Rdir command design

# General & Options

Objective: A Windows command-line utility in Rust showing detailed information from a file or a folder passed as parameter.

While parameter should accept globbing syntax, autorecurse is off unless there is an option such as -s

Example: if C:\Path is a folder, rdir c:\Path should display folder information, rdir C:\Path\File.ext should display information on this specific file, rdir c:\path\\*.txt should display information on each file

Command line options:  
-s to process subdirectories recursively (not by default)  
-l In the case of links, infos such as size, date, … are coming from target instead of file itself

-v verbose output, show a full-page information for each file. By default, a classic compact tabular presentation.

Command line options should allow to select which information we want to see and level of detail, not specified at this point.

Only include "user level" information about files/directories; low-level internal NTFS attributes are out of scope. Features such as NTFS journaling, file-system transactions, volume shadow copy, quotas… are out of scope.

File content is out of scope and should not be accessed so that stub files are not retrieved by running this command.

For easy maintenance, information gathering should be segregated in separate functions and/or separate source files for easy maintenance, and potential sharing.

ToDo: rdir with an invalid globbing, or simple inexistent path (constant globbing) should show "file not found" or an error message. If the globbing is not constant, but returns no file, the message should be "no match found"

# Information Blocks

## Names

Names: Name and full path. Unicode invisible or special chars should be detailed as {NAME}, to be specified more in detail.  
~~Maybe information on volume and filesystem type.~~

C:\Users\Pierr> rdir "d:MusicOD\Sauve MusicOD vers pihole\_nas-Sonos.bat"

Path: d:MusicOD\Sauve MusicOD vers pihole\_nas-Sonos.bat [File]

File name: Sauve MusicOD vers pihole\_nas-Sonos.bat

Parent: d:MusicOD

Canonical path: D:\Pierre\OneDrive\MusicOD\Sauve MusicOD vers pihole\_nas-Sonos.bat

File type: Windows Batch File

More details of names:

Path is always the unmodified path returned by globbing exploration, can be relative, absolute, local or network… No processing of special characters

File name: Only for files and file links, if the path is more than a file, then show filename part with special characters visible

Parent: If path contains a Parent folder or a parent drive or a network path, then show it with special characters visible.

Canonical path: Full version of path, only shown if it's different than Path

Examples

Path: C:\temp\f1.bat [File]

File name: f1.bat

Parent: C:\Temp

Canonical: (identical to path, not displayed)

Path: f1.bat [File]

File name: (identical to Path, not displayed)

Parent: (empty, not displayed)

Canonical: C:\Pierre\f1.bat

Path: d:MusicOD\Sauve MusicOD vers pihole\_nas-Sonos.bat [File]

File name: Sauve MusicOD vers pihole\_nas-Sonos.bat

Parent: d:MusicOD\

Canonical: D:\Pierre\OneDrive\MusicOD\Sauve MusicOD vers pihole\_nas-Sonos.bat

Path: C:\Temp [Directory]

File name: (not displayed for a directory)

Parent: C:\

Canonical: (identical to path, not displayed)

Path: C:Temp [Directory]

File name: (not displayed for a directory)

Parent: C:

Canonical: C:\Temp

Path: . [Directory]

File name: (not displayed for a directory)

Parent: (empty, not displayed)

Canonical: C:\Develomment\GitHub

Path: .. [Directory]

File name: (not displayed for a directory)

Parent: (empty, not displayed)

Canonical: C:\Develomment

Path: … [Inexistent]

File name: (not displayed for inexistent entry)

Parent: (not displayed for inexistent entry)

Canonical: (not displayed for inexistent entry)

## Type

Similar information that is displayed in Windows File Explorer properties dialog. For .docx files it should mention "Microsoft Word Document (.docx)", Opens with "Word".

## Size

- For files, bytes count and size on disk (actual size allocated based on blocks/clusters), in B or KB, MB, GB with 3 decimal places, using thousands separator for values >=1000  
- For directories, numbers of files and subfolders  
Numbers should be formatted according to current user locale

## Dates

Creation, last modification, last access… Note that last access date should be included only if this command does not update it. Date should be formatted according to current user locale

## Attributes

Standard Windows attributes: Archive, Read only, Hidden, System

Extended NTFS attributes: Stub (FILE\_ATTRIBUTE\_RECALL\_ON\_DATA\_ACCESS), encrypted, sparse file, … and possibly extra information for some attributes (for instance, for compressed files, the uncompressed file size and the actual file size).

## Reparse points

- Type of reparse point  
- For links: symlinks (file and directory), junctions, volume mount points: show target and its type (dir, file, link, inexistent). All other info such as attributes or dates should be of the link itself, not the target, unless explicitly required

## Hard links

Show info when hard link count>1

## Alternate Data Streams

list, size. ~~For standard ADS such as Zone.Identifier, maybe show content~~

## File owner

SYSTEM\name or UUID if it can't be resolved to a valid name

## ACL

(File security attributes (ACL): similar information shown by ICACLS command)

## Misc

Previous versions from File History or restore points, similar to what is shown by Windows File Explorer properties

Offline status and other interesting properties of a file