Space*Sniffer*

"Find lost space on your disks the easy way."
Copyright © 2007-2025 Uderzo Umberto – www.uderzo.it

1. Welcome

1.1 - What's that?

It may happen that a day you discover your compter disk is nearly full. *And you have no clue "why"*. The main purpose of Space*Sniffer* is to help you to dig into the content of your computer storage media, *the easy way*. The approach is to hilight big files. This is done by drawing each file and folder with a shape sized proportionally to its occupation.

The bigger on screen, the bigger on disk. This helps in finding what's eating your storage.

To better help in your quest, a versatile filtering system is available. You can search by file and folder name, extension, size, dates, attributes and even more.

In few words:

- Easy find where big files reside on your storage, even network paths.
- Fast and simple to use, simple interface easy to understand.
- Lets you easily search with file masks (`*.jpg,*.txt`...), by file size (>1mb...), file age (<3months...), attributes (`archive, hidden`...), folder names (`\temp, \ *internet*`), combinations, negations and more.
- File and folder tagging, lets you bookmark elements for later processing.
- Direct access to the Windows Explorer files/folders popup menu with mouse right click.
- Reacts to external storage modifications, keeps always in sync, warns you about external modifications by elements blinking. Discover created/modified files in real time!
- Intuitive navigation with animated zooming effects, even during the scan process.
- Multithreaded scanning engine with smart caching system to minimize disk access.
- Lets you focus and complete the scan on a zoomed portion of your disk even if the master scan is in progress (*if you are in hurry*).
- Scans NTFS Alternate Data Streams if needed.

- Customizable interface: geometry, colors, behaviors, effects, look, styles.

- It doesn't clutter your registry, only a plain XML configuration file.

- It's portable, no installation required, just put the executable somewhere and let's go. You can keep it in your flash key ready to use.

- It's FreeWare. Better, it's DonationWare.

By using Space*Sniffer*, you may say:

"Hei! Look there! A lot of old 10+Mb JPEGs!"

"Oh! That old 2Gb database backup! Better moving it out of the way!"

"Look here! The application XYZ modifies all those files!"

"What is this biiiig folder? Oh, that's the O.S. folder. Better leaving it in place, eheh"

2 .   Q u i c k   S t a r t   P a g e

Please have a look at the *Quick Start page*. You will find information about the tool bar that is not covered here. You should find the *Quick Start page* into the same package where you found this document.

3. Choose your space

The main window of the application appears and a smaller dialog (the start dialog) will ask what you want to do (Choose your space).

Two tabs are available: "`Drives or Paths`" and "`Snapshots`".

3.1 - Drives or Paths

Here you can start a new scan on local or network storage.

You can choose one of your disks or type one or more paths (separated by "`;`" char). Paths can be local or network paths. It works well also with Samba shares. You can navigate your media structure by pressing the `Path` button, or drag'n drop one or more folders on the start dialog to avoid typing. Then, press the `Enter` key or click the `Start` button.

If you typed a path and that path does not exist, an error message will appear. If everything is ok, the scan process begins, the start dialog will close (you can reopen it later) and one or more scan views will appear. You can also drag one or more folders directly on the main window, avoiding the start dialog.

For your convenience, the start dialog is easily usable also with keyboard. If you want the dialog to close, then press `ESC` key. When you are in the main window, the `CTRL+N` shortcut will reopen it.

3.2 - Snapshots

Here you can load a snapshot from file. From version 1.2.0.0 Space*Sniffer* features an internal storage file format that can be used to store a scan then reload later for further review. This may come handy to send scans to System Administrators for assistance and so on.

Other input file formats (du) are already supported and newer may be added in future releases.

Like the previous tab, you can navigate through the `File` button or drag&drop a file in the drop area, then press the `Start` button to start the load procedure. You can even drag&drop a snapshot file on to the main form and start the loading process immediately.

If Space*Sniffer* doesn't recognize the file format/extension it will do nothing.

Once a snapshot has been loaded you can examine and navigate it nearly like it was

freshly scanned. The differences between a fresh scan and a snapshot load are:

- **There is no more live connection to the underlying media:** If you load another machine scan this is quite obvious, but you lose live connection (live updates) even if you load a snapshot produced by the same machine. This is because when you persist the scan on file then reload it later, Space*Sniffer* doesn't know what happened in between so, to ensure a correct live link it would require a new scan of the media but better start from a fresh one then.

- **The snapshot is "framed":** You may notice that the loaded snapshot is framed by an element stating the snapshot file name. This way it's easy for you to distinguish from a loaded snapshot and a live scan.

4 . T h e   S c a n   P r o c e s s

The scan progress will be displayed on the main window. This is a real time display of the scan operation. Lots of rectangles appear on the screen popping out as long as the scan process proceeds. You can immediately interact with them even if the scan operation is going on.

Each rectangle represents a folder or a file (*generically an element*). Folder elements show other elements into them, recursively. A limit has been set to the display recursion, to avoid graphic cluttering. You can change this limit as you wish with `CTRL +` and `CTRL -` shortcuts or with dedicated toolbar buttons.

Elements change in size proportionally to the real size of the file or folder.

📓*Note: This type of graphical representation is called Treemap and was invented by **Prof. Ben Shneiderman**, Professor in the University of Maryland. The bigger the element on screen, the bigger the folder or file on disk. Simple.*

4.1 - Seeing stuff in detail

If you want to examine an element in deep, just *left mouse click **once*** on it, and you will start digging. You can dig into folders until you reach a file element (displayed in a different color). When you click on a folder or file, it will be hilighted and it will drop a shadow to make it easy to track during a scan operation (because Space*Sniffer* is continuosly rearraging and moving all elements to best fit them in your display).

If you wish, you can zoom into a folder element by ***double** clicking on it*. This will expand the folder to the entire view, showing more smaller elements previously hidden because of their small size. When you start zooming in and out you will notice that the navigation keys (`back` and `forward`, like a web browser) will activate. You can go back and forth by pressing them or by `BACKSPACE` and `SHIFT+BACKSPACE` shortcuts.

📓*Note: The navigation keyboard shortcuts are active only when the filter field is not focused*

4.2 - What does Space*Sniffer* display?

Due to the intrinsic nature of the Treemap concept, Space*Sniffer* will not show files with zero byte size. Because the display area is proportional to the file size, a zero size file would occupy zero pixels, hence they are not displayed.

Note: There are particular cases when a zero byte length file is displayed. This happens only when you have zoomed into a folder that contains a zero length file. But also this condition is transient. Anyway, the mission of Space Sniffer is to find big guys, so this should not be a real issue.

Note: Due to the limited available display area, also smaller elements are not shown. You can configure the minimum size of an element to be shown into the view, anyway there are cases with lots of elements that cannot be fully displayed. This doesn't mean that Space Sniffer ignores such elements. It always takes them into consideration but it just doesn't display them. If you cannot discover them by zooming into, you can find them by the export function (see next chapters). Once again, Space Sniffer is to find big guys so, you are warned...

5 .   N a v i g a t i o n

You can navigate the disk structure by the tool bar buttons. Please refer to the *Quick Start* for a visual representation of the user interface. The following commands are available, in order:

5.1 - New view (`CTRL+N`)

Opens the Start Dialog, letting you start with a new scan/snapshot view. Space*Sniffer* lets you open more windows and watch different parts of your disks. If you open more than once the same path (or part of it), the disk will be scanned only once. Space*Sniffer* features a smart caching system that links also to the *disk event system* of the O.S. So if something changes outside the application, Space*Sniffer* will be aware of it and will reflect the change into the view.

💡*Hint: You can open twice the same path on two different views and examine it with different filtering criteria.*

5.2 - Go back (`BACKSPACE`) + Go forward (`SHIFT+BACKSPACE`)

When you navigate the disk structure by zooming in and out, all locations you traverse are kept in memory (like an internet browser). So you can go back and forth as you wish.

5.3 - Go upper level (`CTRL+UP`)

Will zoom out by one folder level, until you reach the view root. The view root is the starting point of the scan operation (the disk drive or the specified initial path).

5.4 - Go to home (`CTRL+HOME`)

Will zoom out at the root point of the view (the disk drive or the specified initial path).

5.5 - Perform a new master scan

While scanning, the button lets you to stop the process. While not scanning, the button lets you start another scan process.

💡*Hint: If you need to track only file change operations and you are not interested in the full media content, a full media scan is not needed. Just start a new scan then stop it with this button. Even if not scanning, Space*Sniffer* will keep track of created/deleted/modified files and you will see them popping into your view. This feature is available only where the O.S. disk event system can detect file changes, so it likely won't work on network drives, for example.*

5.6 - Performs a new scan of the zoomed view (sub scan)

When you are on the view root the button is disabled. When you are in a zoomed folder, you can start a secondary scan process to force the scanning of the selected folder. The purpose is to avoid waiting for the termination of the master scan to be sure the zoomed view is complete. Only one active zoomed scan process is permitted for each view. If you want to focus on another part of the structure and the secondary scan is in progress, you must stop and restart it.

💡*Hint: Both master scan and sub scan are not really needed to keep the view in sync with media content if the O.S. gives modification feedbacks. This usually happens with local medie storage, as said previously. If, on the other hand, you are dealing with a network path for example, you should rescan totally or partially to keep in sync.*

5.7 - Less detail (`CTRL -`) + More detail (`CTRL +`)

Will dig less/more into the display structure. Note that the currently selected folder/file element will always be shown, to avoid you a lot of annoying zoom in/out operations.

5.8 - Use File Classes style (`CTRL+T`)

Will switch between `Flat Colors` style and `File Classes` style. When you feel overhelmed by too much color information you can "*go back flat*" and take a breath.

5.9 - Show free space (`CTRL+F`)

Will show an element that represents the free space of the selected drive. This option will work only if you select a drive (or type a drive path) and not with typed paths that are not drives.

📓*Note: The free space element is visible only in the root element, if you zoom into a folder the free space goes out of sight. It's not possible to keep in sight the free space element also in zoomed views because of the nature of the Treemap. A free space element into a zoomed folder would show a distorted information because of shape proportions.*

💡*Hint: You can keep two views of the same media and use one to dig into and one to keep visible the root with free space element active.*

5.10 - Show Not yet scanned (or Unaccessible) space (`CTRL+U`)

Similarly to the preceding option, but this will show `Not yet scanned space`. This space is space that Space*Sniffer* is aware of, but it has not yet examined. While the scanning process goes on, the `Not yet scanned space` will become smaller and smaller giving space to examined elements. Similar to free space, `Not yet scanned space` will not be displayed if the root path is not a drive, and it will be displayed ony in the root element.

💡*Hint: Sometimes after the scan completes, some "`Unaccessible space`" is visible. This is due to the inability to scan some protected folders. Try running Space**Sniffer** with Administrator privileges if you want to completely examine the media.*

5.11 - Donate

Opens a web connection to the donation page. Please support this software, let me know that you like it by donating something. Thank you in advance.

5.12 - The viewable percent bar

When zooming in and out you may notice a small vertical bar on the left side of the view. This small bar will show you *how much of the entire media* is currently showing. If you go to the view root, the bar will fill the vertical space entirely. If you zoom in, the bar will lower its height, because you are displaying less and less disk space as you keep zooming in. This is useful to understand, for example, that all the files displayed in the current zoomed view use a really small portion of the disk, and you should focus your attention elsewhere to recover big space.

5.13 - The progress bar

Another item you can notice is the progress bar in the upper right side of the view. This will show you the scanning or the filtering progress. Since the total size to be scanned is known only if you select a drive, the progress bar is shown only if you select a drive path. In all other cases, a simple message will be displayed. On the other hand, if a filtering operation is in progress, it will take precedence over the scan percentage progress. At the end of the filtering operation, the scan progress will become visible again.

💡*Hint: The scan process and the filtering process are multi threaded, so you can keep changing filters to search what you are looking for and don't mind the scan process, that will go on in the background. Simply explore your media, you don't need to wait for Space**Sniffer** to terminate the scan!*

6. The Filtering System

The filtering system is the most powerful tool that Space*Sniffer* offers to find what you are looking for. By filtering you can virtually "*slice & dice*" your media, molding the view content to suit your search needs. Just type the filter in the filter entry field, then press the `Enter` key or click the `Filter` button. This works both for fresh scans and loaded snapshots.

6.1 - Filtering by file name mask

It is possible to filter the view by file mask. Masks are defined with special characters "`?`" and "`*`". Just like the O.S. command line. The question mark character means "*a single character*", while the asterisk character means "*zero or more characters*". It's possible to type also a complete file name if you know what are you looking for. By placing a pipe character "`|`" in front of file mask you will negate it and this becomes an exclusion filter.

Example: *by typing* `*.jpg` *and pressing the* `Enter` *key, only JPEGs files will be displayed.*

Example: *by typing* `|*.jpg` *and pressing* `Enter` *key, everything but JPEGs will be displayed.*

6.2 - Filtering by folder name mask

The filter by file name mask applies only to files. Sometimes you may want to search only under certain folders, say, `temp` folders. You don't know where they are and at which nesting level on your media device, but *you know they are there*, and possibly more than one.

In this case you can use a folder name mask filter. Folder name mask filters are the same as file name mask filters, except they apply to folders (*ok, it's obvious*). To identify a pattern as a folder mask pattern, simply prepend it by the '`\`' (*backslash*) char. The backslash char doesn't have the meaning of "*root folder*", it is simply a switch. It has been chosen as "switch" char because it is forbidden in file and folder names (it is a path separator), hence it cannot exist in an element name (with exception if you start sniffing from a folder instead of a drive, but this affects only the root element).

When you apply a folder name mask filter, only files whose path contains a matching folder are included. Moreover, folders whose names match the mask are hilighted with the **BOLD** style in their name. This makes easy to understand why stuff has been included by the filter. If, on the other hand, you are using an exclusion filter (by prepending the whole pattern string by the negation pipe "`|`" char), excluded folders are not hilighted, just because they are not displayed.

Example: *by typing* `\temp` *every file under a* `temp` *folder (at any nesting level) is included. For example,* `C:\Windows\`***temp***`\data\abc.txt` *would be included.*

Example: *on the contrary, by typing* `|\temp` *every file under a* `temp` *folder (at any nesting level) is excluded.*

💡***Hint:*** *You can consider folder and file masks as a two dimensional filter. The first axis is the folder mask, the second axis is the file mask. Both axis work independently. When you apply a folder+file mask you are defining some kind of "coordinate" in a 2D space. You first "cut" the directory tree structure and isolate the searched folders, then examine the wanted file names by the other mask.*

6.3 - Filtering by file size

You can look for files bigger/smaller than a given file size. The syntax of the search string is ">" (or "<") character, followed by the size and the measure unit.

(<u>disksize</u>|clustersize|filesize|logicalsize|size)[<|>][**size**][b|kb|mb|gb|tb]

Available measure units are `b` (byte), `kb` (kilo byte: 1024 b), `mb` (mega byte: 1024 kb), `gb` (giga byte: 1024 mb), `tb` (tera byte: 1024 gb)

Example: *by typing* `>100kb` *Space****Sniffer**** will show only files bigger than 100kb.*

You can prepend the size condition by the keywords `disksize`, `clustersize` (synonims for disk space occupation) or `filesize`, `logicalsize`, `size` (synonims for logical file size). The default is <u>`disksize`</u>.

💡***Hint:*** *Space****Sniffer**** uses a fuzzy match algorithm to understand what you are filtering for. So, you can type* `disksize, disk, dsk, dsksz` *and so on.*

A quick word about *disk occupation* and *logical occupation*. Currently, storage devices that you can normally find attached to your computer organize the space into *clusters*. A *cluster* is the smallest storage area you will affect by writing something on the device. Say, for example, that your disk has a 4Kb *cluster* size. This means that any file you will write on that disk will be sliced into 4Kb "pieces".

Example: *A 20 bytes file (logical size) will occupy (partially) one cluster, in this case 4Kb (disk size). A 5Kb file will occupy two clusters, the first will be filled entirely, the second partially (the 1Kb left by the first cluster). And so on. This means that the disk space occupation is normally greater than the logical file size. Space****Sniffer**** defaults to disk size because you are looking to recover space on disk, but if you wish you can switch to logical*

size with the appropriate filter switch.

6.4 - Filtering by file age

It is possible to search files by their age (normally modification date). The syntax of the search string is similar to the previous one, but the measure unit is different.

(creation|<u>modify</u>|access)[<|>][**age**][seconds|minutes|mours|days|weeks|months|<u>years</u>]

Available measure units are seconds/secs/sec/s, minutes/mins/min/m, hours/hour/h, days/day/d, weeks/week/w, months/month, years/year/y.

Example: *by typing* >1year *Space**Sniffer** will show files not modified in the last year.*

By default the filter applies to the <u>modify</u> date, but you can prepend creation, modification, access (and variations because of the fuzzy match algorithm) to specify which file date you want to filter for.

Example: *by typing* a>1year *Space**Sniffer** will show files never accessed in last year.*

6.5 - Filtering by tag

If you want to find tagged files (*see next chapter about tags*) you can use the :1..4 syntax (the colon is part of the syntax). Instead of 1..4, for your convenience you can use :red, :yellow, :green, :blue, :all or :r, :y, :g, :b, :a. By placing a pipe character "|" in front of tag filter you will negate it and this becomes an exclusion filter.

(|):([red|<u>yellow</u>|green|blue](+|−))**+**

Example: *by typing* :1 *or* :red *you filter red tagged files.*

Example: *by typing* :all *or* :a *you filter all tagged files.*

Example: *by typing* |:3 *or* |:green *you exclude green tagged files.*

Example: *by typing* |:all *or* |:a *filter all but tagged files.*

📝***Note:*** *Since release 1.1.2.0, to accomplish the need of many additional filters, a new syntax has been added for the tag filter. The new syntax needs that the filter starts with the* :tag: *or* :tags: *filter specifier, then you can specify one or more colors for filtering. For tag filtering, the old syntax is presented for backward compatibility.*

Example: `:tag:1` *or* `:tag:r` *or* `:tag:red` *filters red tagged files.*

Example: `:tag:2` *or* `:tag:y` *or* `:tag:yellow` *filters yellow tagged files.*

Example: `:tag:3` *or* `:tag:b` *or* `:tag:blu` *or* `:tag:blue` *filters blue tagged files.*

Example: `:tag:4` *or* `:tag:g` *or* `:tag:green` *filters green tagged files.*

Example: `:tag:a` *or* `:tag:all` *filters all tagged files.*

Example: `:tag:red+green-b` *filters red, yellow but not blue tagged files.*

Example: `|:tag:1,3,-red` *starting pipe always negate the following filter.*

6.6 - Filtering by file attributes

The file attributes filter specifier is `:attr:` or `:attrs:` then you can type your attribute filter. The following attributes are available for filtering:

- `archive`/`archi`/`arch`/`arc`/`ar`/`a`: archive files

- `system`/`sys`/`s`: system files

- `readonly`/`rdonly`/`ronly`/`rdo`/`ro`/`r`: readonly files

- `hidden`/`hidn`/`hid`/`h`: hidden files

- `compressed`/`comprsd`/`compr`/`cpr`/`c`: compressed (by os) files

- `encrypted`/`encrptd`/`encptd`/`enc`/`e`/`crypted`/`crypt`/`cptd`: encrypted (by os) files

- `offline`/`off`/`ol`/`o`: offline files

- `temporary`/`temp`/`tmp`/`t`: temporary (os) files

- `notindexed`/`notindx`/`notidx`/`noidx`/`nidx`/`ni`: not indexed (by os) files

- `sparse`/`sp`: sparse files

- `alternatedatastream`/`alternate`/`altern`/`alt`/`ads`: alternate data streams, this filters works only if the "scan alternate data stream" option is active in configuration.

Note: Possible combinations are similar to the `:tag:` *filter.*

Example: `:attr:archive+hidden`

Example: `:attr:+a-ro,h`

Example: `|:attr:temp`

6.7 - Filtering by file class

You can filter by file class, as defined in the *Configuration Dialog* (see later). This will let you type a shorter filter string instead of write all file extensions defined by the file class.

Example: `:class:Audio/Music` *is equivalent to filter for file extensions* `*.mp2;*.mp3;*.wav;*.ogg;*.aif;*.alp;*.au;*.3gp;*.ac3;*.amr;*.aac;*.m4a;*.wma;*.aiff;*.flac;*.m4a;*.voc` *(providing that you have not altered the original configuration, otherways behavior will change).*

In the above example, `Audio/Music` is the name of the file class as defined in the *Configuration Dialog* under *Styles/File Classes* tab. The configuration name is case insensitive. If the configuration name contains spaces, you can type them aswell because the parsing engine will split commands on "`;`" character (see Filter combinations). If you type a non existent file class name, an error message will occur.

Example: `|:class:Audio/Music` *will negate the result of the previous example.*

6.8 - Filter combinations

If you need to specify more than one condition it is possible by separating each condition by the "`;`" character. Rules are:

- File mask and tag conditions (including class file conditions) are **or**-ed together, excluding masks are and-ed together.

- All other conditions are **and**-ed together

Example: `*.jpg;*.gif;>100kb;<6months` *will show all JPEGs and GIFs bigger than 100kb and younger than 6 months.*

💡***Hint:*** *You can change the filter string also during the scan process. The display will react to the new filter, but the process will always scan all elements. This is because you can change your mind and alter the filter whenever you want. If you do that, a new scan is not required because the filtering is applied to the view and not to the smart cached data.*

⚠️***WARNING:*** *Note that if you delete a folder (by Windows Explorer popup menu) that is currently showing part of all the container files/folders due to a filter rule, you are deleting ALL THE CONTENT, not only the visible content! This is because you are deleting the folder, and Windows Explorer knows nothing about the filter applied into Space**Sniffer**.*

7 . T h e   T a g g i n g   S y s t e m

If you want to track files or folders for better handling you can tag them. There are four different tags: `red`, `yellow`, `green`, `blue`. Simply hover the mouse on a file and press `CTRL+1` for `red` tag, `CTRL+2` for `yellow`, `CTRL+3` for `green` and `CTRL+4` for `blue`. By pressing the same key again you clear the tag (it's an on/off switch). The `CTRL+digit` combinations works always, also if the filter box has the focus. If you are sure that the filter box is not focused, you can also use the simpler `1`, `2`, `3`, `4`, `0` key.

With `CTRL+0` you clear the tag on all elements in the zoomed view (even elements hidden by a filter rule) . This means that the reset is applied on elements starting from the zoomed area (files and folders outside the zoomed area are left untouched).

By tagging an element you put a temporary "bookmark" on it. This lets you continue navigating the disk structure and be sure to find each tagged file by filter, for example. The tag is temporary: This means that when you close all the views containing them, they are lost. Nothing is stored on disk, everything is kept in memory. So, just play with tags!

💡*Hint: You can tag each element only once, but elements will inherit the tag also from parents. This gives lots of filtering flexibility. But, to avoid view cluttering, only the element's own tag is shown.*

Example: You want to exclude some big folders from your view. Define, say, a tag exclusion filter, as explained in the filter chapter. Suppose you want to exclude red tagged elements. Type "|:r" and press ENTER key. Then select the annoying file/element and tag it red. It will disappear from the view (it has not been deleted, only hidden). To restore the element into the view, clear the filter or clear the tag with CTRL+0.

Example: You want a "shopping cart". Open two views of the same media device. On the first view set an exclusion filter "|:g". On the second view set an inclusion filter ":g". Now, if you tag elements to green in the first view, they will disappear from the first view and appear into the second view. When you finished "shopping", you can export your cart with the export function (see later).

8. Configuration

Space*Sniffer* can be configured in behavior and aspect. The configuration is stored in a XML file stored in the same folder of the application executable. No registry messing. Just one single XML file.

8.1 - Geometry tab

Element font size

Lets you to set the graphical panel font size, to accommodate your monitor resolution.

Minimum element size (pixel)

The minimum size in pixel an element must have to appear into view (notice that only relevant items appear into view, not everything. This avoids display cluttering).

Element proportions

If you like a more horizontal or vertical layout for elements, play with this setting. Space*Sniffer* features a layout algorithm that tries to keep elements to the specified proportion. Anyway, it's not always possible due to view space constraints.

Initial detail level

This is the initial detail level used by new views when they're created. After creation, each view can be configured independently with its own detail level as preferred.

Show free space

If set, auto activates the "`Show free space`" option on new views.

Show unknown space

If set, auto activates "`Show unknown space`" option on new views.

Sort items

If set, items are graphically sorted by size, so you can easy see what's big. Bigger element will be placed near the top left edge, letting smaller elements to the bottom right edge.

8.2 - Animation tab

Zoom animation duration

Controls how long the zoom effect takes. The zoom effect is useful to make it clear where you are heading when digging into folders.

DynaZoom

Dynamic zoom animates elements layout as long as the zoom animation goes on. It needs more CPU power but looks better (IMHO). Standard zoom effect simply works with a precomputed screen shot that may also look nice and needs less CPU. It's a matter of personal taste.

Animation smoothness

Controls the frame rate of the scan animation. Higher the frequency the better and smoother the look, but slower the scan. Default is mid-way. Anyway, consider that no screen update is done if no new information is gathered.

DynaSmooth

Dynamic smooth changes the frame rate accordingly to user interaction. If the user works with the mouse over the view, the animation is kept at high frame rates (the value is set by the Animation smoothness parameter). When the user stops working on the view, the frame rate decreases smoothly to the lowest frame rate, letting the scan progress squeeze more CPU power. DynaSmooth lets you set the Animation smoothness to smoother levels while keeping really good scan performances. *Keep it active to get maximum performance*.

8.3 - Behavior tab

Scan Alternate Data Streams

Enable NTFS ADS scanning. ADS are "secondary hidden files" attached to normal files. This is a NTFS feature that can be used by some applications to store auxiliary data related to the main file. However, ADS can be used also by viruses because not all virus scanner are able to examine them. When you turn on this option the scan goes slower, so use it only if needed.

File system changes detection

If enabled (the default), Space*Sniffer* will monitor for file system events and react the TreeMap accordingly. If you feel this is annoying, you can deactivate the detector. By altering this checkbox you are required to restart Space*Sniffer*. When the detector is offline, changes can occur on the file system but they won't be reflected on the TreeMap.

Flash window after scan if hidden

If you start a scan and minimize the window, at scan end then the window will flash in the task bar, to warn you.

Keep element tooltip open on change

When active and tooltip is visible, if the mouse moves to another element the tooltip keeps visible and its content changes. When inactive and the tooltip is visible, a mouse movement to another element hides the tooltip. Then, you need to wait a moment for it to be visible again.

Show file date and time

Activates `creation`/`modify`/`access` dates ("dd/mm/yyyy" or whatever your O.S. date format is used) visualization in the tooltip (only for files).

Show file age

Activates `creation`/`modify`/`access` ages ("1 year 3 months") visualization in the tooltip (only for files).

Popup log console on event

If a relevant event happens, the log console will popup. You can also open the log console from the main window drop down menu. The log console itself features a deactivation toolbar button (the one with the lightning) that stops the console to annoyingly popup at each event and closes it until restart or manually reopened. This way if you want to be aware of events, you can keep the popup option active and close/deactivate the console at first popup.

Log scan warnings

By enabling this option, you tell Space*Sniffer* to log events when it cannot scan a folder or a file due to any reason.

8.4 - Look tab

Level contrast

This setting lets you choose from a completely "flat" color scheme to a "deep view" color scheme. Light levels are controlled by the nesting.

Border contrast

Makes elements borders more or less noticeable. This lets you to vary from a "gummy" look to a "hard edge" look.

Highlight halo levels

When the mouse hovers on an element, this is highlighted. As the the halo levels setting increase, more parent elements are affected by the highlighting. If you exaggerate, the effect can be pretty psychedelic.

Mouse hover

Highlights the element under the mouse pointer.

Drop shadow

Selected element will drop a shadow to make it easy to visually track, expecially during the scan process.

Mouse trail

Leaves an highlight trail on elements when mouse moves. This can be annoying to some users. In case, deactivate.

8.5 - Styles tab

This tab contains more tabs, one for each available visualization style. You can define the default style to be used at each new view. When a view is opened, you can control the style from the toolbar.

Flat Colors tab

Here you can specify your preferred color for the `drive`, `folder`, `file`, `free space` and `unknown space` entities. Those are the base colors, but they will be darkened to show nesting accordingly to the Level Contrast parameter.

File Classes tab

Here you can define as many file classes as you wish. Each file class has a description and a list of file extensions (separated by the "`;`" char). Finally, there is a `color`.

When you use the File Classes style, matching files are displayed with the belonging class color. If the same file extension exists in more than one class, the choosed color is the color of the first matching class.

This style can help to understand the file type distribution on the analyzed media, but it can also clutter the view with too many colors. Hence it can be easily turned on/off directly from the toolbar.

9 . T h e   E x p o r t   M o d u l e

The export module is useful when you need to produce a file report of **the current filtered zoomed view**. This means that only files starting at the current zoom level and shown by the filter rules are considered for the output.

The export module has two working modes. The first one lets you to save a Space*Sniffer* binary snapshot, which roughly is a dump of the internal Space*Sniffer* memory structure, the second one lets you export in text mode and gives you the chance to customize the wanted output.

9.1 - SpaceSniffer binary snapshot

This format stores pretty every information that Space*Sniffer* has gathered during the scan operation, plus other information like the tags applied to files and folders. So, you can send a scan snapshot to someone else along the tags information to simplify other's analysis. While other formats are currently supported for loading, this is the only one that guarantees the maximum information level storage.

9.2 - Text mode export

Here you can customize the output format with Space*Sniffer* internal formatting mini scripting language. While text exported data can be used everywhere you need to, it's difficult to get them again into Space*Sniffer*. So if you plan to reload the scan data later, please consider using the internal binary format.

9.3 - How to proceed

The export procedure is as follows:

- First of all, you isolate the files that you want to report by **navigating and filtering** the media structure.

- Then you activate the export module by `File`/`Export` menu command.

- The export window displays. Then you can choose an export configuration from the drop down list. Some built-in configurations are provided for your convenience, along the Space*Sniffer* binary snapshot which is always the first choice.

- After choosing the export configuration, the preview panel will display a short preview of the output (not for the binary format which would be unreadable). Notice that only the first files/folders are considered for the preview production, so the preview will likely show a partial result. The preview may show more or less data based on the script structure.

- By pressing the `Export` button on the tool bar you start the export procedure.

10. Export Customization

The Space**Sniffer** export module is capable of producing text exports. The export engine is customizable and lets you store many export configurations, one for each specific need.

When the export module is first shown in a Space**Sniffer** work session, it is in "`run time`" mode. This means that it's hiding all its complexity and tries to appear as simple as possible to the user. This is the default behavior and all users except who wants to customize the export configuration should use this mode.

To enter customization mode, you need to click the small wrench icon next the configuration drop down list. Then, the "`design time`" mode is engaged. If you press again the wrench icon you switch back to "`run time`" mode.

*Note: customization mode is not available for the Space**Sniffer** binary snapshot. If you press the wrench button with Space**Sniffer** binary snapshot selected, the export module will move on next editable configuration. If no editable configuration exists, a new one will be created.*

The `design time mode` shows more icons on the upper tool bar and a new design section in the middle of the window. This section is splitted in many parts, each one reachable by its own tab.

This is where you customize your configuration. The first tab contains global informations about the export configuration. The following tabs contain the export configuration itself. There is a tab for the `header`, a tab for the `detail` and a tab for the `footer`.

10.1 - The global information panel

This panel contains global information about the current configuration. You can specify the preferred `file name` and `extension` (there is a syntax to correctly define the file extension, please have a look directly at the example in the panel).

You can also specify the primary and secondary sorting:

Block sorting

It's the primary sorting, where you specify if, for each folder, you want listed first all sub folders, first all contained files or if you don't care.

Fine sorting

It's the secondary sorting, where you specify how to sort folders and files. If you don't care,

you can specify here. The descending check box is used to invert the sorting.

The header, detail, footer panels

All these three panels work the same. The header is used to define a report header, the footer to define a report footer and the detail is used for each folder or file to be exported.

The only difference between the three panels is that for header and footer panels only the information of the root container is available (consider the root as the currently zoomed element in the active view). Let's now consider the detail panel.

There is a small text editor where you can type the text that will be exported to file. If you type simple text, this will be exported as-is, without modifications. Even tabs and newlines are considered for the export. When you type something, the preview panel reflects your changes.

Typing simple text is boring. What we want is to export file and folder data. Here we introduce the two main elements of the scripting engine:

- `Tags`: a tag is a keyword surrounded by `<%` and `%>`. For example, `<%path%>`. When you type a tag into a script, it is expanded with the corresponding value of the folder or file. There are lots of tags, just explore all of them by right clicking the mouse over the script editor. A popup menu will appear listing all tags by category.

- `Commands`: while tags contain textual information about the exported element, command are used to adapt the output to your needs. Commands are always enclosed in `{...}` chars. There are commands for formatting, for testing and so on. You can find all available commands by right clicking the mouse on the script editor.

💡*Hint: Commands can be nested to get the desired result.*

10.2 - Tags and commands description

Here follows a quick description for tags and commands that may be used inside export scripts:

Tags

- `<%pathfile%>`: full `filename`, `path` and `extension` included.
- `<%path%>`: only the `path` part.
- `<%file%>`: only the `filename` part, without `extension` and the separating dot.
- `<%filename%>`: only the `filename` part, with `extension`.
- `<%fileext%>`: only the file `extension`, without the separating dot.
- `<%size%>`: descriptive `size` (logical) of the element.

- `<%sizebytes%>`: unformatted `size` (logical) in bytes of the element.

- `<%disksize%>`: descriptive `size` (on disk) of the element.

- `<%disksizebytes%>`: descriptive `size` (on disk) of the element.

- `<%file(create|modify|access)date%>`: `creation|modify|acces` date.

- `<%file(create|modify|access)time%>`: `creation|modify|acces` time.

- `<%file(create|modify|access)age%>`: `creation|modify|acces` age description.

- `<%isfile%>`: `true` if the element is a file, `false` otherwise.

- `<%isfolder%>`: `true` if the element is a folder, `false` otherwise.

- `<%iscontainer%>`: `true` if the element is a container (disk or folder), `false` otherwise.

- `<%containsfiles%>`: `true` if the element contains files, `false` otherwise.

- `<%containsfolders%>`: `true` if the element contains folders, `false` otherwise.

- `<%containscontainers%>`: `true` if the element contains containers, `false` otherwise.

- `<%nestinglevel%>`: nesting level, starting from `0`.

- `<%count%>`: progressive absolute counter (increments on every element). Starts from `0`.

- `<%localcount%>`: progressive local counter (increments on every element, each container has a local counter). Starts from `0`.

- `<%filecount%>`: progressive file absolute counter (increments on every file). Starts from `0` .

- `<%localfilecount%>`: progressive file local counter (increments on every file, each container has a local counter). Starts from `0`.

- `<%foldercount%>`: progressive folder absolute counter (increments on every folder). Starts from `0`.

- `<%localfoldercount%>`: progressive folder absolute counter (increments on every folder, each container has a local counter). Starts from `0`.

- `<%containercount%>`: progressive container absolute counter (increments on every folder). Starts from `0`.

- `<%localcontainercount%>`: progressive container absolute counter (increments on every folder, each container has a local counter). Starts from `0`.

Commands

- `{&br}`: newline.

- `{&tab}`: tab char.

- `{leftpad{text}{padtext}{length}}`: fills `text` with `padtext` on the left until `length` is reached. If `text` overflows `length` it is left truncated.

- `{rightpad{text}{padtext}{length}}`: fills `text` with `padtext` on the right until `length` is reached. If `text` overflows `length` it is right truncated.

- `{&curlyop}`: "{" char.

- `{&curlycl}`: "}" char.

- `{&angleop}`: "<" char.

- `{&anglecl}`: ">" char.

- `{script}`: you may notice that everything you type is exported. So you cannot indent your code because the indentation will be exported. This mode is suited for very simple scripts but when you need to indent your code you need to switch to the "`script`" mode, by inserting the `{script}` command at the start of the script. This command will disable the exporting of tabs and newlines. At this point, if you really need to export a tab or a newline, you must use the `{&tab}` and `{&br}` commands. These commands work also in simple mode, they just mix with normal tabs and newlines.

- `{nest{text}}`: `text` is repeated `N` times where `N` is the nesting level.

- `{if{value1}(==|!=|<|<=|>|>=){value2}?{result}[:{else}]}`: this is a conditional operator. You can compare `value1` with `value2` (numeric or string) with equality and inequality operators. If the comparison gives true, `result` is produced, otherwise `else` is produced. If `else` is not provided, empty text is produced. First, a numeric comparison is tried. If the numeric conversion fails, then a string comparison is performed.

💡 *Hint: While you can build a complete filtering system through the export script, it's not the best way to face the filtering issue. Since the export module exports only elements filtered by the scan view filter field, it is advisable to use both the filter field and the export script to get to the wanted result. This will lead to a simpler script and a more efficient export operation because while the script commands are interpreted for each element, the filter field is a builtin function and will work faster.*

This is a really short introduction to the scripting engine. Hope this is useful to introduce you to the export customization.

Now, let's experiment, *have fun*!

<div style="background:gray">

11. Command Line Parameters

</div>

You can start Space*Sniffer* also through the command prompt. Space*Sniffer* can understand some parameters commands. To view all the possibilities you can type in a command prompt:

```
SpaceSniffer.exe help
```

Or you can choose the "`Command line help`" in the "`About`" menu of the main window.

This will display the list of available commands, along with a concise explanation of how they work:

- `help`: shows the help message itself.

- `scan "<path>"`: opens a new scan view on `<path>`. You can specify more than one scan command or you can separate paths by '`;`' char.

- `load "<file>"`: opens a snapshot view on `<file>` as long as Space*Sniffer* recognizes its format/extension. You can specify more than one load command or you can separate paths by '`;`' char.

- `filter "<filter>"`: applies `<filter>` to the preceeding `scan`/`load` command.

- `export "<name>" "<path>"`: applies named `<name>` export on `<path>` file to the preceeding `scan`/`load` command. `<name>` is a (case sensitive) export config name. Any filter, stop or similar UI commands will deactivate the auto export function.

- `save "<path>"`: same as export but you don't need to specify the export configuration name as the internal SpaceSniffer binary format is implicit.

- `autoclose`: auto closes Space*Sniffer* after all auto exports. Any filter, stop or similar UI commands will deactivate the auto close function.

11.1 - Command line by example

Let's explain how to drive Space*Sniffer* through parameters:

```
SpaceSniffer.exe scan c:\
```

will start a single scan on drive c:\

```
SpaceSniffer.exe scan c:\;d:\
```

will start two scan views on respective paths.

```
SpaceSniffer.exe scan "c:\Program Files;c:\Windows"
```

you can type also a complete path, just consider that if the path contains spaces, all the compound paths parameter must be surrounded with "" chars.

```
SpaceSniffer.exe scan c:\ filter *.jpg;*.gif
```

the filter command will apply to the previous scan command. So this will scan the c:\ drive for JPEGs and GIFs.

```
SpaceSniffer.exe scan c:\ filter *.jpg scan d:\;e:\ filter *.gif;>100kb
```

got the idea?

11.2 - Export command

It is possible to automatically export scan result at the end of the scan operation.

```
SpaceSniffer.exe scan c:\ filter *.jpg export "Grouped by folder" c:\export.txt
```

You can also export a loaded snapshot. This way you can load a snapshot file, filter it then export to another snapshot file.

You can automatically export a view by using the `export` command. The `export` command needs two parameters: an export configuration name and a destination file name.

📑**Note:** *The export configuration name is simply the name you give to an export configuration as seen in the export dialog. If export name or the file path contain spaces, you need to enclose them with double quotes, respectively.*

💡**Hint:** *You can specify zero or more* `export` *commands for each scan command. This means that you can start two scans and export them with two different export commands. The behavior is similar to* `filter` *command.*

💡**Hint:** *If you want to export the same scan with two different* `export` *commands, please specify the* `scan` *command twice on the same path. The smart cache engine will scan the path once (but produce two scan views, each with a different* `export` *command attached to them).*

📑**Note:** *the export phase will automatically begin at the end of all scan operations. All export operations are serialized to avoid disk trashing. If the user cancels a scan operation or an export operation, all remaining exports will be aborted.*

```
SpaceSniffer.exe scan c:\ export "Grouped by folder" c:\export.txt autoclose
```

By specifying the `autoclose` command, Space**Sniffer** will terminate itself after all export operations are finished.

📔 **Note:** *If there is no export command or the user cancels a scan or an export operation, the autoclose command will be ignored.*

1 2 .   F i n a l   N o t e s

12.1 - File handling

Space*Sniffer* algorithms works in read only mode. It is possible to access the Windows Explorer popup menu for a folder or a file by right mouse clicking on a view element. So, if you delete a file it's because you deleted it through Windows Explorer functions, not Space*Sniffer*'s. The only exception is when Space*Sniffer* saves the configuration into his configuration XML file. If it's possible to save, then it will be done. On the contrary, no saving is performed and the program quietly accepts the fact (you will not be able to persist your customizations).

12.2 - Privilege request

Whenever possible, Space*Sniffer* tries to get the Backup Operator privilege, that gives the possibility to examine protected folders. If the application can obtain this right or not depends on your privileges in the O.S. If it's not possible to get this privilege, the application still works but you'll probably notice that some unknown space will stay in the way also after the end of the scan process.

12.3 - File system events

Space*Sniffer* listens to file system events, so it can reflect changes made outside of the application. This feature may not be supported, depending on the O.S. and media type.

12.4 - Internet connection

Space*Sniffer* plays polite. It will never attempt to connect to the internet by itself. The only exceptions are the "`Check for update`" option, that will open the download page letting you check if a new version has been released, the "`Support this project!`" option, that will open the donation page in your default browser and the "`About Box`" that contains a link to "www.uderzo.it". So, if Space*Sniffer* connects to the internet, it's because you told it to do so.

12.5 - Compatibility

The x64 Space*Sniffer* version requires at least Windows 7 (64bit) systems. There is no plan to maintain a 32 bit version.

12.6 - Contact infos

If you want to contact me for information or to submit hints or bug reports, please write to info@uderzo.it and specify the keyword `spacesniffer` in the mail subject. Remember to specify the keyword or your mail could be blocked by the antispam filter.

12.7 - Please support!

If you feel this software comes handy to you, please consider donating to the project. By donating, you show me your appreciation and encourage further product development. You can do so on the donation internet page by choosing the donation tool bar button or the "`Support this project!`" option in the help menu. <u>Thank you for your support</u>.

Copyright © 2007-2025 Uderzo Umberto, please read the Disclaimer