experimental. This option allows one to point to a Volume Shadow copy and process any user hives in the standard users' directories. Syntax is -vss <index of volume shadow copy> . Only applies to Windows Vista, Win7, Win8 and beyond. Does not apply to Windows XP.
-csv	Outputs the data fields delimited by commas. Since filenames can have commas, to ensure the fields are uniquely separated, any commas in the filenames get converted to spaces.
-xml	Output the data in XML format. The syntax is -xml <filename> .
-alldates	Include all the dates/timestamps in the output.
-level	Recurse down to the specified levels. The syntax is either -level <# levels> or -recurse <# levels> .
-base10	Ensure all size/address output is displayed in base-10 format vice hexadecimal format. Default is hexadecimal format.
-indx	Include INDX type records in the output.
-objid	Include the object ID (if it exists) in the output.
-filepath_only	Show just the path/file as the output. This option is only available with the -recurse option.
-no_whitespace	Used in conjunction with -csv option to remove any whitespace between the field value and the CSV separator.
-csv_separator	Used in conjunction with the -csv option to change the CSV separator from the default comma to something else. Syntax is -csv_separator "/" to change the CSV separator to the pipe 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	Output the time using the specified format. Default behavior is -timeformat "hh:mm:ss.xxx" One can adjust the format to microseconds,

	via " hh:mm:ss.xxxxxx " or nanoseconds, via " hh:mm:ss.xxxxxxxxxx ", or no fractional seconds, via " hh:mm:ss ". The restrictions with this option is a 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' is used to represent number of fractional seconds. (Note: the fractional seconds applies only to those time formats that have the appropriate precision available. The Windows internal filetime has, for example, 100 nsec unit precision available. The DOS time format and the UNIX 'time_t' format, however, have no fractional seconds). Some of the times represented by this tool may use a time format without fractional seconds, and therefore, will not show a greater precision beyond seconds when using this option.
-pair_datetime	Output the date/time as 1 field vice 2 for csv option
-split_pathfile	Output path and file into separate csv fields

6 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 (Windows and macOS).

The other mechanism is the runtime authentication, which applies to all the versions of the tools (Windows, Linux and macOS). The runtime authentication 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.

6.1 *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.

7 References

1. <http://en.wikipedia.org/wiki/NTFS> website
1. Brian Carrier's book, File System Forensic Analysis, sections on NTFS
2. Various Microsoft Technet articles