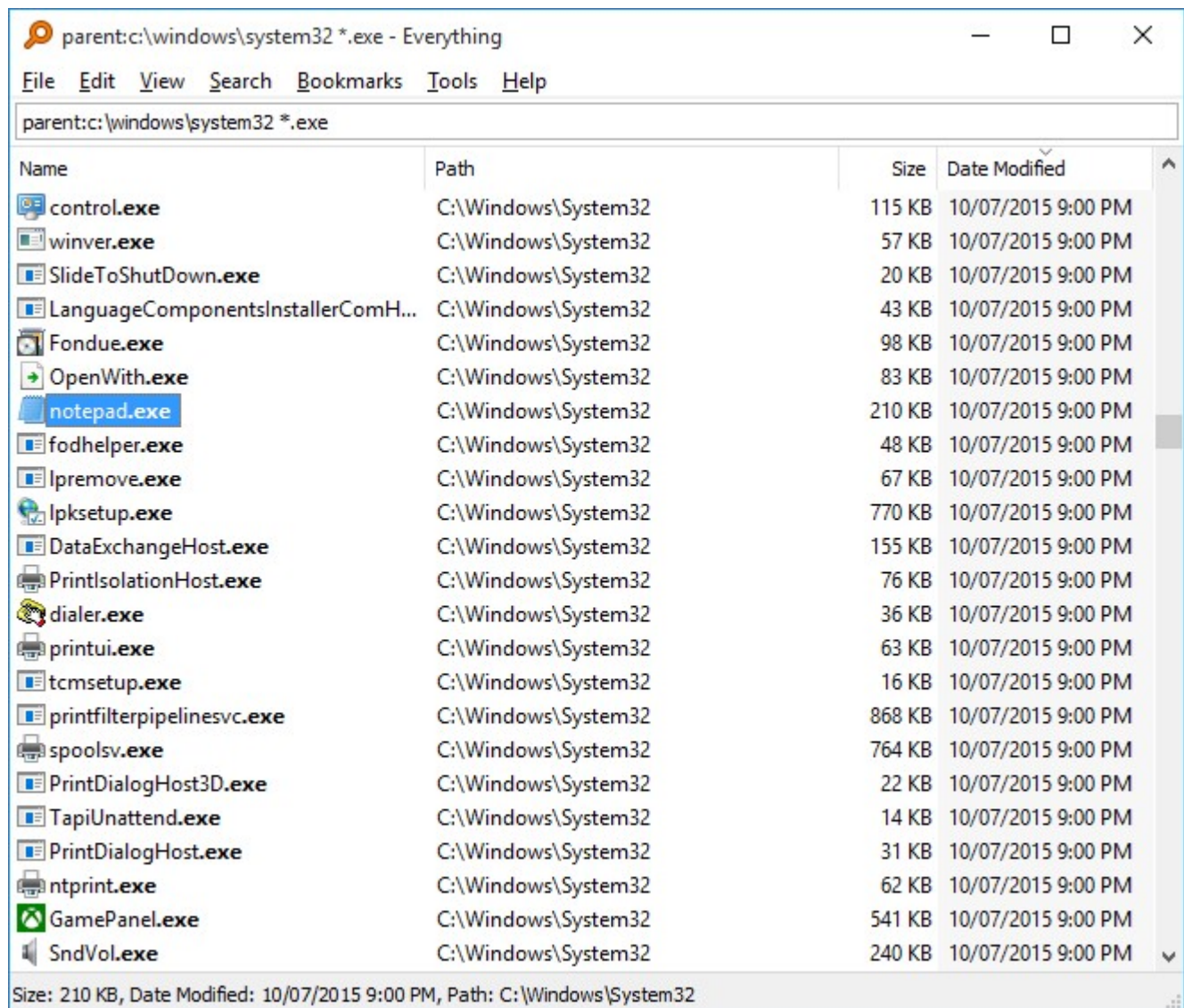




Everything



"Everything" is a filename search engine for Windows.

How is Everything different from other search engines

- Small installation file.
- Clean and simple user interface.
- Quick file indexing.
- Quick searching.
- Quick startup.
- Minimal resource usage.

- Small database on disk.
- Real-time updating.

See also

- [Installing Everything](#)
- [Using Everything](#)
- [Download offline Compiled HTML help \(chm\)](#)
- [Download offline HTML help \(html\)](#)

Installing Everything

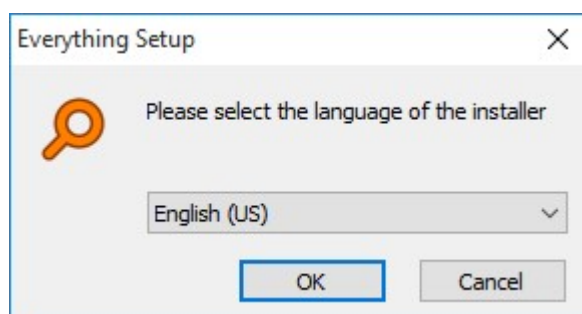
Everything can be installed or run as a portable executable.

The installer and portable version of Everything can be found on the [downloads page](#).

What version of Everything should I download?

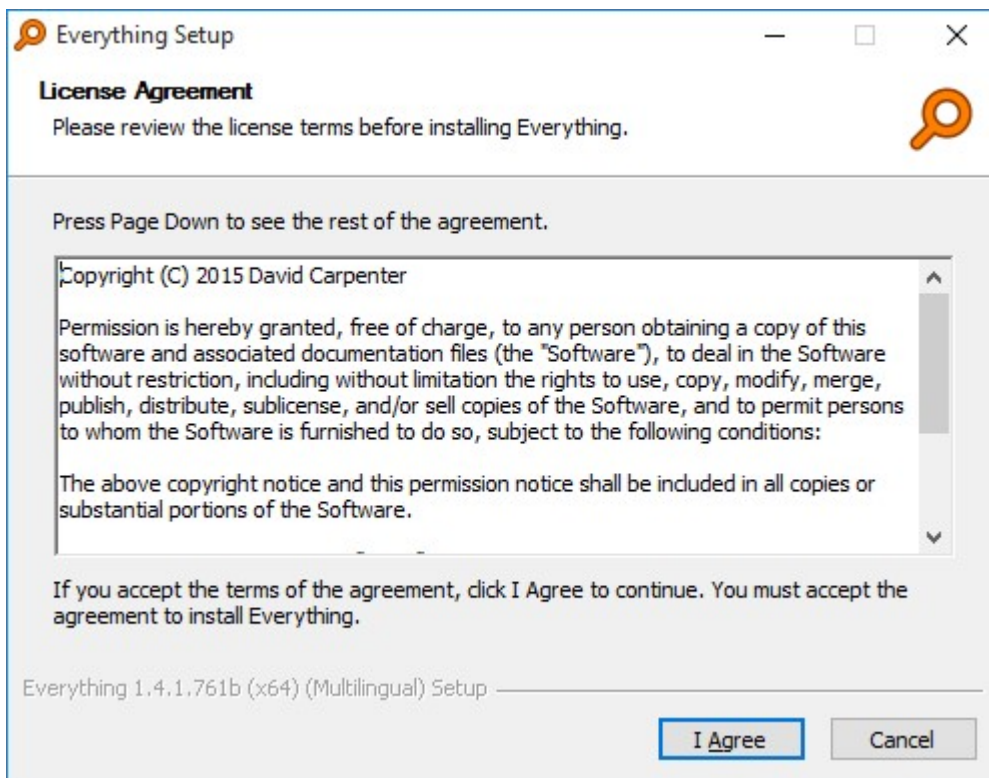
Version	Description
Installer	The installer version will guide the user through the installation of Everything (recommended). The installed Everything.exe is the same as the portable Everything.exe.
Portable	The user must extract the Everything.exe to the desired location. No changes to the system are made. Most of the options in the installer are available in the portable version in the Everything Options .
x86	Only x86 context menu extensions will work in Everything. This version will work on both x86 and x64 versions of Windows.
x64	The x64 version requires a 64bit x64 OS. Supports databases larger than 2GB. Only x64 context menu extensions will work in Everything. The x64 version will use more memory than the x86 version.
English (US)	Only English is available for this version, the language pack can be installed manually.
Multilingual	Includes the Everything language pack for multiple language support.
Lite	This light version does not contain ETP/FTP/HTTP servers and does not allow IPC .

Language selection



Choose the language of the installer. This language will also be used in Everything. See [Supported Languages](#) for a list of supported languages.

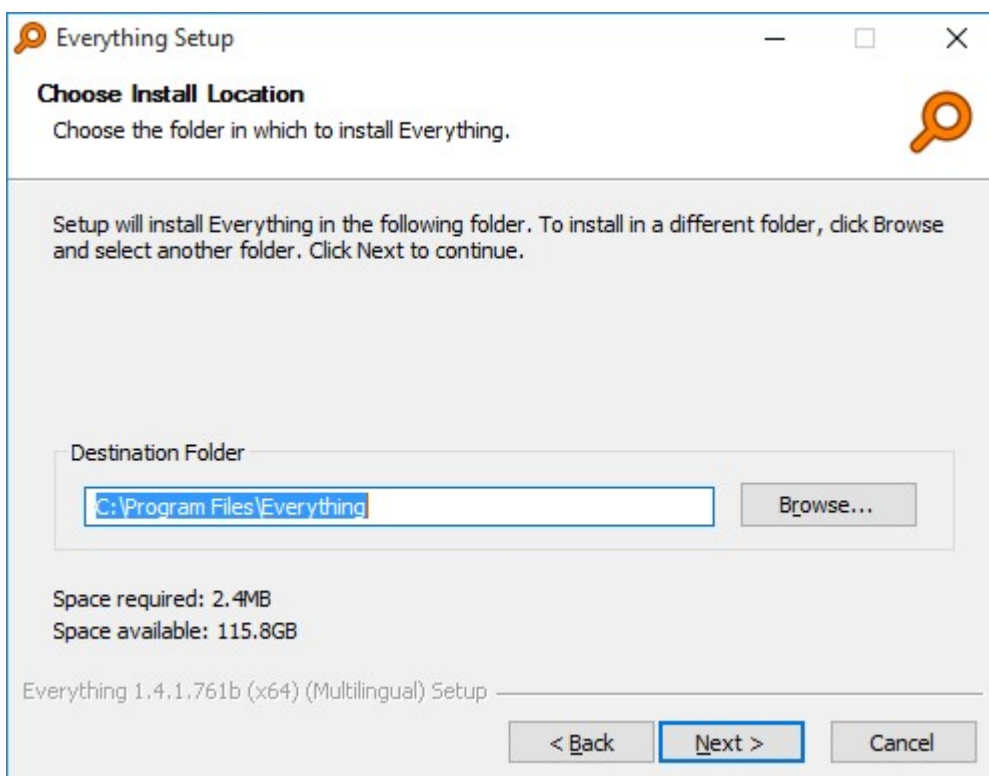
License Agreement



You must accept the agreement to install Everything.

You can find a copy of the license [here](#).

Install Location



Choose the installation location of Everything.

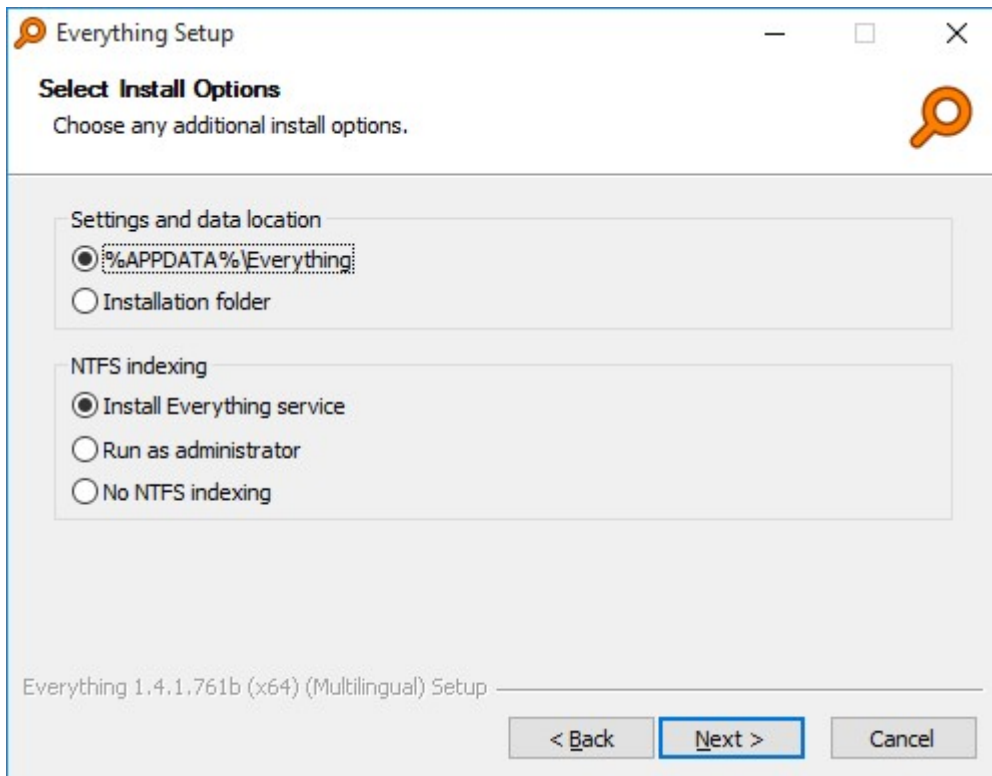
By default Everything is installed to:

C:\Program Files\Everything

or if you install the x86 version of "Everything" on a x64 OS:

c:\Program Files (x86)\Everything

Location for data and settings



Choose a location to store data and settings.

%APPDATA%\Everything is recommended.

This setting can be changed in the [Everything options](#) at anytime.

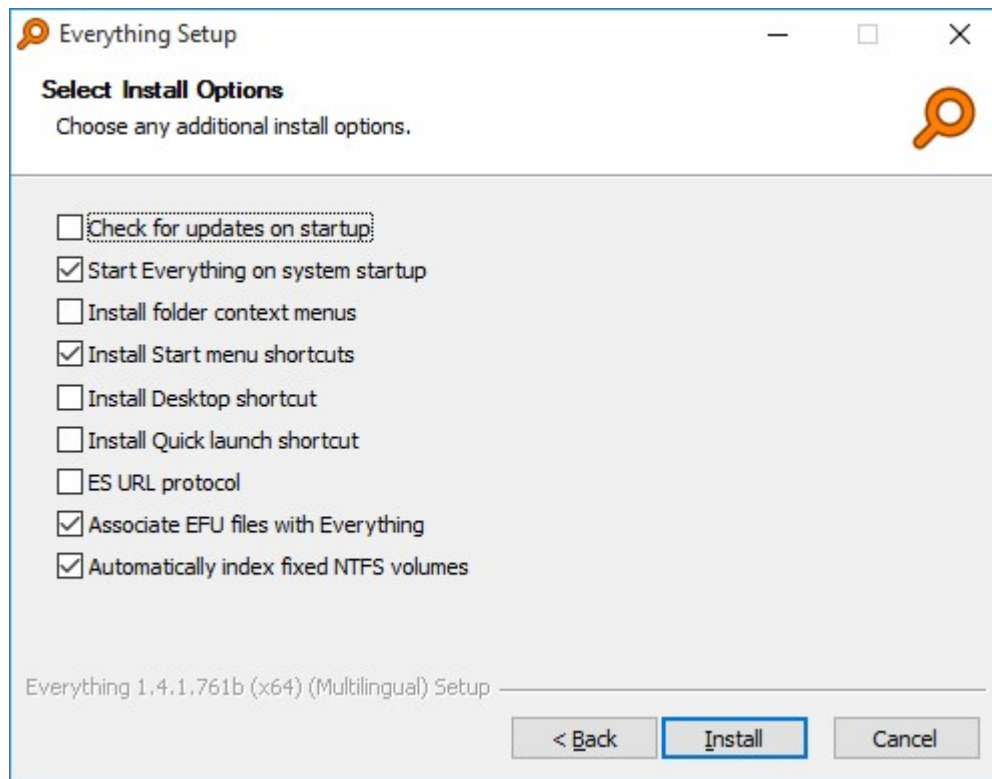
Indexing NTFS volumes

Choose a method for indexing NTFS volumes

Method	Description
Install Everything service	Install the Everything service which will do all the low level volume reading required for NTFS indexing. The service runs with the necessary privileges so the Everything client can run as a standard user account.
Run as administrator	Run Everything with administrative privileges.
No NTFS indexing	Everything will not attempt to index NTFS volumes.

This setting can be changed in the [Everything options](#) at anytime.

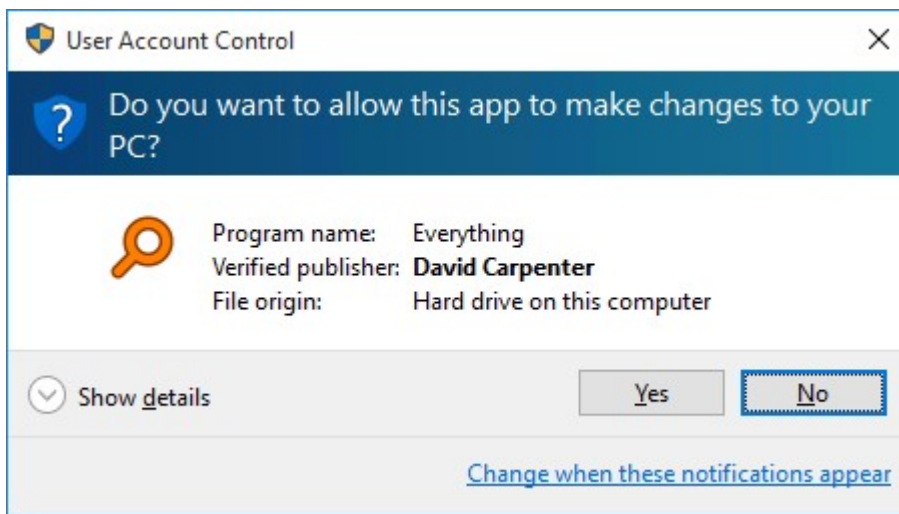
Install Options



Option	Description
Check for updates	Enabling this setting will make Everything check www.voidtools.com for updates when you run Everything.
Start Everything on system startup	Run Everything when Windows starts. This will load the Everything database into memory and create the Everything tray icon.
Install folder context menus	Enabling this will show the context menu item "Search Everything..." when you right click a folder in Windows. Clicking on this context menu item will show the Everything window with the search set to the selected folder.
Install start menu items.	Creates the Search Everything and Uninstall Everything shortcuts in the Start menu.
Install Desktop shortcut	Creates the Search Everything shortcut on the Desktop.
Install Quick launch shortcut	Creates the Search Everything shortcut in the Quick launch toolbar.
ES URL Protocol	Open Everything from es: urls.
Associate EFU files with Everything	Associate Everything file lists (EFU files) with Everything.
Automatically index fixed NTFS volumes	If enabled Everything will automatically add all NTFS volumes to the index. Disable to manually setup which volumes are included.

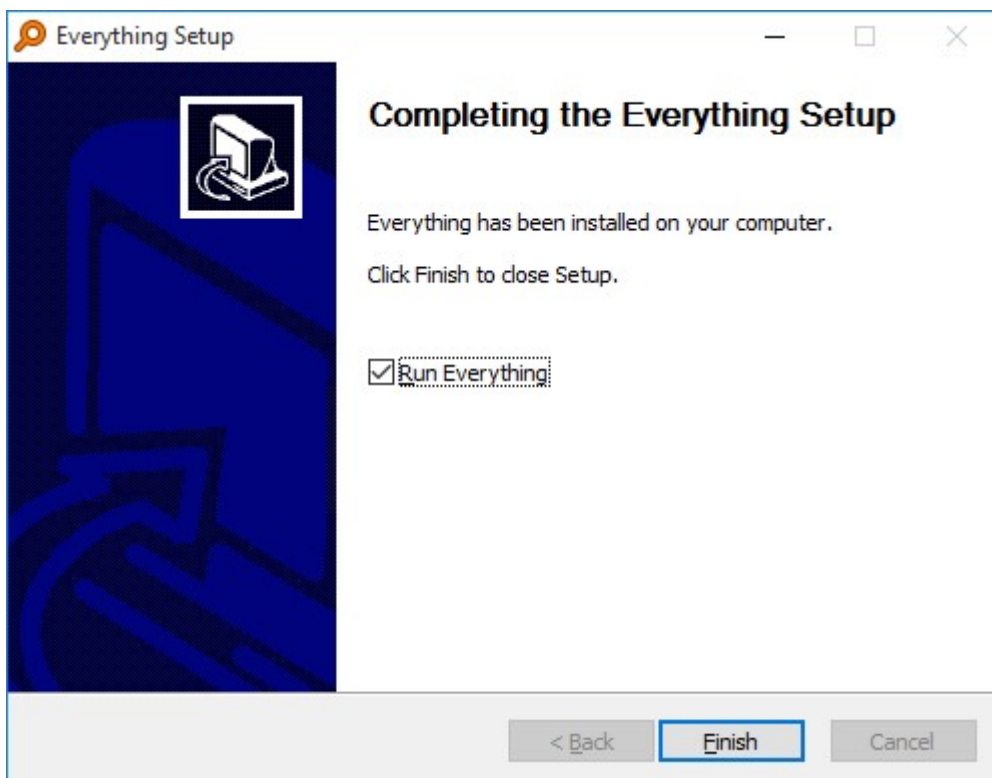
These settings can be changed in the [Everything options](#) at anytime.

UAC



Everything installation requires administrative privileges.
Click Yes to continue installation with administrative privileges.

Finish



The installation of Everything is complete. Click Finish to launch Everything.
When Everything launches it will begin creating the initial index which will take a few seconds.
For a basic guide on using Everything, please see [Using Everything](#).

Troubleshooting

Error	Description
OS is not x64 Install anyway?	This error is displayed when attempting to install the x64 version of Everything on a 32bit OS. Please make sure you have the correct x86 or x64 version of Everything for your OS (See What version of Everything should I download?).
Failed to execute admin command	The installer failed to run Everything as an administrator to make changes to the system. Please make sure Everything can run with elevated privileges.
Failed to execute user command	The installer failed to run Everything to install the options for the current user. This could be due to anti-virus software. Please make sure the installer can execute Everything from the temp directory.

See also

- [Using Everything](#)
- [Options](#)
- [Uninstalling Everything](#)

Using Everything

An index of all your local NTFS volumes is created when running Everything for the first time.

Indexing can take a few seconds.

Once the indexing stage is complete, All your files will be displayed.

Type in a partial filename in the search box at the top to begin.

parent:c:\windows\system32 *.exe - Everything

File Edit View Search Bookmarks Tools Help

parent:c:\windows\system32 *.exe

Name	Path	Size	Date Modified
control.exe	C:\Windows\System32	115 KB	10/07/2015 9:00 PM
winver.exe	C:\Windows\System32	57 KB	10/07/2015 9:00 PM
SlideToShutDown.exe	C:\Windows\System32	20 KB	10/07/2015 9:00 PM
LanguageComponentsInstallerComH...	C:\Windows\System32	43 KB	10/07/2015 9:00 PM
Fondue.exe	C:\Windows\System32	98 KB	10/07/2015 9:00 PM
OpenWith.exe	C:\Windows\System32	83 KB	10/07/2015 9:00 PM
notepad.exe	C:\Windows\System32	210 KB	10/07/2015 9:00 PM
fodhelper.exe	C:\Windows\System32	48 KB	10/07/2015 9:00 PM
lpremove.exe	C:\Windows\System32	67 KB	10/07/2015 9:00 PM
lpksetup.exe	C:\Windows\System32	770 KB	10/07/2015 9:00 PM
DataExchangeHost.exe	C:\Windows\System32	155 KB	10/07/2015 9:00 PM
PrintIsolationHost.exe	C:\Windows\System32	76 KB	10/07/2015 9:00 PM
dialer.exe	C:\Windows\System32	36 KB	10/07/2015 9:00 PM
printui.exe	C:\Windows\System32	63 KB	10/07/2015 9:00 PM
tcmsetup.exe	C:\Windows\System32	16 KB	10/07/2015 9:00 PM
printfilterpipelinesvc.exe	C:\Windows\System32	868 KB	10/07/2015 9:00 PM
spoolsv.exe	C:\Windows\System32	764 KB	10/07/2015 9:00 PM
PrintDialogHost3D.exe	C:\Windows\System32	22 KB	10/07/2015 9:00 PM
TapiUnattend.exe	C:\Windows\System32	14 KB	10/07/2015 9:00 PM
PrintDialogHost.exe	C:\Windows\System32	31 KB	10/07/2015 9:00 PM
ntprint.exe	C:\Windows\System32	62 KB	10/07/2015 9:00 PM
GamePanel.exe	C:\Windows\System32	541 KB	10/07/2015 9:00 PM
SndVol.exe	C:\Windows\System32	240 KB	10/07/2015 9:00 PM

Size: 210 KB, Date Modified: 10/07/2015 9:00 PM, Path: C:\Windows\System32

The Everything search window contains the following parts:

Menu

The menu contains commands for working with the Everything search window, results, view, search, bookmarks, tools and help.

Search edit

Type in the partial filename to locate a file or folder.

Use the [Advanced Search](#) from the Search menu to perform more complex searches.

See [Searching](#) for the complete search syntax.

Basic search syntax can be displayed in Everything from the Help menu.

Result list header

Click a column header to sort the results by that column. Click the same column header to change the direction of the sort.

Right click to show or hide different columns.

Result list

Double click or select an item and press Enter to open it.

Items can be dragged and dropped on another program.

Right click on an item to show a list of actions for the selected results.

See [Results](#) for more information.

Status bar

Displays the current status of Everything and the current search options.

Right click the status bar to change the search options.

Double click a search option to disable it.

Show an Everything search window

To show the search window:

- Double click the [Everything tray icon](#).

-or-

- Use a [Hotkey](#).

-or-

- Run Everything from a shortcut, such as the Everything Desktop shortcut, Everything start menu shortcut or Everything quick launch shortcut.

Normally, Everything will only use one search window.

To allow multiple windows, enable [Create a new window from the system tray](#) and/or [Create a new window when running Everything](#).

Find recently modified files instantly and monitor file system changes real-time

For example, to find files and folders modified today, search for:

```
dm:today
```

Right click an empty spot in the result list and click Sort by -> Date Modified to see changes to the file system in real-time.

Open frequently run files easily

Everything remembers the run count for each result.

Press Enter from the search edit to select the result with the highest run count.

Right click a file or folder to manually set the run count.

See [Run History](#) for more information.

To sort by run count, right click an empty spot in the result list and click Sort by -> Run Count.

Export results

Results can be exported to CSV, TXT or [EFU](#) files.

To export results:

- In **Everything**, From the **File** menu, click **Export....**

Filters

Filters are predefined searches, such as audio and video files types.

To select a filter, click a filter from the Search menu. The selected filter will show in the status bar. Double click the filter in the status bar to disable it.

To show the filter bar, click Filters from the View menu.

See [Customizing filters](#) for more information.

Bookmarks

Save the current search, filter, sort and index so it can be quickly restored later.

See [Bookmarks](#) for more information.

Access Everything from a web browser

You can start your own web server in Everything to access your files from your phone or other device.

See [HTTP Server](#) for more information.

Access Everything from another computer

You can start your own ETP server in Everything to access your files from another computer.

See [ETP Server](#) for more information.

Change fonts and colors

The result list font and colors can be customized in the [Everything options](#).

To customize all fonts and colors, see [Everything.ini](#).

External File Manager

To open folders with a custom file manager, see [External File Manager](#).

Keyboard shortcuts

For a list of existing keyboard shortcuts, see [Keyboard Shortcuts](#).

See [Customizing keyboard shortcuts](#) for more information.

Indexes

Include local NTFS volumes, any folders and/or file lists in the index.

See [Indexes](#) for more information.

File lists

Create a snap shot of files on a NAS, CD, DVD or BluRay and include them in your index.

To show the built in file list editor, from the Tools menu, click File List Editor....

See [File Lists](#) for more information.

Options

Customize the installation of Everything, user interface, home search, search settings, how the results are displayed, how the results look, right click menus, fonts and colors, keyboard shortcuts, search and run history, ntfs volumes, folder indexing, file lists, excluded folders and file types, ETP and HTTP servers.

See [options](#) for more information.

Exit Everything

When you close the Everything search window, Everything will continue to run in the background.

To exit Everything completely, click Exit from the File menu or right click the Everything tray icon and click Exit.

To make Everything exit when the Everything search window is closed:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **UI** tab.
- Uncheck **Run in background**.
- Click **OK**.

See also

- [Searching](#)
- [Results](#)
- [Indexes](#)

Command Line Interface

ES is a command line interface to search Everything from a command prompt.

```
C:\Windows\system32\cmd.exe

C:\Users\limited\Desktop>es -size -dm -sizecolor 13 -dmcolor 11 -sort dm -n 10
  78,440 26/10/2016 20:02 C:\Users\limited\Desktop\es.exe
 2,023,424 26/10/2016 13:57 C:\Users\limited\Desktop\Everything.exe
  32,768 21/10/2016 16:37 C:\Boot\BCD
 1,048,576 21/10/2016 16:37 C:\Users\limited\NTUSER.DAT
  53,767 21/10/2016 16:37 C:\Users\limited\AppData\Local\IconCache.db
268,435,456 21/10/2016 16:08 C:\swapfile.sys
738,197,504 21/10/2016 16:08 C:\pagefile.sys
 387,332 19/10/2016 19:31 C:\Users\limited\Desktop\Everything-Debug.txt
   181 19/10/2016 18:47 C:\Users\limited\Desktop\Everything.ini
 7,328,537 14/10/2016 23:25 C:\dev\Everything.db

C:\Users\limited\Desktop>_
```

Download

ES can be downloaded from the [Downloads page](#).

Requirements

- Everything must be installed and running.

Usage

```
es.exe [options] [search text]
```

[option]

Optional option.

<option>

Required option.

Search syntax

ES uses the [Everything search syntax](#).

General Command Line Options

These options will work with any version of Everything.

-r <search>

-regex <search>

Search using regular expressions.

Escape spaces with double quotes.

-i

-case

Match case.

-w

-ww

-whole-word

-whole-words

Match whole words.

-p

-match-path

Match full path and file name.

-h

-help

Display this help.

-o <offset>

-offset <offset>

Show results starting from the zero based offset.

-n <num>

-max-results <num>

Limit the number of results shown to <num>.

-s

Sort by full path.

Everything 1.4 Command Line Options

The following switches require Everything 1.4 or later.

-a

-diacritics

Match diacritical marks.

- name
- path-column
- full-path-and-name
- filename-column
- extension
- ext
- size
- date-created
- dc
- date-modified
- dm
- date-accessed
- da
- attributes
- attribs
- attrib
- file-list-file-name
- run-count
- date-run
- date-recently-changed
- rc

Show the specified column. The full path and name column will be used if no name, path or full path and name column is specified.

Columns are displayed in the order specified.

- sort name
- sort path
- sort size
- sort extension
- sort date-created
- sort date-modified
- sort date-accessed
- sort attributes
- sort file-list-file-name
- sort run-count
- sort date-recently-changed
- sort date-run
- sort-name
- sort-path
- sort-size
- sort-extension
- sort-date-created
- sort-date-modified
- sort-date-accessed
- sort-attributes
- sort-file-list-file-name
- sort-run-count
- sort-date-recently-changed
- sort-date-run
- sort name-ascending
- sort name-descending
- sort path-ascending
- sort path-descending

- sort size-ascending
- sort size-descending
- sort extension-ascending
- sort extension-descending
- sort date-created-ascending
- sort date-created-descending
- sort date-modified-ascending
- sort date-modified-descending
- sort date-accessed-ascending
- sort date-accessed-descending
- sort attributes-ascending
- sort attributes-descending
- sort file-list-file-name-ascending
- sort file-list-file-name-descending
- sort run-count-ascending
- sort run-count-descending
- sort date-recently-changed-ascending
- sort date-recently-changed-descending
- sort date-run-ascending
- sort date-run-descending
- sort-name-ascending
- sort-name-descending
- sort-path-ascending
- sort-path-descending
- sort-size-ascending
- sort-size-descending
- sort-extension-ascending
- sort-extension-descending
- sort-date-created-ascending
- sort-date-created-descending
- sort-date-modified-ascending
- sort-date-modified-descending
- sort-date-accessed-ascending
- sort-date-accessed-descending
- sort-attributes-ascending
- sort-attributes-descending
- sort-file-list-file-name-ascending
- sort-file-list-file-name-descending
- sort-run-count-ascending
- sort-run-count-descending
- sort-date-recently-changed-ascending
- sort-date-recently-changed-descending
- sort-date-run-ascending
- sort-date-run-descending

Specify the sort. Results are sorted by name ascending if not specified.

- sort-ascending
- sort-descending

Set sort order. For example, sort by size ascending: -sort size -sort-ascending

If the sort order is not specified: size is sorted largest first, dates latest first and run count largest first, All other sorts are sorted alphabetically.

-instance <name>

Connect to the unique Everything instance name.

See [Multiple Instances](#) for more information.

-highlight

Highlight results.

Highlighting many results can decrease the performance of "Everything".

-highlight-color <color>

<color> can be one of the following color codes:

	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x10		0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
0x20	0x21		0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
0x30	0x31	0x32		0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
0x40	0x41	0x42	0x43		0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
0x50	0x51	0x52	0x53	0x54		0x56	0x57	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
0x60	0x61	0x62	0x63	0x64	0x65		0x67	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
0x70	0x71	0x72	0x73	0x74	0x75	0x76		0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F
0x80	0x81	0x82	0x83	0x84	0x85	0x86	0x87		0x89	0x8A	0x8B	0x8C	0x8D	0x8E	0x8F
0x90	0x91	0x92	0x93	0x94	0x95	0x96	0x97	0x98		0x9A	0x9B	0x9C	0x9D	0x9E	0x9F
0xA0	0xA1	0xA2	0xA3	0xA4	0xA5	0xA6	0xA7	0xA8	0xA9		0xAB	0xAC	0xAD	0xAE	0xAF
0xB0	0xB1	0xB2	0xB3	0xB4	0xB5	0xB6	0xB7	0xB8	0xB9	0xBA		0xBC	0xBD	0xBE	0xBF
0xC0	0xC1	0xC2	0xC3	0xC4	0xC5	0xC6	0xC7	0xC8	0xC9	0xCA	0xCB		0xCD	0xCE	0xCF
0xD0	0xD1	0xD2	0xD3	0xD4	0xD5	0xD6	0xD7	0xD8	0xD9	0xDA	0xDB	0xDC		0xDE	0xDF
0xE0	0xE1	0xE2	0xE3	0xE4	0xE5	0xE6	0xE7	0xE8	0xE9	0xEA	0xEB	0xEC	0xED		0xEF
0xF0	0xF1	0xF2	0xF3	0xF4	0xF5	0xF6	0xF7	0xF8	0xF9	0xFA	0xFB	0xFC	0xFD	0xFE	

Highlight color 0x00-0xFF.

The default highlight color is 0x0a (light Green on Black).

-csv

-efu

-txt

-m3u

-m3u8

Change display format.

Use > to redirect to a file or | to pipe to another application.

See *-export* options below to also write to a file.

-export-csv <out.csv>

-export-efu <out.efu>

-export-txt <out.txt>

-export-m3u <out.m3u>

-export-m3u8 <out.m3u8>

Export to a file. No output is displayed on screen.

-export-efu will only export indexed information. Combine *-export-efu* with *-size*, *-date-modified*, *-date-created* and/or *-attributes* to gather this information.

UTF-8 encoding is used for exporting as txt and csv.

-size-format <format>

Format can be one of the following values:

Value	Description
-------	-------------

0	Auto
1	Bytes (default)
2	KB
3	MB

-date-format <format>

Format can be one of the following values:

Value	Description
0	System format (default)
1	ISO-8601
2	Windows FILETIME in decimal
3	ISO-8601 as UTC

-pause

-more

Pause after each page of output.

-hide-empty-search-results

Don't show any results when the search is empty.

-empty-search-help

Show help when no search is specified.

-timeout <milliseconds>

Timeout after the specified number of milliseconds to wait for the Everything database to load before sending a query. The default timeout is 0 milliseconds.

-filename-color <color>

-name-color <color>

-path-color <color>

-extension-color <color>

-size-color <color>

-date-created-color <color>

-dc-color <color>

-date-modified-color <color>

-dm-color <color>

-date-accessed-color <color>

-da-color <color>

-attributes-color <color>

-file-list-filename-color <color>

-run-count-color <color>

-date-run-color <color>

-date-recently-changed-color <color>

-rc-color <color>

<color> can be one of the following color codes:

0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F
0x80	0x81	0x82	0x83	0x84	0x85	0x86	0x87	0x88	0x89	0x8A	0x8B	0x8C	0x8D	0x8E	0x8F
0x90	0x91	0x92	0x93	0x94	0x95	0x96	0x97	0x98	0x99	0x9A	0x9B	0x9C	0x9D	0x9E	0x9F
0xA0	0xA1	0xA2	0xA3	0xA4	0xA5	0xA6	0xA7	0xA8	0xA9	0xAA	0xAB	0xAC	0xAD	0xAE	0xAF
0xB0	0xB1	0xB2	0xB3	0xB4	0xB5	0xB6	0xB7	0xB8	0xB9	0xBA	0xBB	0xBC	0xBD	0xBE	0xBF
0xC0	0xC1	0xC2	0xC3	0xC4	0xC5	0xC6	0xC7	0xC8	0xC9	0xCA	0xCB	0xCC	0xCD	0xCE	0xCF
0xD0	0xD1	0xD2	0xD3	0xD4	0xD5	0xD6	0xD7	0xD8	0xD9	0xDA	0xDB	0xDC	0xDD	0xDE	0xDF
0xE0	0xE1	0xE2	0xE3	0xE4	0xE5	0xE6	0xE7	0xE8	0xE9	0xEA	0xEB	0xEC	0xED	0xEE	0xEF
0xF0	0xF1	0xF2	0xF3	0xF4	0xF5	0xF6	0xF7	0xF8	0xF9	0xFA	0xFB	0xFC	0xFD	0xFE	0xFF

Set the column color 0x00-0xFF.

-filename-width <width>
-name-width <width>
-path-width <width>
-extension-width <width>
-size-width <width>
-date-created-width <width>
-dc-width <width>
-date-modified-width <width>
-dm-width <width>
-date-accessed-width <width>
-da-width <width>
-attributes-width <width>
-file-list-filename-width <width>
-run-count-width <width>
-date-run-width <width>
-date-recently-changed-width <width>
-rc-width <width>

Set the column width 0-200.

-size-leading-zero
-run-count-leading-zero

Format the number with leading zeros, use with -no-digit-grouping.

-no-digit-grouping

Don't group numbers with commas.

-path <path>

Search for subfolders and files in path.

-parent-path <path>

Search for subfolders and files in the parent of path.

-parent <path>

Search for files with the specified parent path.

/o[sort]

DIR style sorts.

sort must be one the following sorts:

Sort	Description
N	Name ascending.
-N	Name descending.
S	Size ascending.
-S	Size descending.
E	Extension ascending.
-E	Extension descending.
D	Date modified ascending.
-D	Date modified descending.

/ad

Folders only.

/a-d

Files only.

/a[attributes]

DIR style attributes search.

attributes can be any one or more of the following attributes:

Attribute	Description
R	Read only.
H	Hidden.
S	System.
D	Directory.
A	Archive.
V	Device.
N	Normal.
T	Temporary.
P	Sparse file.
L	Reparse point.
C	Compressed.

O	Offline.
I	Not content indexed.
E	Encrypted.

To exclude an attribute prefix it with a -

For example, find files with no read-only attribute set: `es.exe /a-r`

The Directory attribute is determined by the result being a file or folder in Everything, not the file attribute. Search for `attributes:d` for the true directory attribute.

-set-run-count <filename> <count>

Set the run count for the specified filename. No search is performed.

-inc-run-count <filename>

Increment the run count for the specified filename by one. No search is performed.

-get-run-count <filename>

Display the run count for the specified filename. No search is performed.

-get-result-count

Display the total result count for the specified search.

No filenames are displayed. `-n` is ignored.

-save-settings

-clear-settings

Save or clear settings. No search is performed.

Settings are stored in `es.ini` in the same location as your `es.exe`.

Limitations

ES does not have access to bookmarks or filters.

Examples

Export all mp3 files to an Everything file list named `mp3.efu`

```
es.exe *.mp3 -export-efu mp3.efu
```

Show the top 10 largest files:

```
es.exe -sort size -n 10
```

Show the last 10 modified files:

```
es.exe -sort dm -n 10
```

Highlight the search terms `foo bar`

```
es.exe foo bar -highlight
```

Make ES show the size column, date modified column and set some pretty colors and make them the default settings:

```
es.exe -size -dm -sizecolor 0x0d -dmcolor 0x0b -save-settings
```

Return Codes

ES can return one of the following errorlevel codes:

Errorlevel	Description
0	No known error, search successful.
1	Failed to register window class.
2	Failed to create listening window.
3	Out of memory.
4	Expected an additional command line option with the specified switch.
5	Failed to create export output file.
6	Unknown switch.
7	Failed to send Everything IPC a query.
8	Everything IPC window not found. Please make sure the Everything search client is running.

Notes

Dashes (-) inside switches can be omitted, for example, instead of using -no-digit-grouping, use: -nodigitgrouping

Switches can also start with a /

Use double quotes to escape spaces and switches.

Switches can be disabled by prefixing them with no-, eg: -no-size.

Use ^ to escape \, &, |, >, < and ^.

ES source code is included with the download.

See also

- [IPC](#)
- [Multiple Instances](#)
- [Downloads](#)

Command Line Options

A list of all the command line options available in Everything.
Most of these options are available within Everything, in the [Everything options](#).

Syntax

```
Everything.exe [file-list-filename] [options]
```

[option]

Optional option.

<option>

Required option.

Installation

These command line options return immediately and do not show an Everything search window.

Some command line options require administrative privileges. Running Everything with a command line option that requires administrative privileges from a standard user account will automatically relaunch Everything with the specified command line option as administrator.

-app-data

-noapp-data

Store settings and data in %APPDATA%\Everything or in the same location as the executable.

When enabled, user settings in the Everything.ini in the same location as Everything.exe will be lost.

Requires administrative privileges.

-choose-language

Show the language selection page.

-choose-volumes

Do not automatically index volumes. Removes all NTFS volumes from the index.

-enable-run-as-admin

-disable-run-as-admin

Enable or disable run as administrator.

Requires administrative privileges.

-enable-update-notification

-disable-update-notification

Enable or disable update notification on startup.

-install <location>

Copies Everything.exe and uninstall.exe to the new location. Creates an uninstall entry in Programs and Features if uninstall.exe is present. Executes the copied Everything.exe with the install options specified in [-install-options](#).

Requires administrative privileges.

-install-client-service

-uninstall-client-service

Install the "Everything" client as a service.
Requires administrative privileges.

-install-config <filename>

Install the specified configuration file. The specified ini file is merged with any existing configuration file before any other install options are applied.

-install-desktop-shortcut

-uninstall-desktop-shortcut

Create or delete the desktop shortcut for the current user.
Requires administrative privileges.

-install-all-users-desktop-shortcut

-uninstall-all-users-desktop-shortcut

Create or delete the desktop shortcut for all users.
Requires administrative privileges.

-install-efu-association

-uninstall-efu-association

Create or remove the EFU file association with Everything.
Requires administrative privileges.

-install-folder-context-menu

-uninstall-folder-context-menu

Install or uninstall folder context menus.
Requires administrative privileges.

-install-options <command line options>

Command line options to pass to the newly installed Everything.exe.
Combine with [-install](#).
Requires administrative privileges.
install options are executed with administrative privileges.

-install-quick-launch-shortcut

-uninstall-quick-launch-shortcut

Create or delete the Search Everything quick launch shortcut.

-install-run-on-system-startup

-uninstall-run-on-system-startup

Add or remove Everything from the system startup.
Requires administrative privileges.

-install-service

-uninstall-service

Install or uninstall the "Everything" service. The service is started automatically. Combine with [-install-service-port](#) to set the service port. Combine with [-install-service-pipe-name](#) to set the service pipe name.

Requires administrative privileges.

-install-service-port <port>

Install the Everything service on the specified port.

Combine with [-install-service](#).

Removed in Everything 1.4, use [-install-service-pipe-name](#) instead.

Requires administrative privileges.

-install-service-pipe-name <name>

Install the Everything service with the specified pipe name.

Combine with [-install-service](#).

Requires administrative privileges.

-install-start-menu-shortcuts

-uninstall-start-menu-shortcuts

Create or delete the Everything shortcuts in the Start menu for the current user.

Requires administrative privileges.

-install-all-users-start-menu-shortcuts

-uninstall-all-users-start-menu-shortcuts

Create or delete the Everything shortcuts in the Start menu for all users.

Requires administrative privileges.

-install-url-protocol

-uninstall-url-protocol

Install or uninstall the URL Protocol.

Requires administrative privileges.

-language <langID>

Set the language to the specified language ID in decimal. 1033 = English (US). 0 = System default. See [Language IDs](#) for a list of langIDs.

-uninstall [path]

Uninstall Everything from the specified path. If no path is specified Everything is uninstalled from the executables location. This removes Everything from system startup, removes the service, folder context menu, start menu shortcuts, desktop shortcut, quick launch shortcut, URL protocol, efu file associations and deletes any files created by Everything.

Requires administrative privileges.

-uninstall-user

Uninstall Everything user files.

-create-usn-journal <volume> <max-size-bytes> <allocation-delta-bytes>

Create a USN Journal on the specified volume with the specified maximum size and allocation delta in bytes.

Requires administrative privileges.

-delete-usn-journal <volume>

Delete the USN Journal on the specified volume.

Requires administrative privileges.

-install-language <langID>

Include in [-install-options](#) to set the installation language.

Set the language to the specified language ID in decimal. 1033 = English (US). 0 = System default. See [Language IDs](#) for a list of langIDs.

Requires administrative privileges.

-save-install-options <user-install-option-flags>

Include in [-install-options](#) to save user install options to the registry.

user-install-option-flags in decimal, can be zero or more of the following:

Flag	Description
1	Update notifications
2	Install Quick Launch shortcut
4	Choose volumes

Requires administrative privileges.

File Lists

Command line options to manipulate Everything file lists EFU files.

[file-list-filename]

Open the specified file list.

-create-file-list <filename> <path>

Create a file list of a path. When this command completes Everything.exe will exit, no search window is shown.

-create-file-list-exclude-files <filters>

Set a semicolon delimited wildcard filter of files to exclude with -create-file-list.

Prefix a filter with regex: to use Regex.

-create-file-list-exclude-folders <filters>

Set a semicolon delimited wildcard filter of folders to exclude with -create-file-list.

Prefix a filter with regex: to use Regex.

-create-file-list-include-only-files <filters>

Set a semicolon delimited wildcard filter of files to include only with -create-file-list.

Prefix a filter with regex: to use Regex.

-edit <filename>

Open the specified file list with the file list editor.

-f <filename>

-filelist <filename>

Open a file list.

ETP

Command line options to connect to an ETP server.

-admin-server-share-links

-server-share-links

-ftp-links

-drive-links

Link type for ETP connections. Use with [-connect](#).

-connect <[username[:password]@]host[:port]>

Connect to an ETP server.

Searching

Command line options to manipulate the search.

-bookmark <name>

Open a bookmark.

-case

-nocase

Enable or disable case matching.

-diacritics

-nodiacritics

Enable or disable diacritics matching.

-filename <filename>

Search for a file or folder by filename.

-filter <name>

Select a search filter.

-l

-local

Load the local database.

-matchpath

-nomatchpath

Enable or disable full path matching.

-p <path>

-path <path>

Search for a path.

-parent <path>

Search for files and folders in path. Does not search subfolders.

-parentpath <path>

Search for the parent of a path.

-regex

-noregex

Enable or disable Regex.

-s <text>

-search <text>

Set the search.

-url <[es:]search>

Set the search from an ES: URL.

-wholeword

-nowholeword

-ww

-nowww

Enable or disable match whole word.

-home

Open the home search.

-name-part <filename>

Search for the name part of a filename.

-search-file-list <filename>

Search the specified text file for a list of file names.

Each filename must appear on a single line.

Include a backslash to match the full path and filename.

Results

Command line options to manipulate the results.

-sort <name>

Set the sort to the specified name.

-sort-ascending

Sort ascending.

-sort-descending

Sort descending.

-details

View results in the detail view.

-thumbnail-size <size>

Specify the size of thumbnails in pixels.

-thumbnails

Show results in thumbnail view.

-focus-bottom-result

Focus the bottom result.

-focus-last-run-result

Focus the last run result.

-focus-most-run-result

Focus the most run result.

-focus-results

Focus the result list.

-focus-top-result

Focus the top result.

-select <filename>

Focus and select the specified result.

General

General command line options.

-?

-h

-help

Show this help.

-admin

Run Everything as an administrator.

-client-svc

Everything client service entry point.

-config <filename>

The filename of the ini file.

-console

-debug

Show the debugging console.

-debug-log

Enables debug mode and [logs debugging information](#) to disk.

-exit

-quit

Exit an existing "Everything" instance.

-instance <name>

The name of the Everything [instance](#).

-is-run-as

Everything was executed with "runas" and should not attempt to runas again.

-start-client-service

-stop-client-service

Start or stop the Everything client service.

-start-service

-stop-service

Start or stop the Everything service.

-startup

Run Everything in the background.

-svc

Service entry point. Optionally combine with *-svc-port*.

-svc-port <port>

Run the Everything service on the specified port.

-svc-pipe-name <name>

Host the pipe server with the specified name.

-svc-security-descriptor <sd>

Host the Everything Service pipe server with the specified security descriptor.

Requires Everything 1.4.1.994 or later.

-service-port <port>

Connect to the Everything service on the specified port.

-service-pipe-name <name>

Connect to the Everything service using the specified pipe name.

-verbose

-noverbose

Display all or basic debug messages. combine with [-debug](#).

-first-instance

Only run Everything if this is the first instance of Everything.

If Everything is already running this call will be ignored.

-no-first-instance

Only run Everything if Everything is already running.

If Everything is not running this call will be ignored.

Database

Command line options to manipulate the Everything database.

-db <filename>

The filename of the database. Combine with -read-only.

-load-delay <milliseconds>

The delay in milliseconds before loading the database.

The database is loaded if required before this delay.

-nodb

Do not save to or load from the "Everything" database file.

-read-only

Do not update the database.

Prevents new database creation.

-reindex

Force database rebuild.

-update

Save the database to disk.

-monitor-pause

Pause NTFS, ReFS and folder index monitors.

-monitor-resume

Resume NTFS, ReFS and folder index monitors.

Window

Command line options to manipulate the search window and results.

-fullscreen

-nofullscreen

Show the search window fullscreen or in a window.

-maximized

-nomaximized

Maximize or restore the search window.

-minimized

-nominimized

Minimize or restore the search window.

-newwindow

-nonewwindow

Create a new search window or show an existing window.

-ontop

-noontop

Enable or disable always ontop.

-close

Close the current search window.

-toggle-window

Shows or hides the current search window.

Multi File Renaming

Shows the multi file renamer dialog and does not start Everything.

-copyto [filename1] [filename2] [filename3] [...]

Show the Everything multi file renamer for a copy to operation.

-moveto [filename1] [filename2] [filename3] [...]

Show the Everything multi file renamer for a move to operation.

-rename [filename1] [filename2] [filename3] [...]

Show the Everything multi file renamer for a rename operation.

Notes

Dashes (-) inside switches can be omitted, for example, instead of using `-no-new-window`, use: `-nonewwindow`

Use double quotes to escape spaces and switches.

Use `^` to escape `\`, `&`, `|`, `>`, `<` and `^`.

Use `""` to escape a literal `"`

Examples

Search for ABC or 123:

```
Everything.exe -s "ABC|123"
```

Connect to an ETP server:

```
Everything.exe -connect "ComputerName" -drive-links
```

Create a filelist mp3 and flac files in D:\Music

```
Everything.exe -create-file-list "music.efu" "D:\Music" -create-file-list-include-on
```

See also

- [Multiple Instances](#)

Customizing

Everything can be customized through the Everything.ini, registry and Everything options. This guide will help you customize Everything.

Window Caption

To customize the Everything search window caption, HTTP server title and/or taskbar notification title:

- Exit Everything
- Open your [Everything.ini](#) file.
- Change the following lines:

```
window_title_format=$s?{$s - }$t$i?{ ($i)}
http_title_format=$s?{$s - }$t
taskbar_notification_title_format=$t$i?{ ($i)}
```

- Set a new format:

Variable	Description
\$s	Current search
\$t	Everything (localized)
\$v	Version
\$i	Instance name

- Use the following syntax to optionally display text:

Syntax	Description
\$x?{text}	Display text if \$x is not empty.

- Save changes and close the Everything.ini.
- Restart Everything.

Example formats:

Format	Search	Title
\$s?{\$s - }My Search	apples	apples - My Search
\$s?{\$s - }My Search		My Search

Note: Changing the window_title_format will also affect the title on the taskbar button.

Icon

To change the Everything icon:

- Copy your custom icon file to:

```
%APPDATA%\Everything\Everything.ico
```

If [Store settings and data in %APPDATA%\Everything](#) is disabled, copy your custom icon file to Everything.ico in the same location as your Everything.exe.

- Exit **Everything** (Right click the Everything system tray icon and click Exit)
- Restart **Everything**.

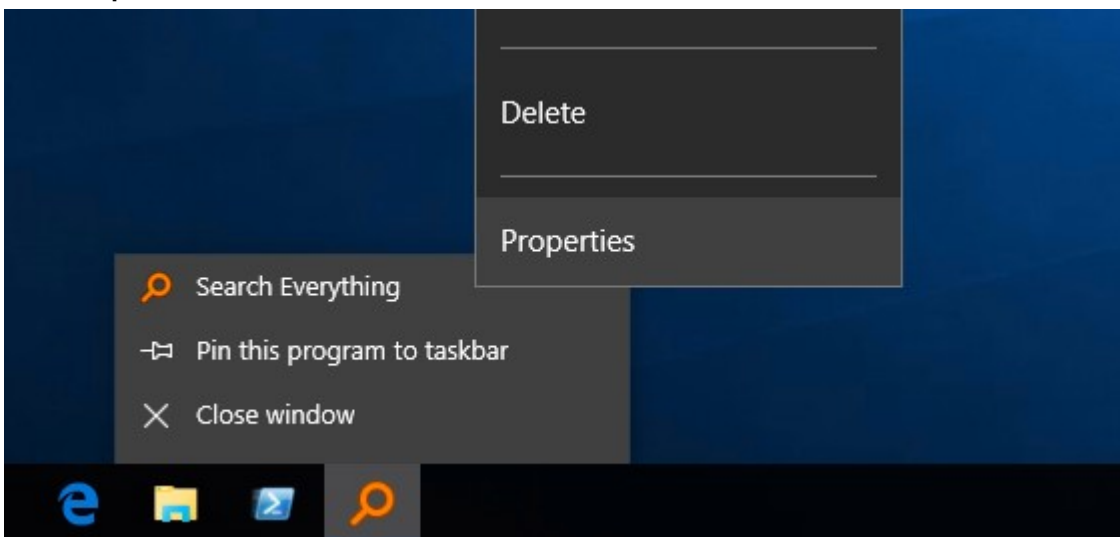
Download the colorful Everything Icon Pack:

[Everything-Icons.zip](#)

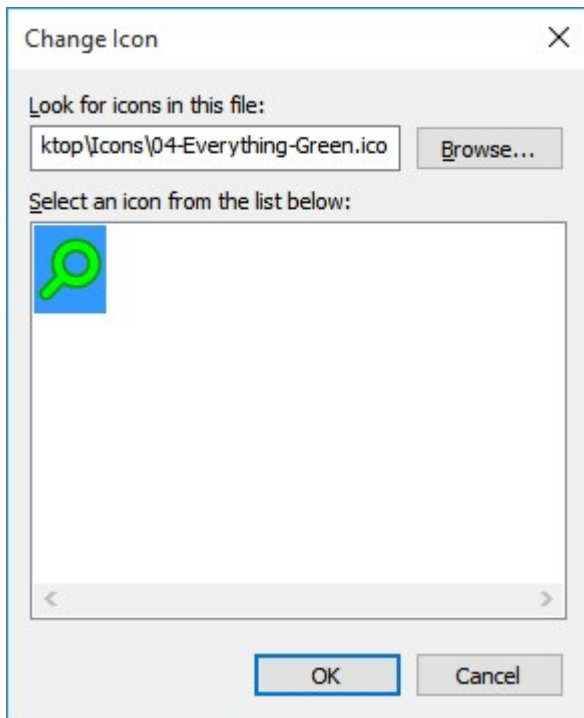
Taskbar Group Icon

To change the Everything Taskbar group icon:

- Right click the Everything taskbar button.
- Hold down the Shift key and right click Search Everything.
- Click **Properties**.

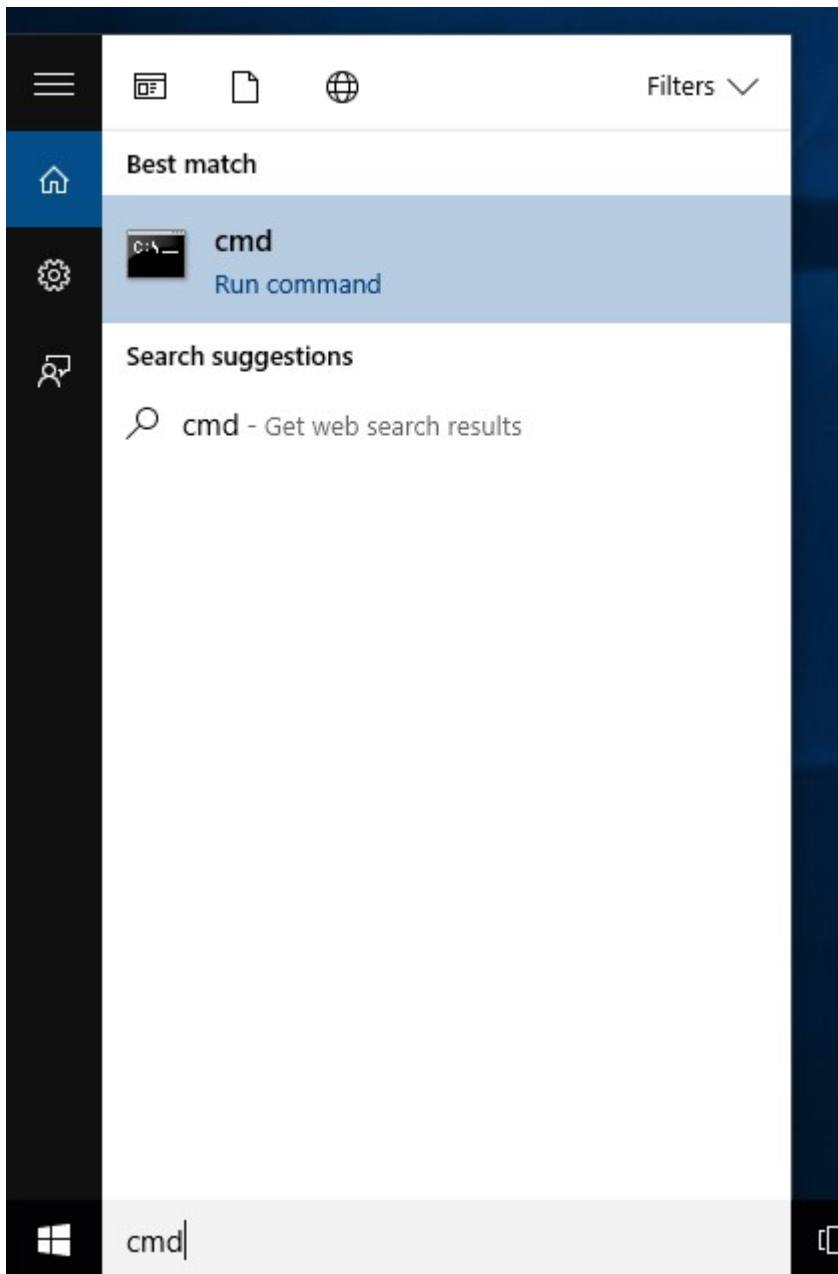


- Click **Change Icon**.
- Click **Browse...**, select a new icon and click **Open**.
- Click **OK**.
- Click **OK**.



- Rebuild the icon cache:
- From the Start menu, type in the following search and press enter:

cmd



- In the command prompt, navigate to local app data folder by typing in the following and pressing enter:

```
cd %localappdata%
```

- Remove the hidden attribute from the icon cache database by typing the following and pressing enter:

```
attrib iconcache.db -h
```

- Delete the icon cache database by typing the following and pressing enter:

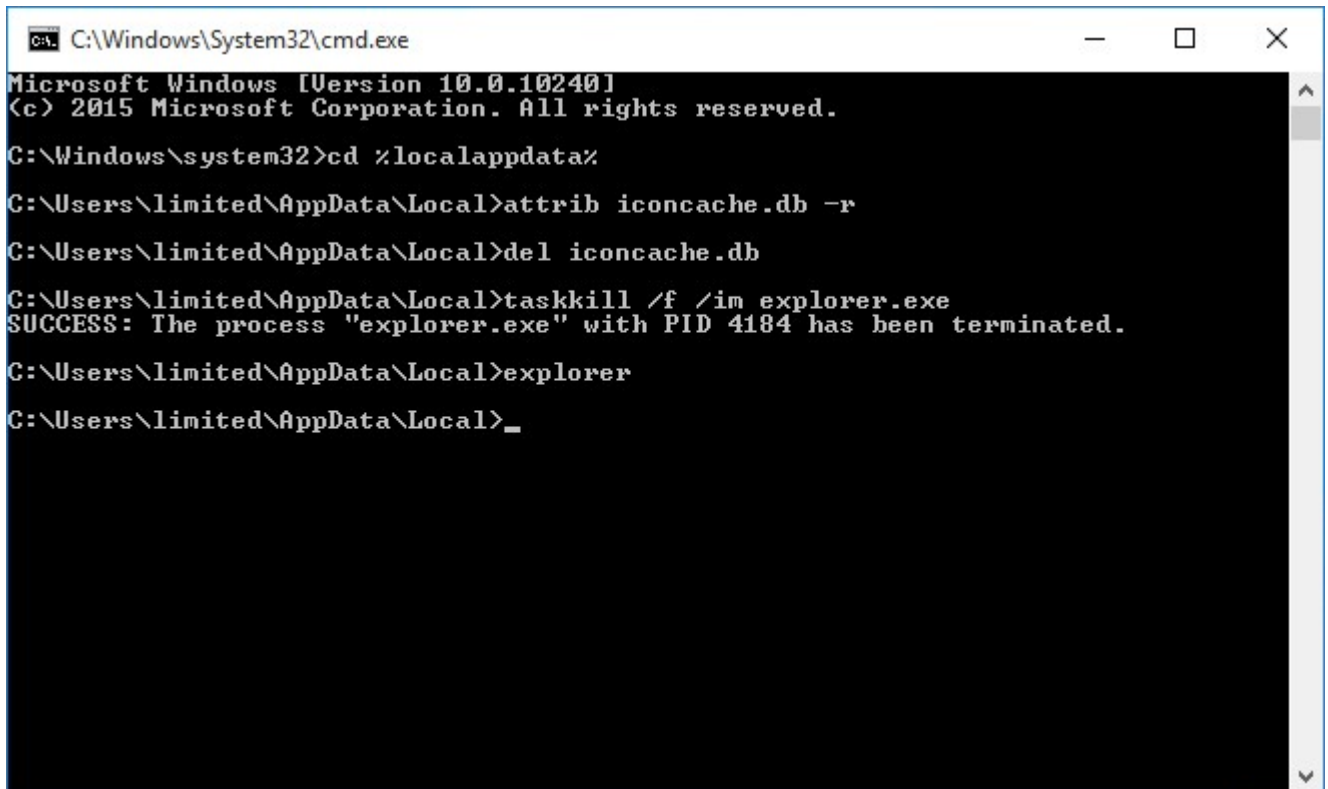
```
del iconcache.db
```

- Terminate Windows Explorer by typing the following and pressing enter:

```
taskkill /f /im explorer.exe
```

- Restart Windows Explorer by typing the following and pressing enter:

explorer



```
ca. C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd %localappdata%
C:\Users\limited\AppData\Local>attrib iconcache.db -r
C:\Users\limited\AppData\Local>del iconcache.db
C:\Users\limited\AppData\Local>taskkill /f /im explorer.exe
SUCCESS: The process "explorer.exe" with PID 4184 has been terminated.
C:\Users\limited\AppData\Local>explorer
C:\Users\limited\AppData\Local>_
```

External File Manager

To open paths in an external file manager:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Context Menu** tab.
- Select **Open Path**.
- Change the **command** to:

```
$exec("C:\Program Files $(x86$)\ExternalFileManager\ExternalFileManager.exe" "%1")
```

- Where C:\ExternalFileManager\ExternalFileManager.exe is the location of the external file manager.
- Use \$\$, \$%, \$(and \$) to escape \$, %, (and).
- Check your external file manager help for any required command line parameters.
- Click OK.

Date Format

To use a custom date format in Everything:

- Exit Everything.
- Open your [Everything.ini](#).
- Change the following line:

```
date_format=
```

Date format can be zero or more of the following codes:

code	Description
d	Day of month as digits with no leading zero for single-digit days.
dd	Day of month as digits with leading zero for single-digit days.
ddd	Day of week as a three-letter abbreviation. The function uses the <code>LOCALE_SABBREVDAYNAME</code> value associated with the specified locale.
dddd	Day of week as its full name. The function uses the <code>LOCALE_SDAYNAME</code> value associated with the specified locale.
M	Month as digits with no leading zero for single-digit months.
MM	Month as digits with leading zero for single-digit months.
MMM	Month as a three-letter abbreviation. The function uses the <code>LOCALE_SABBREVMONTHNAME</code> value associated with the specified locale.
MMMM	Month as its full name. The function uses the <code>LOCALE_SMONTHNAME</code> value associated with the specified locale.
y	Year as last two digits, but with no leading zero for years less than 10.
yy	Year as last two digits, but with leading zero for years less than 10.
yyyy	Year represented by full four digits.
gg	Period/era string. The function uses the <code>CAL_SERASTRING</code> value associated with the specified locale. This element is ignored if the date to be formatted does not have an associated era or period string.

- Save changes to the `Everything.ini`
- Restart Everything.

Time Format

To use a custom time format in Everything:

- Exit Everything.
- Open your [Everything.ini](#).
- Change the following line:

```
time_format=
```

Time format can be zero or more of the following codes:

Code	Description
h	Hours with no leading zero for single-digit hours; 12-hour clock.
hh	Hours with leading zero for single-digit hours; 12-hour clock.
H	Hours with no leading zero for single-digit hours; 24-hour clock.
HH	Hours with leading zero for single-digit hours; 24-hour clock.
m	Minutes with no leading zero for single-digit minutes.
mm	Minutes with leading zero for single-digit minutes.

s	Seconds with no leading zero for single-digit seconds.
ss	Seconds with leading zero for single-digit seconds.
t	One character time-marker string, such as A or P.
tt	Multicharacter time-marker string, such as AM or PM.
SSS	Milliseconds with leading zeros, must be the last part of the format.

- Save changes to the Everything.ini
- Restart Everything.

Translucent Selection Rectangle

To change the translucency see [translucent_selection_rectangle_alpha](#)

To change the background color see [translucent_selection_rectangle_background_color](#)

To change the border color see [translucent_selection_rectangle_border_color](#)

Change the default search and search settings

See [home](#)

Fonts and colors

To change the result list font see [Fonts and colors](#)

To change the search edit font see [search_edit_font](#) and [search_edit_font_size](#)

To change the status bar font see [status_bar_font](#) and [status_bar_font_size](#)

To change the result list header font see [header_font](#) and [header_font_size](#)

Status bar format

To change the status bar format see [statusbar_selected_item_format](#)

HTTP Server

See [Change the default HTTP files](#).

ETP/FTP Welcome Message

See [etp_server_welcome_message](#) in the Everything.ini.

See Also

- [Ini](#)
- [Translating](#)

- [Indexes](#)
- [Searching](#)
- [Command line options](#)

DB

This is very basic description and is likely to change...

Since the database is just a cache of the NTFS mft with some volume information, it can be re-built from scratch easily.

BZIP compression can be applied.

I usually see a total 11:1 compression ratio.

After BZIP decompression it should look something like:

Offset	Size	Description
0x00000000	DWORD	0x42445A45 or EZDB or (0x455A4442 or BDZE for byte swapped)
0x00000004	DWORD	0x01060006 or version 0xXXYYZZZZ x.y.z (major.minor.revision)
0x00000008	DWORD	flags: 0x00000002 exclude hidden files and folders. 0x00000004 exclude system files and folders.
0x0000000C	DWORD	folder count
0x00000010	DWORD	file count
0x00000014	DWORD	folder decode size
0x00000018	DWORD	file decode size

Followed immediately by the monitoring status for each volume.

There are 26 of the following structures, one for each drive:

Offset	Size	Description
0x00000000	BYTE	included: 0x00 there is no data for this volume so start reading the next volume. 0x01 continue reading the information for this volume.
0x00000001	DWORD	serial number
0x00000005	QWORD	USN Journal ID
0x0000000D	QWORD	Next USN

Followed immediately by the exclude list count:

0x00000000	DWORD	number of exclude items.
------------	-------	--------------------------

Followed immediately by the exclude list:

Use the exclude list count from above to determine how many exclude items there are..

Offset	Size	Description
0x00000000	BYTE	exclude type

0x00000001	DWORD	length of the exclude string
0x00000005	length of the exclude string	exclude string

Followed immediately by the sorted folders:
We know the number of folders from above.

Offset	Size	Description
0x00000000	BYTE	drive this folder belongs too.
0x00000001	QWORD	FRN
0x00000005	DWORD	Parent folder offset, or 0xFFFFFFFF if root.
0x00000009	DWORD	FRN offset
0x0000000D	BYTE	code length, if 0 it is the same name, skip to the next folder after adding the same name.
0x0000000E	BYTE	code offset
0x0000000F	code length	code

Followed immediately by the sorted files:
We know the number of files from above.

Offset	Size	Description
0x00000000	DWORD	Parent folder offset
0x00000004	BYTE	code length, if 0 it is the same name, skip to the next file after adding the same name.
0x00000005	BYTE	code offset
0x00000006	code length	code

ETP

The Everything ETP/FTP server allows you to search and access your files from an Everything client or FTP client.

Start an ETP/FTP server

To start an ETP/FTP server:

- On the computer whose files you wish to share, In **"Everything"**, from the **Tools** menu, click **Options**.
- Click the **ETP/FTP** tab.
- Check **Enable ETP/FTP server**.
- Click **OK**.

Please make sure you share each drive on your server as its drive letter.
For example, share C: drive as: **C**

To use custom shares, please see [ETP path rewriting](#).

Connect to an ETP server

To connect to an ETP server:

- In "**Everything**", from the **Tools** menu, click **connect to ETP server...**
- Type in the **server name** or **IP** for the host.
- Click **OK**.

ETP link types

There are four link types that change the way the Everything client accesses files on the ETP server.

When connecting to an ETP server you can specify the link type.

Link type	Description
C:	No change, the paths are the same as they are on the server. This is useful if you index a substed drive on the server and are using mapped network drives on the client.
\\Server\C	Use Windows shares (this is the default link type). By default these shares do not exist, you will need to create them on the server if you wish to use this method.
\\Server\C\$	Use the default admin drive shares.
ftp://host/C:	Use ftp links.

Username and password

To change the ETP/FTP server username and password:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **ETP/FTP Server** tab.
- Type in a new **username** and **password**.
- Click **OK**.

Disable file downloading

To disable ETP/FTP file downloading:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **ETP/FTP Server** tab.
- Uncheck **Allow file download**.
- Click **OK**

Different indexes

To index different NTFS volumes for the ETP server, see [Multiple Instances](#).

Create a Windows share

Everything works best if you share each drive as a single letter, for example, share your C: drive as C, D: drive as D and so on.

To create a windows share for a single folder, please follow the guide below:

For example, we want to host an ETP/FTP server for one folder:

```
C:\share
```

Substitute a drive letter for the local folder c:\share:

- From a command prompt, run:

```
subst H: c:\share
```

Setup an index to include only the H: drive.

- In **"Everything"**, from the **Tools** menu, click **Options**.
- Click the **NTFS** tab.
- Uncheck **Include in database** for all volumes, except the H: drive.
- Click **OK**.

Add a share name to c:\share

- In **Explorer**, right click the **c:\share** and click **properties**.
- Click the **Sharing** tab.
- Click **Advanced Sharing....**
- Click **Add**.
- Type in **H** for the **share name**.
- Click **OK**.
- Click **OK**.
- Click **Close**.

Connect to the ETP/FTP server with \\Server\C links.

Security

Every file and folder indexed by Everything can be searched and downloaded via the ETP server.

To disable file downloading:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **ETP Server** tab.
- Uncheck **allow file download**.

See [Disable ETP/FTP Server](#) to remove the ETP server options and prevent the ETP server from starting.

Disable ETP/FTP Server

To disable the ETP/FTP server:

- Exit Everything (right click the Everything system tray icon and click Exit)
- Open your Everything.ini in the same location as your Everything.exe
- Change the following line:

```
allow_etp_server=1
```

to:

```
allow_etp_server=0
```

- Save changes and restart Everything.

ETP Client path rewriting

ETP clients can rewrite the ETP server paths so they can be accessed from different shares.

For example, rewrite the path D:\music to \\server\music and "D:\Install Files" to "\\server\Install Files"

To rewrite the paths on the ETP client.

- On the **Everything** ETP client PC:
- Completely exit **Everything** (Right click the **Everything** system tray icon and click Exit)
- Open your %APPDATA%\Everything\Everything.ini
- Change the following lines:
etp_client_rewrite_patterns=
etp_client_rewrite_substitutions=
to:
etp_client_rewrite_patterns=D:\music;"D:\\Install Files"
etp_client_rewrite_substitutions=\\server\music;"\\\\server\\Install Files"
- Save changes and restart Everything.

The pattern must match the path on the server. It is not effected by the link type.

Running an ETP server as a service

To run an Everything ETP server as a client service (not to be confused with the Everything service):

- Copy your Everything.exe to an empty folder.
- Run Everything.exe as administrator
- Please make sure [Store settings and data in %APPDATA%\Everything](#) is disabled.
- Please make sure the [Everything service](#) is not installed.
- Setup your [indexes](#).
- Setup the [ETP server settings](#).
- Completely exit Everything (right click the Everything tray icon and click Exit).
- From a command prompt, navigate to your Everything.exe
- Run the following command to install the client service:

```
Everything.exe -install-client-service
```

This will install and start the Everything ETP server as a service.

To uninstall the Everything client service:

- From a command prompt, navigate to your Everything.exe
- Run the following command to install the client service:

```
Everything.exe -uninstall-client-service
```


Automatically connect to an ETP server

To automatically connect to an ETP server when starting Everything:

- On the **"Everything"** client, from the **Tools** menu, click **Options**.
- Click the **Home** tab.
- Change **Index** to: **ETP server**.
- Set the **ETP server** to: username:password@host:port
- Set the desired **Link type**.
- Click **OK**.

Trouble Shooting

Unable to start ETP server: bind failed 10048

Please make sure no FTP servers are already running on port 21 or use a different ETP server port.

To change the ETP server port:

- In **"Everything"**, from the **Tools** menu, click **Options**.
- Click the **ETP/FTP Server** tab.
- Change the **port** to: **2121**
- Click **OK**.

Please match the same port number when connecting to the ETP server.

See also

- [Multiple Instances](#).
- [ETP/FTP Server command line options](#).
- [ETP/FTP Server Everything.ini options](#).
- [ETP/FTP Server options](#).
- <https://en.wikipedia.org/wiki/FTP>

Everything Service

The Everything service helps Everything index NTFS volumes and monitor USN Journals.

Everything can run as a standard user when the Everything service is installed.

On XP or earlier the Everything service is not required.

On Vista or later with a standard user account you will need to install the Everything service to use NTFS indexing.

On Vista or later with the true administrator account the Everything service is not required.

The Everything service is installed by default on Windows Vista or later when the installer is run with a standard user account.

Installing

- Check the Install Everything Service option when running the Everything installer.

-or-

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **General** Tab.
- Check **Everything Service**.
- Uncheck **Run as administrator**.
- Click **OK**.

How is the Everything service started?

Windows automatically starts the Everything service on system start-up.

Stopping or starting the service

The Everything service is always running and generally does not need to be stopped.

However, you can start or stop the Everything service from Start -> Control Panel -> Administrative Tools -> Services. Right click the Everything service and select Start or Stop.

How can I manage the Everything service from the command line?

```
Everything.exe -install-service [-install-service-pipe-name <name>] [-install-service-  
Everything.exe -uninstall-service  
Everything.exe -start-service  
Everything.exe -stop-service
```

How can I manage the Everything service from Everything?

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **General** Tab.
- Check or uncheck the **Everything service** to install or uninstall the Everything service.
- Click **OK**.

How can I run the Everything service from the portable version of Everything?

To launch the Everything service as a normal administrator application, run the following:

```
Everything.exe -svc
```

To specify the pipe name, use the -svc-pipe-name <name> command line option, for example:

```
Everything.exe -svc -svc-pipe-name "my Everything pipe name"
```

Running the Everything service this way does not modify the system.

The Everything service has no interface.

Use Task Manager to terminate the service application when it is no longer needed.

You can close the portable service application easily by making a batch file that runs Everything.exe -svc

Simply close the console window to terminate the service application.

How many resources does the service use?

The Everything service will use about 1MB of RAM and will have none to very little CPU usage.

Security

Any local user can use the Everything Service to create a list of all NTFS filenames.

Note: The Everything Service can not be used to access files, only filenames.

File Lists

Everything File List (EFU) are recommended for off-line files or files that do not change, for example a CDROM, DVDROM or read only NAS.

An EFU file is a Comma-separated Values (CSV) file containing a list of filenames, sizes, dates and attributes.

File lists can be opened with Everything or included in your Everything index.

File lists can be searched and sorted by size, date or attributes instantly.

How do I make an EFU file list?

- In **Everything**, from the **Tools** menu, click **File List Editor...**
- In the **File List Editor**, **drag and drop** files into the list to add to your file list.
- You can also add files and folders from the Edit menu.
- When you are finished, from the **File** menu, click **Save**.

How do I make an EFU file list from Everything?

To save the current search as an EFU file list

- In **"Everything"**, from the **File** menu, click **Export...**
- Change **Save as Type** to **EFU Everything File List (*.efu)**.
- Type in a filename and click **OK**.

How do I make an EFU File list from the command line?

To create a file list from the command line:

```
Everything.exe -create-file-list <output-filename> <path-to-scan>
```

Combine with [-create-file-list-exclude-files](#) <filters> to exclude files.

Combine with [-create-file-list-exclude-folders](#) <filters> to exclude folders.

Combine with [-create-file-list-include-only-files](#) <filters> to include only the specified files.

Examples

To create a file list excluding tmp files:

```
Everything.exe -create-file-list "myfilelist.efu" "c:\" -create-file-list-exclude-fi
```

To create a file list excluding lnk and thumbs.db files:

```
Everything.exe -create-file-list "myfilelist.efu" "c:\" -create-file-list-exclude-fi
```

To exclude folders with the name 1033:

```
Everything.exe -create-file-list "myfilelist.efu" "c:\" -create-file-list-exclude-fc
```

To exclude a specific folder

```
Everything.exe -create-file-list "myfilelist.efu" "c:\" -create-file-list-exclude-fc
```

To exclude specific folders

```
Everything.exe -create-file-list "myfilelist.efu" "c:\" -create-file-list-exclude-fc
```

To include specified files only:

```
Everything.exe -create-file-list "myfilelist.efu" "c:\" -create-file-list-include-on
```

regex can be used in the exclude lists, for example, exclude files starting with A-M

```
Everything.exe -create-file-list "myfilelist.efu" "c:\" -create-file-list-exclude-fi
```

Path matching can be done by including a \ in your filter (\\ for regex).

```
Everything.exe -create-file-list "myfilelist.efu" "c:\" -create-file-list-exclude-fi
```

Does Everything access the files in the file list?

Everything will only access the files if you try to open a file or right click a file.

File size, dates and attributes are read from the file list.

How do I change the volume for all the files in a file list?

- Open the file list in Notepad.
- Press **Ctrl + H** to replace.
- Change **Find what** to the old volume (eg: D:)
- Change **Replace with** to the new volume (eg: E:)
- Click **Replace All**.
- **Save** changes to the file list.

Can I use relative paths in a file list?

Yes, filenames are relative to the file list location.

Optionally use . and .. for locations relative to the file list location.

You can use \ at the start of a filename to specify a file relative to the root of the file list.

Specify a filename with a full path to override relative filenames.

Update a file list with the Task Scheduler

To rebuild a file list every hour with the task scheduler:

- Open the **Task Scheduler** from Start -> Programs -> Accessories -> System Tools -> Task Scheduler.
- Click **Create Task....**
- Set the **Name** to **Everything File List**

- Click the **Triggers** tab.
- Click **New....**
- Select **One time**.
- Check **Repeat task every**.
- Change the **Repeat task every time** to **1 hour**.
- Change **for a duration of** to **Indefinitely**.
- Click **OK**.

- Click the **Actions** tab.
- Click **New....**
- Select Start a program for Action.
- Click Browse to select Everything.exe
- Set **Add arguments to**:

```
-create-file-list c:\filelist.efu c:\
```

- Click **OK**.
- Click **OK**.

Include a file list in the Everything index

- In **"Everything"**, from the **Tools** menu, click **Options**.
- Click the **File Lists** tab.
- Click **Add...** to add a file list.
- Select a file list.
- Click **Open**.
- Click **OK**.

Modifying file lists

"Everything" will automatically reload modified file lists that are included in the Everything index.

To disable this feature:

- In **"Everything"**, from the **Tools** menu, click **Options**.
- Click the **File Lists** tab.
- Select the file list to stop monitoring.
- Uncheck **Monitor changes**.
- Click **OK**.

What is the format of an EFU file list?

EFU files are comma-separated values ([CSV](#)) files.

A header is required with at least the Filename column specified.

File size is specified in bytes.

Dates are FILETIMES (100-nanosecond intervals since January 1, 1601.) in decimal or [ISO 8601](#) dates.

Attributes can be zero or more of the [Windows File Attributes](#).

EFU files are encoded with UTF-8.

Opening a file list with Excel

To open a file list with excel:

- In **Excel**, from the **Data** menu, from the **Import External Data** submenu, click **Import data....**
- Select your exported csv and click **Open**.
- Change **File origin** to **65001 : Unicode (UTF-8)**.
- Click **Next**.
- Check **Comma**.
- Click **Next**.
- Click **Finish**.
- Click **OK** to put the data in the existing worksheet at **\$A\$1**.

See also

- [Folder indexing](#)

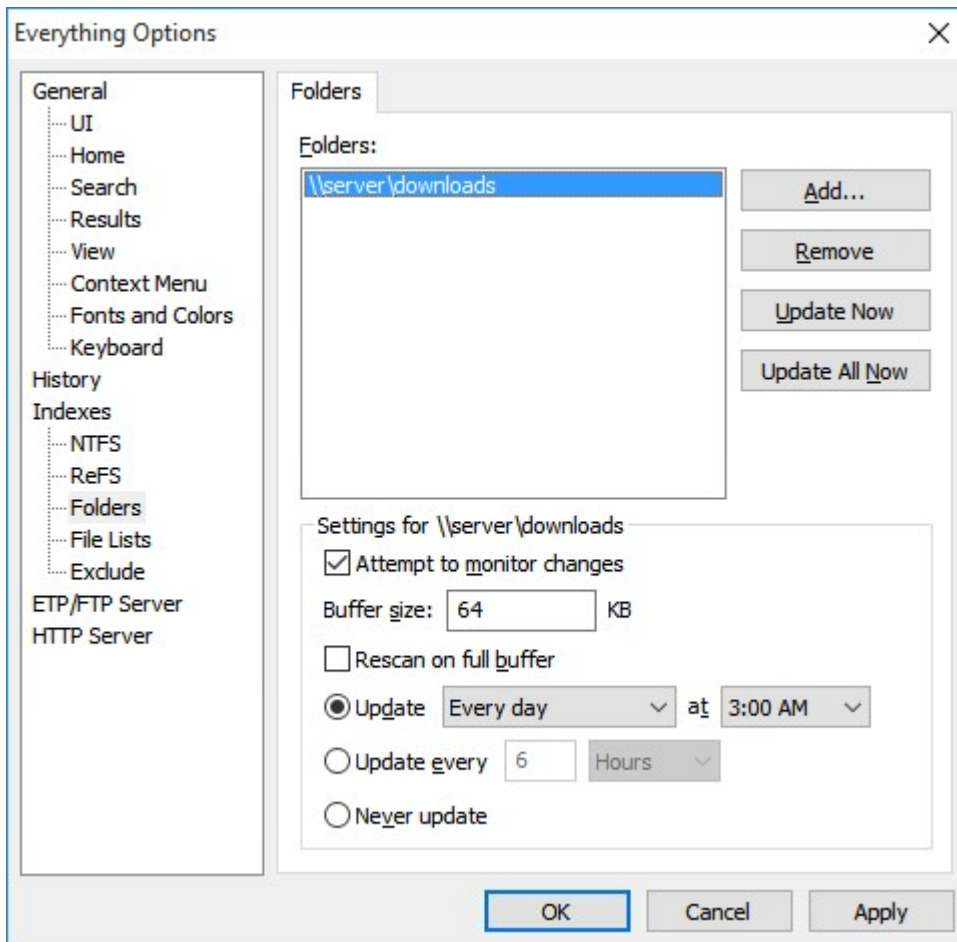
Folder Indexing

Folder indexing allows Everything to include any file system folder in the Everything index.

Indexed folders can go offline and remain in the Everything index.

The following folders can be added to your Everything index:

- Network shares or mapped network drives.
- FAT32 and other volumes.
- Any physical file-system folder.



Do I need administrative privileges or the Everything service to index folders?

No, folder indexing does not require administrative privileges or the Everything service.

Why is indexing so slow?

Folder indexing uses the same approach as the Windows search.

This can be a lot slower than NTFS indexing.

Everything can take a couple minutes to scan a folder and all its subfolders and files.

How do I add a folder to the Everything index?

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Folders** tab.
- Click **Add....**
- Select a folder to add to the Everything index.
- Click **OK**.
- Click **OK**.

How do I add a network share to the Everything index?

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Folders** tab.

- Click **Add....**
- Select the network share to add to the Everything index, for example:

```
\\server\share
```

- Click **OK**.
- Click **OK**.

How do I add a mapped network drive to the Everything index?

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Folders** tab.
- Click **Add....**
- Select the mapped network drive.
- Click **OK**.
- Click **OK**.

If network drives are not listed in Everything, please try running Everything as a standard user:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **General** tab on the left.
- Check **Everything service**.
- Uncheck **Run as administrator**.
- Click **OK**.
- Restart Everything (right click the Everything tray icon and click Exit).

How do I stop Everything from scanning folders on startup

To prevent Everything from rescanning folders on startup:

- In **Everything**, type in the following search and press **ENTER**:
[/folder_update_rescan_asap=0](#)

Folder rescanning will occur at the next scheduled time.

Will Everything detect all file name changes made to this folder?

Almost all changes are detected.

Changes can be missed if too many occur in a short time.

To rebuild the folder index when too many changes occur in a short time:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Folders** tab.
- Select a folder.
- Check **Rescan on full buffer**.

Everything can schedule an update time or update interval to rescan the entire folder for changes that might have been missed.

To customize the update time or update interval:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Folders** tab.
- Select a folder.

- Choose one of the options below:
 - Update each day, or weekly on a specific day, at a specific time.
 - Update periodically in the specified hours or minutes.
 - Never update.

How do I manually update a folder index?

To manually update all folder indexes:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Folders** tab.
- Click **Rescan All Now**.
- Click **OK**.

To manually update a folder index:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Folders** tab.
- Select the folder to update and click **Rescan Now**.
- Click **OK**.

What happens if the indexed folder is offline or not available?

The folder index will remain unchanged.

Everything will continue to re-scan the folder at the specified update time or update interval and only update the folder when it is online.

Displaying icons and file information of offline folders can take several seconds to time-out.

You can press F5 to refresh this cache when the folder is back online.

HTTP

The Everything HTTP server is a web server that allows you to search and access your files from a web browser.

Start a HTTP server

To start a HTTP server:

- In **Everything**, From the **Tools** menu, click **Options**.
- Click the **HTTP Server** tab.
- Check **Enable HTTP server**.
- Click **OK**.

View a HTTP server

Start the HTTP server and open `http://ComputerName` in your web browser. Where ComputerName is the name of the computer running the HTTP server.

Set a username and password

Changing the username and password will take effect immediately.

To change the HTTP server username and password

- In **Everything**, from the **Tools** menu, click **Options**
- Click the **HTTP server** tab.
- Type in a new **username** and **password**.
- Click **OK**

Disable file downloading

You can disable file downloading and allow clients to list results only.

To disable HTTP file downloading:

- In **Everything**, from the **Tools** menu, click **Options**
- Click the **HTTP server** tab.
- Uncheck **Allow file download**.
- Click **OK**

URL query string

Syntax:

```
http://localhost/?s=&o=0&c=32&j=0&i=0&w=0&p=0&r=0&m=0&path_column=0&size_column=0&da
```

key=value pairs can be omitted if not required.

Keys:

s

q

search

search text

o

offset

display results from the nth result

c

count

return no more than value results

j

json

return results as a JSON object if value is nonzero

i

case

match case if value is nonzero

w

wholeword

search whole words if value is nonzero

p

path

search whole paths if value is nonzero

r

regex

perform a regex search if value is nonzero

m

diacritics

match diacritics if value is nonzero

path_column

list the result's path in the json object if value is nonzero

size_column

list the result's size in the json object if value is nonzero

date_modified_column

list the result's modified date in the json object if value is nonzero

sort

where value can be one of the following:

Sort name	Description
name	Sort by Name.
path	Sort by Path.
date_modified	Sort by Date Modified.
size	Sort by Size.

ascending

sort by ascending order if value is nonzero

Default html query strings values:

Key	Value
search	
offset	0
count	32
json	0
case	0
wholeword	0
path	0
regex	0
diacritics	0
sort	name

ascending 1

Default JSON object query strings values:

Key	Value
search	
offset	0
count	4294967295
json	1
case	0
wholeword	0
path	0
regex	0
diacritics	0
path_column	0
size_column	0
date_modified_column	0
date_created_column	0
attributes_column	0
sort	name
ascending	1

For example, search for ABC AND 123, from the starting offset of 0, displaying only the first 100 results, sorted by size descending:

```
http://localhost/?search=ABC+123&offset=0&count=100&sort=size&ascending=0
```

Change the default HTTP files

You can customize the layout of the server, the icon, folder image, file image, everything logo, sort up image, sort down image and up one folder image.

- Create the folder

```
HTTP Server
```

in:

```
%APPDATA%\Everything
```

- If [Store settings and data in %APPDATA%\Everything](#) is disabled, the HTTP Server folder must be created in the same location as your Everything.exe.
- In **Everything**, from the **Tools** menu, click **Start HTTP Server**.

- Download the following files to your HTTP Server folder:
- [Everything-HTTP.Server.Files.zip](#)
- Edit these files in the HTTP Server folder in your "*Everything*" installation folder.
- Everything will load these files instead of the embedded HTTP server files.
- Hold Shift and press the reload button to force your browser to refresh.

Change the default HTTP server page

To change the default HTTP server page:

- In ***Everything***, from the **Tools** menu, click **Options**.
- Click the **HTTP Server** tab.
- Set the **Default page** to your custom page.

Custom strings

To customize the builtin HTTP server strings:

- Download the HTTP server strings template: [http_server_strings.zip](#)
- Extract the http_server_strings.ini file to: %APPDATA%\Everything\HTTP server
- Make any changes to your http_server_strings.ini
- In Everything, type in the following search and press ENTER:

```
/http_server_strings=C:\Users\<user>\AppData\Roaming\Everything\HTTP Server\http_s
```

where <user> is your username.

- Restart the HTTP Server:
 - In ***Everything***, from the **Tools** menu, click **Options**.
 - Click the **HTTP Server** tab.
 - Uncheck **Enable HTTP Server**.
 - Click **Apply**.
 - Check **Enable HTTP Server**.
 - Click **OK**.

Security

Every file and folder indexed by Everything can be searched and downloaded via the web server.

To disable file downloading:

- In ***Everything***, from the **Tools** menu, click **Options**.
- Click the **HTTP Server** tab.
- Uncheck **allow file download**.

See [Disable HTTP Server](#) to remove the HTTP server options and prevent the HTTP server from starting.

Disable HTTP Server

To disable the HTTP server:

- Exit Everything (right click the Everything system tray icon and click Exit)
- Open your Everything.ini in the same location as your Everything.exe
- Change the following line:

```
allow_http_server=1
```

to:

```
allow_http_server=0
```

- Save changes and restart Everything.

Trouble shooting

How do I fix the Unable to start HTTP server: bind failed 10048 error?

There is already another service running on port 80.

Please try changing the Everything HTTP server port to another port.

To change the HTTP server port:

- In **Everything**, from the **Tools** menu, click **Options**
- Click the **HTTP server** tab.
- Change **Listen on port** to a new port, for example 8080.
- Click **OK**

Please make sure to specify this port when connecting to the web server with your web browser, for example:

```
http://localhost:8080
```

Range request

Everything supports range requests for streaming support.

See also

- [Multiple Instances](#).
- [HTTP Server Everything.ini options](#).
- [HTTP Server options](#).
- <http://en.wikipedia.org/wiki/HTTP>

Indexes

By default, "*Everything*" indexes all local fixed NTFS volumes.

Indexes can be customized in the [Everything options](#).

Database location

Everything stores the database in:

```
%LOCALAPPDATA%\Everything\Everything.db
```

If [Store settings and data in %APPDATA%\Everything](#) is disabled, Everything.db is stored in the same location as Everything.exe.

To set the database location:

- In "*Everything*", from the **Tools** menu, click **Options**.
- Select the **Indexes** tab.
- Set the **Database location**.
- Click **OK**.

Size, Dates and Attributes Indexing

Everything indexes file size and date modified by default.

Everything will maintain this extra file information in real-time.

File information that is indexed can be searched instantly.

Extra file information, such as date created and file attributes is not indexed.

This file information is gathered when displayed in Everything.

Searching or sorting by this file information can be quite slow for many results.

To include extra file information in the Everything index:

- In "*Everything*", from the **Tools** menu, click **Options**
- Click the **Indexes** tab.
- Check the desired options:
 - Index file size
 - Index folder size
 - Index date created
 - Index date modified
 - Index date accessed
 - Index attributes
- Click **OK**.

Note: Including extra file information in the index will require more memory.

Hover over an extra file information option to see a tooltip showing the extra memory requirements.

Folder sizes

To show folder sizes in "*Everything*":

- In "*Everything*", from the **Tools** menu, click **Options**
- Click the **Indexes** tab.
- Check **Index folder size**.
- Click **OK**.

Fast Sorting

"*Everything*" maintains a "Fast sort" index for size and date modified information.

Fast sorting is persistent, that is, "*Everything*" will remember the sort between searches.

When sorting by file information that does not have fast sorting enabled "*Everything*" must gather this information from disk which can be slow.

To enable fast sorting:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **Indexes** tab.
- Check the desired fast sort options:
 - Fast size sort
 - Fast date created sort
 - Fast date modified sort
 - Fast date accessed sort
 - Fast attribute sort
 - Fast path sort
 - Fast extension sort
- Click **OK**.

Note: Enabling fast sort requires more memory. The corresponding extra file information must be included in the index.

Hover over a fast sort option to see a tooltip showing the extra memory requirements.

Name, Date Run, Run Count and Date Recently Changed are always fast sort enabled.

NTFS

"*Everything*" will automatically index all fixed NTFS volumes.

To manually index fixed NTFS volumes:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **NTFS** tab
- uncheck **Auto include new fixed volumes**.
- Click **OK**.

To include removable NTFS volumes in the index:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **NTFS** tab
- Select the removable volume.
- Check **Include in database**.
- Click **OK**.

ReFS

"*Everything*" will automatically index all fixed ReFS volumes.

To manually index fixed ReFS volumes:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **ReFS** tab
- uncheck **Auto include new fixed volumes**.
- Click **OK**.

To include removable ReFS volumes in the index:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **ReFS** tab
- Select the removable volume.
- Check **Include in database**.
- Click **OK**.

Folders

Use folder indexing to include mapped network drives, CDROMs, DVDROMs, FAT and FAT32 volumes.

See [Folder Indexing](#) for more information.

File Lists

Use file lists to index CDs, DVDs, BluRays and other offline files and folders.

See [File lists](#) for more information.

Exclude

"Everything" will index every file and folder, you can then exclude files from the index.

To exclude files or folders with the hidden or system attributes from the *"Everything"* index:

- In *"Everything"*, from the **Tools** menu, click **Options**.
- Click the **Exclude** tab
- Check **Exclude hidden files and folders** to exclude files and folders with the hidden attribute.
- Check **Exclude system files and folders** to exclude files and folders with the system attribute.
- Click **OK**.

To exclude a folder from the *"Everything"* index:

- In *"Everything"*, from the **Tools** menu, click **Options**.
- Click the **Exclude** tab
- Click **Add Folder...**
- Select the folder to exclude.
- Click **OK**.
- Click **OK**.

To include only certain types of files in the *"Everything"* index:

- In *"Everything"*, from the **Tools** menu, click **Options**.
- Click the **Exclude** tab
- Type in the semicolon delimited list of wildcard filters for **include only files**, for example, include only mp3 and jpg files:

```
*.mp3;*.jpg
```

- Click **OK**.

To exclude certain types of files from the *"Everything"* index:

- In *"Everything"*, from the **Tools** menu, click **Options**.
- Click the **Exclude** tab
- Type in the semicolon delimited list of wildcard filters for **exclude files**, for example, exclude thumbs.db, desktop.ini and tmp files:

```
thumbs.db;desktop.ini;*.tmp
```

- Click **OK**.

To use regex in your exclude filters, prefix the exclude filter with regex:, for example, to exclude folders starting with A-N, set the exclude filter to:

regex:^(A-N)

Optimizing for smallest memory foot print

To reduce the memory requirements for *"Everything"*:

- In *"Everything"*, from the **Tools** menu, click **Options**.
- Click the **Indexes** tab
- Uncheck **Index recent changes**.
- Uncheck **Index file size**.
- Uncheck **Index folder size**.
- Uncheck **Index date created**.
- Uncheck **Index date modified**.
- Uncheck **Index date accessed**.
- Uncheck **Index attributes**.
- Uncheck **Fast path sort**.
- Uncheck **Fast extension sort**.
- Click **OK**.

Recover filenames from an Everything.db

To recover a list of filenames from a Everything.db:

- Please make sure you have a backup of your Everything.db.
- Download [db2efu.zip](#).
- Open db2efu.zip and run db2efu.exe
- Select your Everything.db file.
- Choose a filename to store the list of filenames.
- This [file list](#) file can be opened with *"Everything"*, notepad or Excel.

Force rebuild

To force *"Everything"* to rebuild its indexes:

- In *"Everything"*, from the **Tools** menu, click **Options**.
- Click the **Indexes** tab
- Click **Force Rebuild**.
- Click **OK**.

Keep offline NTFS volumes in the index

To keep offline NTFS volumes in your Everything index:

- In *"Everything"*, from the **Tools** menu, click **Options**.
- Click the **NTFS** tab
- Uncheck **Automatically remove offline volumes**.
- Click **OK**.

To manually remove an offline volume:

- In *"Everything"*, from the **Tools** menu, click **Options**.
- Click the **NTFS** tab
- Select the offline NTFS volume and click **Remove**.
- Click **OK**.

Fast reindexing

"*Everything*" must perform a reindex when a new volume is added to the index or an existing volume is removed from the index.

When a reindex occurs with "*Everything*" 1.3 and earlier, all volumes must be completely rescanned, even if they have not changed.

"*Everything*" 1.4 can quickly reindex by reusing the existing up-to-date indexes.

Limitations

While all NTFS/ReFS hard links are indexed, changes made to a hard link will only update the first hard link entry in "*Everything*".

"*Everything*" does not follow junctions with NTFS/ReFS indexing.

Pausing file system monitors

To pause NTFS monitors, ReFS monitors, folder monitors and file list monitors:

- In "***Everything***", type in the following search and press ENTER:
/monitor_pause

To resume NTFS monitors, ReFS monitors, folder monitors and file list monitors:

- In "***Everything***", type in the following search and press ENTER:
/monitor_resume

Monitors will remain paused until they are resumed.

See also

- [Indexing options](#)
- [NTFS](#)
- [Folder indexing options](#)
- [File lists options](#)
- [Exclude options](#)
- [Database command line options](#)
- [Indexing Everything.ini options](#)

INI

The Everything.ini stores all the settings for Everything.

- [General Settings](#)
- [Search Window Settings](#)
- [Search Settings](#)
- [Indexing](#)
- [ETP](#)
- [HTTP](#)
- [Keyboard](#)
- [Fonts and colors](#)

- [Hidden settings](#)

Viewing and changing settings in Everything

Most of these settings can be changed from within Everything, in the [Everything Options](#).

To show the current value for a setting in the status bar, type in the following search and press ENTER:

```
/<ini option>
```

where <ini option> is one of the ini options below. Not all ini options are supported.

For example, to check the current value for the ini option **snap**, type in the following search and press ENTER:

```
/snap
```

The current value is shown in the status bar for 3 seconds.

To change the value for a setting, type in the following search and press ENTER:

```
/<ini option>=<value>
```

where <ini option> is one of the ini options below and <value> is the new value. Not all ini options are supported.

For example, to enable the **snap** ini option, type in the following search and press ENTER:

```
/snap=1
```

The new value is shown in the status bar for 3 seconds.

Editing the Everything.ini

The Everything.ini is located in:

```
%APPDATA%\Everything
```

If [Store settings and data in %APPDATA%\Everything](#) is disabled, the Everything.ini is stored in the same location as Everything.exe.

The [Store settings and data in %APPDATA%\Everything](#) setting is stored in the Everything.ini in the same location as Everything.exe.

To open your Everything.ini file, type in the following search and press ENTER:

```
about:config
```

The Everything.ini is automatically generated. Any changes made to the Everything.ini while Everything is running are lost when Everything exits.

Please exit "*Everything*" before making changes to the Everything.ini file.

To exit Everything:

- Right click the Everything system tray icon and click Exit.

The Everything.ini should be opened and saved as UTF-8.

Syntax

```
[section]  
key=value
```

[Everything]

```
[Everything]
```

This begins the Everything section.

Only the Everything section is used.

General Settings

General settings for Everything.

app_data

The location of settings and data. If set to 1, settings and data are stored in %APPDATA%\Everything. If set to 0, settings and data are stored in the same location as the Everything.exe.

This option is always stored in the Everything.ini in the same location as your Everything.exe.

run_as_admin

Run Everything as an admin. If set to 1 and Everything was run as from a standard user account, Everything will relaunch as an admin. If set to 1 and Everything was run as an admin, Everything will continue as normal. If set to 0, Everything will not relaunch as an admin.

This option is always stored in the Everything.ini in the same location as your Everything.exe.

allow_multiple_windows

Create a new search window or show an existing search window when Everything.exe is run. If set to 1, a new search window is always created. If set to 0 and no search window exists a new one is created. If set to 0 and a search window already exists, the existing search window is shown.

run_in_background

Specify whether Everything should exit after all the search windows have been closed. If set to 1, Everything will continue to run in the background after closing all the search windows. If set to 0, Everything will exit when all search windows are closed.

show_tray_icon

Show the Everything icon in the notification area. If set to 1, the tray icon is shown. If set to 0, the tray icon is not shown.

language

The [Language Identifier](#) in decimal for displaying text. If set to 1033, English (US) is used. If set to 0, The user default language for displaying text is used.

open_folder_command2

open_file_command2

open_path_command2

explore_command2

explore_path_command2

The command to execute when opening or exploring a folder, file or path.

See [External file manager](#) for more information.

last_export_type

The last export type. Can be one of the following:

Type	Description
0	EFU File
1	CSV File
2	TXT File

check_for_updates_on_startup

Check for updates when Everything is started. Set to 1 to enable. Set to 0 to disable.

connect_history_hosts

A list of successful ETP connections.

connect_history_ports

A list of ports for each connect_history_hosts.

connect_history_usernames

A list of usernames for each connect_history_hosts.

connect_history_link_types

A list of history link types for each connect_history_hosts. Can be one of the following values:

Type	Description
0	C:\ links.
1	\\server\C links.
2	\\server\\$\C links.
3	ftp://server/C links.

search_history_enabled

Enable or disable search history. Set to 1 to enable. Set to 0 to disable.

run_history_enabled

Enable or disable run history. Set to 1 to enable. Set to 0 to disable.

search_history_days_to_keep

The number of days to keep search history.

run_history_days_to_keep

The number of days to keep run history.

search_history_keep_forever

Keep search history forever.

Set to 1 to enable. Set to 0 to disable.

run_history_keep_forever

Keep run history forever.

Set to 1 to enable. Set to 0 to disable.

search_history_always_suggest

Show search suggestions on each keystroke or only show suggestions when Ctrl + Space is pressed. Set to 1 to always show suggestions. Set to 0 to show suggestions when Ctrl + Space is pressed.

search_history_sort

The search history organizer sort.

Can be one of the following values:

Value	Description
0	Sort by search
1	Sort by search count
2	Sort by search date

search_history_sort_ascending

The search history organizer sort order.

Set to 1 to sort ascending. Set to 0 to sort descending.

search_history_x

search_history_y

The search history window position.

search_history_wide

search_history_high

The search history window dimensions.

search_history_column_search_wide

The width of the search column in pixels.

search_history_column_search_order

The position of the search column.

search_history_column_count_wide

The width of the search count column in pixels.

search_history_column_count_order

The position of the search count column.

search_history_column_date_wide

The width of the last search date column in pixels.

search_history_column_date_order

The position of the last search date column.

Search Window Settings

settings that affect appearance.

window_x

The last horizontal position of the search window's upper left corner, in screen coordinates before it was closed. New search windows are positioned based on this setting.

window_y

The last vertical position of the search window's upper left corner, in screen coordinates before it was closed. New search windows are positioned based on this setting.

window_wide

window_high

The last size of the Everything search window, in pixels before it was closed. New search windows are sized based on this setting.

maximized

Control how new search windows are displayed. If set to 1, new search windows are shown maximized. If set to 0, new search windows are not shown maximized.

minimized

Control how new search windows are displayed. If set to 1, new search windows are minimized, If set to 0, new search windows are not shown minimized. Everything does not save this value, it must be set manually.

fullscreen

Show the Everything search window in a full screen window. Set to 1 to enable full screen mode. Set to 0 to window mode.

ontop

Keep the Everything window on top of other windows. Set to 0 to never be on top. Set to 1 to be always on top. Set to 2 to be on top only while the search is not empty.

alternate_row_color

Set to 1 to alternate the result list row color. Set to 0 to use a solid color for the result list.

show_mouseover

Show the row the mouse is currently over in a different color. Set to 1 use a custom color. Set to 0 to use the default color.

show_highlighted_search_terms

Show highlighted search terms. Set to 1 to enable. Set to 0 to disable.

text_size

Offset in pixels of the font height for displaying results. Set to 0 to reset.

hide_empty_search_results

Don't show any results when the search is an empty string. Set to 1 to enable. Set to 0 to show all results when the search is empty.

clear_selection_on_search

Reset the selection when changing a search parameter. Set to 1 to enable. Set to 0 to keep the selection between searches.

show_selected_item_in_statusbar

Show the full path and filename of the currently selected item in the status bar. Set to 1 to enable. Set to 0 to disable.

translucent_selection_rectangle_alpha

The alpha level for the selection rectangle. Ranges from 0 - 255. 0 = transparent, 255 = opaque.

last_options_page

The last Everything options page. When opening a new option window, this page will be shown.

invert_layout

Toggle LTR/RTL view. Set to 1 to invert.

allow_multiple_windows_from_tray

Create a new search window or show an existing search window when opening Everything from the tray icon. Set to 1 to always create a new search window. Set to 0 to show an existing window before creating a new search window.

single_click_tray

Open the Everything search window with a single or double click from the tray icon. Set to 1 to open with a single click. Set to 0 to open with a double click.

close_on_execute

Close the Everything search window after executing an item. Set to 1 to close after executing. Set to 0 to keep the window open after executing.

double_click_path

Open the path in the path column when double clicking it. Set to 1 to allow a double click to open the path. Set to 0 to ignore the double click.

update_display_after_scroll

Force the display to update immediately after it is scrolled. Set to 1 to update after scrolling. Set to 0 to invalidate and update later.

auto_scroll_view

Automatically scroll the view when dragging the selection mask outside the window. Set to 1 to automatically scroll. Set to 0 to only scroll when the mouse is moved.

double_quote_copy_as_path

Include double quotes around the path when copied to the clipboard. Set to 1 to include the double quotes. Set to 0 to copy without the double quotes.

rename_select_filepart_only

Select only the file part only (not the extension) or the entire filename when renaming. Set to 1 to select only the file part and not select the extension. Set to 0 to select the entire filename.

select_search_on_mouse_click

Select all the search text when focusing the search edit.

focus_search_on_activate

Focus the search edit when the Everything window is activated.

single_click_open

Open a result with a single click or double click.

Set to 0 to use the system default setting for single click open. Set to 1 to open with a single click. Set to 2 to open with a double click.

underline_icon_titles

Show result text with underlines.

Set to 0 to use the system default setting for underline icon titles. Set to 1 to always show underlines. Set to 2 to never show underlines. Set to 3 to underline result text when hovering with the mouse cursor.

wrap_focus

Can be one of the following values:

Value	Description
0	focus is clamped to the result list.
1	focus wraps around the top and bottom of the result list.
2	focus wraps around the top and bottom of the result list and search edit.

load_icon_priority

load_thumbnail_priority

load_fileinfo_priority

The thread priority of the load action.

Can be one of the following values:

Value	Description
-15	Idle priority.
-1	Below normal priority.
0	Normal priority.

1	Above normal priority.
2	High priority.

full_row_select

Allow clicking anywhere in the row to select an item and show the selection for the entire row.
Set to 1 to enable. Set to 0 to disable.

double_buffer

Use an off screen buffer when rendering the result list. Set to 1 to use the off screen buffer. Set to 0 to render directly to the screen. Use a double buffer to remove flickering when rendering. Drawing directly to the screen may improve performance on older computers.

show_number_of_results_with_selection

Show the total number of results with the selection count in the status bar. Set to 1 to enable. Set to 0 to disable.

date_descending_first

Show dates in descending order first.

size_descending_first

Show sizes in descending order first.

size_format

The size format type, can be one of the following:

Type	Description
0	Best format depending on the size.
1	Always show sizes in bytes.
2	Always show sizes in KB.
3	Always show sizes in MB.

alpha_select

Draw the selection mask as a translucent rectangle or a XOR outline mask. Set to 1 to use a translucent rectangle. Set to 2 to use a XOR outline mask. Set to 0 to use the system default selection rectangle.

show_size_in_statusbar

Set to 1 to show the selection or total size in status bar.
Set to 0 to disable.
Enabling can decrease search performance.

tooltips

Show or don't show tooltips in the option window. Set to 1 to show tooltips. Set to 0 to not show tooltips.

editor_x

editor_y

The last file list editor window position.

editor_wide

editor_high

The last file list editor window dimensions.

editor_maximized

The file list editor is maximized.

Set to 1 to enable. Set to 0 to disable.

rename_x

rename_y

The multi-file rename window position.

rename_wide

rename_high

The multi-file rename window dimensions.

rename_match_case

Match or ignore case in a multi-file rename operation.

Set to 1 to match case. Set to 0 to ignore case.

rename_regex

Enable regex in a multi-file rename operation.

Set to 1 to enable. Set to 0 to disable.

advanced_copy_to_x

advanced_copy_to_y

The advanced copy to window position.

advanced_copy_to_wide

advanced_copy_to_high

The advanced copy to window dimensions.

advanced_copy_to_match_case

Match or ignore case in a advanced copy to operation.

Set to 1 to match case. Set to 0 to ignore case.

advanced_copy_to_regex

Enable regex in a advanced copy to operation.

Set to 1 to enable. Set to 0 to disable.

advanced_move_to_x

advanced_move_to_y

The advanced move to window position.

advanced_move_to_wide

advanced_move_to_high

The advanced move to window dimensions.

advanced_move_to_match_case

Match or ignore case in a advanced move to operation.

Set to 1 to match case. Set to 0 to ignore case.

advanced_move_to_regex

Enable regex in a advanced move to operation.

Set to 1 to enable. Set to 0 to disable.

advanced_search_x

advanced_search_y

The advanced search window position.

advanced_search_wide

advanced_search_high

The advanced search window dimensions.

advanced_search_page_y_offset

The vertical scroll offset from the top of the advanced search page.

advanced_search_focus_id

The ID of the advanced search item that had focus last.

show_copy_name

Show or do not show the copy name context menu item. Set to 1 to show the context menu item. Set to 0 to not show the context menu item. Set to 2 to use the default show setting. Set to 3 to show when holding down the Shift key.

show_copy_path

Show or do not show the copy path context menu item. Set to 1 to show the context menu item. Set to 0 to not show the context menu item. Set to 2 to use the default show setting. Set to 3 to show when holding down the Shift key.

show_copy_full_name

Show or do not show the copy full name context menu item. Set to 1 to show the context menu item. Set to 0 to not show the context menu item. Set to 2 to use the default show setting. Set to 3 to show when holding down the Shift key.

show_open_path

Show or do not show the open path context menu item. Set to 1 to show the context menu item. Set to 0 to not show the context menu item. Set to 2 to use the default show setting. Set to 3 to show when holding down the Shift key.

show_explore

Show or do not show the Explore context menu item. Set to 1 to show the context menu item. Set to 0 to not show the context menu item. Set to 2 to use the default show setting. Set to 3 to show when holding down the Shift key.

show_explore_path

Show or do not show the Explore path context menu item. Set to 1 to show the context menu item. Set to 0 to not show the context menu item. Set to 2 to use the default show setting. Set to 3 to show when holding down the Shift key.

filters_visible

Show the filter bar in the toolbar.

Set to 1 to show. Set to 0 to hide.

filters_wide

Width of the filter bar in pixels.

filters_right_align

Left or right alignment of the filter bar. Set to 1 to align to the right. Set to 0 to align to the left.

filter_everything_name;

The name of the Everything filter.

filter_organize_x

filter_organize_y

The filter organize window position.

filter_organize_wide

filter_organize_high

The filter organize window dimensions.

preview_visible

Show or hide the preview pane.

Set to 1 to show. Set to 0 to hide.

preview_x

The width of the preview pane in pixels.

name_column_pos

path_column_pos

size_column_pos

extension_column_pos

type_column_pos

last_write_time_column_pos

creation_time_column_pos

date_accessed_column_pos

attribute_column_pos

file_list_filename_column_pos

date_run_column_pos

run_count_column_pos

date_recently_changed_column_pos

The position of the column. 0 being the left most.

name_column_width

path_column_width

size_column_width
extension_column_width
type_column_width
last_write_time_column_width
creation_time_column_width
date_accessed_column_width
attribute_column_width
file_list_filename_column_width
date_run_column_width
run_count_column_width
date_recently_changed_column_width

The width of the column in pixels.

path_column_visible
size_column_visible
extension_column_visible
type_column_visible
last_write_time_column_visible
creation_time_column_visible
date_accessed_column_visible
attribute_column_visible
file_list_filename_column_visible
date_run_column_visible
run_count_column_visible
date_recently_changed_column_visible

Visibility of the column. Set to 1 to show. Set to 0 to hide.

Search Settings

Settings that affect the search.

match_whole_word

Match whole words. Set to 1 to enable. Set to 0 to match anywhere.

match_path

Match full paths. Set to 1 to enable. Set to 0 to match filenames only.

match_case

Match case. Set to 1 to enable. Set to 0 to ignore case.

match_diacritics

Match diacritical marks. Set to 1 to enable. Set to 0 to disable.

match_regex

Use regular expressions. Set to 1 to enable. Set to 0 to disable.

home_match_case

The match case value for the home search. Set to 0 to use the last value. set to 1 to use a custom value.

home_match_whole_word

The match whole word value for the home search. Set to 0 to use the last value. set to 1 to use a custom value.

home_match_path

The match path value for the home search. Set to 0 to use the last value. set to 1 to use a custom value.

home_match_diacritics

The match diacritics value for the home search. Set to 0 to use the last value. set to 1 to use a custom value.

home_regex

The Regex value for the home search. Set to 0 to use the last value. set to 1 to use a custom value.

home_search

The search type for the home search. Set to 0 to use the last value. Set to 1 to use a custom search.

home_filter

The filter type for the home search. Set to 0 to use the last value. Set to 1 to use a custom filter.

home_sort

The sort type for the home search. Set to 0 to use the last value. Set to 1 to use a custom sort.

home_view

The result list view mode for the home search.

Can be one of the following values:

Value	Description
0	Details
1	Thumbnails

home_index

The index type for the home search. Set to 0 to use the last value. Set to 1 to use a custom index.

fast_ascii_search

Use single byte ASCII lower case comparisons. Set to 0 to fix Turkish I with dot mappings.

match_path_when_search_contains_path_separator

Enable full path matching when the search contains a backslash (\). Set to 1 to enable. Set to 0 to disable.

allow_literal_operators

Allow use of the literal operators AND, OR and NOT. Set to 1 to allow. Set to 0 to disallow. Can cause conflicts if enabled and a filename contains the text AND, OR or NOT.

allow_round_bracket_parenthesis

Allow (and) for grouping. Set to 1 to allow. Set to 0 to disallow. Can cause conflicts if enabled and a filename contains a (or).

expand_environment_variables

Expand environment strings inside % quotes. Set to 1 to enable. Set to 0 to disable. Can cause conflicts if enabled and a filename contains a %.

search_as_you_type

Search as you type or search when you press enter. Set to 1 to search as you type. Set to 0 to search when you press enter.

convert_forward_slash_to_backslash

Convert backslashes (/) to forward slashes (\). Set to 1 to convert / to \. Set to 0 to allow /. Can cause conflicts if enabled and a filename contains a /. Note: filenames on NTFS volumes can not contain a /. It is possible to use / in file lists for URLs.

match_whole_filename_when_using_wildcards

Match the whole filename or anywhere when using wildcards. Set to 1 to match the whole filename. Set to 0 to match anywhere.

operator_precedence

Order of operations.

Can be one of the following values:

Value	Description
0	OR > AND
1	AND > OR

search

The last search string. Only valid if home_search is set to 1.

bookmark_remember_case

bookmark_remember_wholeword

bookmark_remember_path

bookmark_remember_diacritic

bookmark_remember_regex

bookmark_remember_sort

bookmark_remember_view

bookmark_remember_filter

bookmark_remember_index

Save the search settings to the bookmark. Set to 1 to remember the setting. Set to 0 to keep the current setting. If enabled, when the bookmark is opened the search settings are restored.

bookmark_organize_x

bookmark_organize_y

The current bookmark window position.

bookmark_organize_wide

bookmark_organize_high

The current bookmark window dimensions.

filter

The last filter name. If home_filter is 1, this filter is restored when opening a new search window.

sort

The last sort. If home_sort is 1, this sort is restored when opening a new search window. If always_keep_sort is 0, this value will be ignored if it is a slow sort.

Can be one of the following sort types:

Type	Description
0	Name
1	Path
2	Extension
3	Type
4	Date modified
5	Date created
6	Date accessed
7	Attributes
8	Date recently changed
9	Run count
10	Date run
11	File list filename

sort_ascending

Sort ascending or descending. Set to 1 to sort ascending. Set to 0 to sort descending.

index

The last index type. If home_index is 1, this index is restored when opening a new search window.

Can be one of the follow index types:

Type	Description
0	Local DB.
1	File list.
2	ETP server.

index_file_list

The file list filename when the last index is a file list.

index_etp_server

The ETP server when the last index is a ETP server.

index_link_type

The ETP link type when the last index is a ETP server.

view

The result list view mode.

Can be one of the following values:

Value	Description
0	Details
1	Thumbnails

thumbnail_size

The thumbnail size in pixels.

status_bar_visible

Show or hide the status bar. Set to 1 to show the status bar. Set to 0 to hide the status bar.

Indexing

settings that affect indexing.

auto_include_fixed_volumes

Automatically rebuild the database to include a new fixed NTFS volume has been added to the system.

auto_include_removable_volumes

Automatically rebuild the database to include a new removable NTFS volume has been added to the system.

auto_remove_offline_ntfs_volumes

Automatically remove offline NTFS volumes included in the index.

If disabled, offline volumes can be removed manually.

auto_include_fixed_refs_volumes

Automatically rebuild the database to include a new fixed ReFS volume has been added to the system.

auto_include_removable_refs_volumes

Automatically rebuild the database to include a new removable ReFS volume has been added to the system.

auto_remove_offline_refs_volumes

Automatically remove offline ReFS volumes included in the index.

If disabled, offline volumes can be removed manually.

exclude_list_enabled

Enable or disable the exclude list. Set to 1 to enable. Set to 0 to disable.

exclude_hidden_files_and_folders

Exclude or include hidden files and folders. Set to 1 to exclude hidden files and folders. Set to 0 to include hidden files and folders.

exclude_system_files_and_folders

Exclude or include system files and folders. Set to 1 to exclude system files and folders. Set to 0 to include system files and folders.

include_only_files

semicolon delimited list of which extensions to include. eg: *.mp3;*.jpg

Set to an empty string to include all file types.

exclude_files

semicolon delimited list of which extensions to exclude. eg: *.db

Set to an empty string to not exclude any file types.

db_location

The path where the Everything.db is located.

db_multi_user_filename

Use a unique filename for the database for each computer and user. Set to 1 to enable. Set to 0 to disable. Might be useful when using the portable version on different computers.

db_compress

Use Bzip 2 to compress the database. Set to 1 to enable. Set to 0 to disable. Can improve loading performance on slow hard drives at the cost of higher CPU usage when loading and saving.

index_size

Include file size information in the index.

Set to 1 to enable. Set to 0 to disable.

fast_size_sort

Maintain a size sorted index for instant size sorting.

Set to 1 to enable. Set to 0 to disable.

index_date_created

Include file and folder date created information in the index.

Set to 1 to enable. Set to 0 to disable.

fast_date_created_sort

Maintain a date created sorted index for instant date created sorting.

Set to 1 to enable. Set to 0 to disable.

index_date_modified

Include file and folder date modified information in the index.

Set to 1 to enable. Set to 0 to disable.

fast_date_modified_sort

Maintain a date modified sorted index for instant date modified sorting.

Set to 1 to enable. Set to 0 to disable.

index_date_accessed

Include file and folder date accessed information in the index.

Set to 1 to enable. Set to 0 to disable.

fast_date_accessed_sort

Maintain a date accessed sorted index for instant date accessed sorting.

Set to 1 to enable. Set to 0 to disable.

index_attributes

Include file and folder attributes information in the index.

Set to 1 to enable. Set to 0 to disable.

fast_attributes_sort

Maintain an attributes sorted index for instant attributes sorting.

Set to 1 to enable. Set to 0 to disable.

index_folder_size

Include folder size information in the index.

Set to 1 to enable. Set to 0 to disable.

fast_path_sort

Maintain a path sorted index for instant path sorting.

Set to 1 to enable. Set to 0 to disable.

fast_extension_sort

Maintain an extension sorted index for instant extension sorting.

Set to 1 to enable. Set to 0 to disable.

extended_information_cache_monitor

Monitor changes to file sizes, dates and attributes. Set to 1 to enable. Set to 0 to disable.

index_recent_changes

Maintain a list of changes to the database. These changes can be searched, or all results can be sorted by date recently changed.

ntfs_volume_guids

The list of NTFS volumes GUIDs to potentially include in the database.

ntfs_volume_paths

The list of NTFS volumes paths to potentially include in the database.

ntfs_volume_roots

The list of NTFS volumes roots to potentially include in the database. The root is a path without the root part to specify the base folder to index.

ntfs_volume_includes

List of include or exclude values for each NTFS volume in the `ntfs_volume_path` list. Set to 1 to include. Set to 0 to exclude.

ntfs_volume_load_recent_changes

List of load recent change values for each NTFS volume in the `ntfs_volume_path` list. Set to 1 to load recent changes. Set to 0 to not load recent changes.

ntfs_volume_include_onlys

List of load include only values for each NTFS volume in the `ntfs_volume_path` list. Set to an empty string to include all folders. Set to a semicolon delimited list of folders to exclude.

ntfs_volume_monitors

List of monitor values for each NTFS volume in the `ntfs_volume_path` list. Set to 1 to monitor changes. Set to 0 to not monitor changes.

refs_volume_guids

The list of ReFS volumes guides to potentially include in the database.

refs_volume_paths

The list of ReFS volumes paths to potentially include in the database.

refs_volume_roots

The list of ReFS volumes roots to potentially include in the database. The root is a path without the root part to specify the base folder to index.

refs_volume_includes

List of include or exclude values for each ReFS volume in the `refs_volume_path` list. Set to 1 to include. Set to 0 to exclude.

refs_volume_load_recent_changes

List of load recent change values for each ReFS volume in the `refs_volume_path` list. Set to 1 to load recent changes. Set to 0 to not load recent changes.

refs_volume_include_onlys

List of load include only values for each ReFS volume in the `refs_volume_path` list. Set to an empty string to include all folders. Set to a semicolon delimited list of folders to exclude.

refs_volume_monitors

List of monitor values for each ReFS volume in the `refs_volume_path` list. Set to 1 to monitor changes. Set to 0 to not monitor changes.

filelists

A list of file list filenames to include in the index.

folders

A comma separated list of folders to include in the index.

folder_monitor_changes

A comma separated list of monitor values for each folder in the folders list. Set to 1 to monitor changes. Set to 0 to not monitor changes.

folder_buffer_size_list

A comma separated list of buffer sizes in bytes for each folder in the index.

folder_rescan_if_full_list

A comma separated list of rescan if full option for each folder in the index.
Set to 1 to rescan if full. Set to 0 to ignore changes if buffer is full.

folder_update_types

A comma separated list of update types for each folder in the folders list. Can be one of the following values:

Type	Description
0	Never update.
1	Update at a specified interval.
2	Update on the specified day.

folder_update_days

A comma separated list of update days for each folder in the folders list. Can be one of the following values:

Type	Description
0	Update everyday.
1	Update every Sunday
2	Update every Monday
3	Update every Tuesday
4	Update every Wednesday
5	Update every Thursday
6	Update every Friday
7	Update every Saturday

folder_update_ats

A comma separated list of update ats in hours for each folder in the folders list. Can be in the range from 0 - 23.

folder_update_intervals

A comma separated list of update intervals in minutes or hours (depending on folder_update_interval_types) for each folder in the folders list.

folder_update_interval_types

A comma separated list of update intervals types for each folder in the folders list. Can be one of the following values:

Type	Description
0	Minutes
1	Hours

exclude_folders

A comma separated list of folders to exclude. Set to an empty string to include all folders.

ETP

settings that affect the ETP server

etp_server_enabled

Enable or disable the ETP server. Set to 1 to enable the ETP server. Set to 0 to disable the ETP server.

etp_server_bindings

The ETP server listening socket bindings. Set to an empty string to bind to all interfaces.

etp_server_port

The ETP server listening socket port.

etp_server_username

The ETP server username. Set to an empty string to ignore.

etp_server_password

The ETP server password. Set to an empty string to ignore.

etp_server_welcome_message

The ETP welcome message.

The following macros can be included:

Macro name	Description
%%	A single %
%v	Current Everything version.
%n	A new line.

Leave empty to use the default welcome message.

etp_server_log_file_name

The filename of the ETP server log. Set to an empty string to use the default filename.

etp_server_logging_enabled

Enable or disable ETP server logging. Set to 1 to enable. Set to 0 to disable.

etp_server_log_max_size

The maximum size of the ETP server log file in bytes. If the log file exceeds this size it is truncated [etp_server_log_delta_size](#) bytes.

etp_server_log_delta_size

The number of bytes removed from the start of the ETP server log when it exceeds the maximum size.

etp_server_allow_file_download

Allow or disallow file downloading from the ETP server. Set to 1 to allow. Set to 0 to disallow.

HTTP

Settings that affect the HTTP server

http_server_enabled

Enable or disable the HTTP server. Set to 1 to enable. Set to 0 to disable.

http_server_bindings

The HTTP server listening socket bindings. Set to an empty string to bind to all interfaces.

http_server_port

The HTTP server listening socket port.

http_server_username

The HTTP server username. Set to an empty string to ignore.

http_server_password

The HTTP server password. Set to an empty string to ignore.

http_server_home

The HTTP server folder.

http_server_default_page

The HTTP server default page.

http_server_log_file_name

The HTTP server log filename. Set to an empty string to use the default log filename.

http_server_logging_enabled

Enable or disable HTTP server logging. Set to 1 to enable. Set to 0 to disable.

http_server_log_max_size

The maximum size of the HTTP server log file in bytes. If the log file exceeds this size it is truncated [http_server_log_delta_size](#) bytes.

http_server_log_delta_size

The number of bytes removed from the start of the HTTP server log when it exceeds the maximum size.

http_server_allow_file_download

Allow or disallow file downloading from the HTTP server. Set to 1 to allow. Set to 0 to disallow.

Keyboard

Settings for hotkeys and keyboard shortcuts:

new_window_key

show_window_key

toggle_window_key

The hotkey code. The LOBYTE is the Virtual key. HIBYTE is the modifier flags.

The modifier flags can be 0 or more of the following:

Value	Description
0x01	Control
0x02	Alt
0x04	Shift
0x08	Windows key

file_new_search_window_keys

file_open_file_list_keys

file_close_file_list_keys

file_close_keys

file_export_keys

file_copy_full_name_to_clipboard_keys

file_copy_path_to_clipboard_keys

file_set_run_count_keys

file_create_shortcut_keys

file_delete_keys

file_delete_permanently_keys

file_edit_keys

file_open_keys

file_open_selection_and_close_everything_keys

file_explore_path_keys

file_open_new_keys

file_open_path_keys

file_open_with_keys

file_open_with_default_verb_keys

file_play_keys

file_preview_keys

file_print_keys

file_print_to_keys

file_properties_keys

file_read_extended_information_keys

file_rename_keys

file_run_as_keys

file_exit_keys
file_copy_name_to_clipboard_keys
file_open_selection_and_do_not_close_everything_keys
file_open_most_run_keys
file_open_last_run_keys
file_custom_verb_1_keys
file_custom_verb_2_keys
file_custom_verb_3_keys
file_custom_verb_4_keys
file_custom_verb_5_keys
file_custom_verb_6_keys
file_custom_verb_7_keys
file_custom_verb_8_keys
file_custom_verb_9_keys
file_custom_verb_10_keys
file_custom_verb_11_keys
file_custom_verb_12_keys
indexes_folders_rescan_all_now_keys
indexes_force_rebuild_keys
edit_cut_keys
edit_copy_keys
edit_paste_keys
edit_select_all_keys
edit_invert_selection_keys
edit_copy_to_folder_keys
edit_move_to_folder_keys
edit_advanced_advanced_copy_to_folder_keys
edit_advanced_advanced_move_to_folder_keys
view_filters_keys
view_preview_keys
view_status_bar_keys
view_details_keys
view_medium_thumbnails_keys
view_large_thumbnails_keys
view_extra_large_thumbnails_keys
view_increase_thumbnail_size_keys
view_decrease_thumbnail_size_keys
view_window_size_small_keys
view_window_size_medium_keys
view_window_size_large_keys
view_window_size_auto_fit_keys
view_zoom_zoom_in_keys
view_zoom_zoom_out_keys
view_zoom_reset_keys
view_go_to_back_keys
view_go_to_forward_keys
view_go_to_home_keys
view_go_to_show_all_history_keys
view_sort_by_name_keys
view_sort_by_path_keys
view_sort_by_size_keys
view_sort_by_extension_keys
view_sort_by_type_keys

view_sort_by_date_modified_keys
view_sort_by_date_created_keys
view_sort_by_attributes_keys
view_sort_by_file_list_filename_keys
view_sort_by_run_count_keys
view_sort_by_date_run_keys
view_sort_by_date_recently_changed_keys
view_sort_by_date_accessed_keys
view_sort_by_ascending_keys
view_sort_by_descending_keys
view_refresh_keys
view_fullscreen_keys
view_toggle_ltrrtl_direction_keys
view_on_top_never_keys
view_on_top_always_keys
view_on_top_while_searching_keys
search_match_case_keys
search_match_whole_word_keys
search_match_path_keys
search_match_diacritics_keys
search_enable_regex_keys
search_advanced_search_keys
search_add_to_filters_keys
search_organize_filters_keys
bookmarks_add_to_bookmarks_keys
bookmarks_organize_bookmarks_keys
tools_options_keys
tools_console_keys
tools_file_list_editor_keys
tools_connect_to_etp_server_keys
tools_disconnect_from_etp_server_keys
help_everything_help_keys
help_search_syntax_keys
help_regex_syntax_keys
help_command_line_options_keys
help_everything_website_keys
help_check_for_updates_keys
help_about_everything_keys
help_donate_keys
search_edit_focus_search_edit_keys
search_edit_delete_previous_word_keys
search_edit_auto_complete_search_keys
search_edit_show_search_history_keys
search_edit_show_all_search_history_keys
result_list_item_up_keys
result_list_item_down_keys
result_list_page_up_keys
result_list_page_down_keys
result_list_start_of_list_keys
result_list_end_of_list_keys
result_list_item_up_extend_keys
result_list_item_down_extend_keys
result_list_page_up_extend_keys

result_list_page_down_extend_keys
result_list_start_of_list_extend_keys
result_list_end_of_list_extend_keys
result_list_focus_up_keys
result_list_focus_down_keys
result_list_focus_page_up_keys
result_list_focus_page_down_keys
result_list_focus_start_of_list_keys
result_list_focus_end_of_list_keys
result_list_focus_up_extend_keys
result_list_focus_down_extend_keys
result_list_focus_page_up_extend_keys
result_list_focus_page_down_extend_keys
result_list_focus_start_of_list_extend_keys
result_list_focus_end_of_list_extend_keys
result_list_focus_result_list_keys
result_list_focus_highest_run_count_result_keys
result_list_focus_last_run_result_keys
result_list_toggle_path_column_keys
result_list_toggle_size_column_keys
result_list_toggle_extension_column_keys
result_list_toggle_type_column_keys
result_list_toggle_date_modified_column_keys
result_list_toggle_date_created_column_keys
result_list_toggle_attributes_column_keys
result_list_toggle_file_list_filename_column_keys
result_list_toggle_run_count_column_keys
result_list_toggle_date_recently_changed_column_keys
result_list_toggle_date_accessed_column_keys
result_list_toggle_date_run_column_keys
result_list_size_all_columns_to_fit_keys
result_list_size_result_list_to_fit_keys
result_list_context_menu_keys
result_list_scroll_left_or_thumbnail_left_keys
result_list_scroll_right_or_thumbnail_right_keys
result_list_scroll_page_left_or_thumbnail_focus_left_keys
result_list_scroll_page_right_or_thumbnail_focus_right_keys
result_list_left_extend_keys
result_list_right_extend_keys
result_list_focus_left_extend_keys
result_list_focus_right_extend_keys
result_list_select_focus_keys
result_list_toggle_focus_selection_keys
result_list_copy_as_csv_keys
preview_focus_preview_keys

Comma delimited list of keyboard shortcuts for the command. The LOBYTE is the virtual key. The HIBYTE can be 0 or more of the follow values:

Flag	Description
0x01	Control modifier.
0x02	Alt modifier.

0x04	Shift modifier.
0x08	Windows key modifier.
0x10	Use in search edit only.
0x20	Use in result list only.

Fonts and colors

settings for fonts and colors.

result_list_font

The family name of the result list font.

result_list_font_size

The size of the result list font in pixels.

search_edit_font

The family name of the search edit font.

search_edit_font_size

The size of the search edit font in pixels.

status_bar_font

The family name of the status bar font.

status_bar_font_size

The size of the status bar font in pixels.

header_font

The family name of the result list header font.

header_font_size

The size of the result list header font in pixels.

normal_background_color

current_sort_alternate_row_highlighted_foreground_color

current_sort_alternate_row_highlighted_background_color

current_sort_alternate_row_foreground_color

current_sort_alternate_row_background_color

alternate_row_highlighted_foreground_color

alternate_row_highlighted_background_color

alternate_row_foreground_color

alternate_row_background_color

mouseover_current_sort_highlighted_foreground_color

mouseover_current_sort_highlighted_background_color

current_sort_mouseover_foreground_color
current_sort_mouseover_background_color
mouseover_highlighted_foreground_color
mouseover_highlighted_background_color
mouseover_foreground_color
mouseover_background_color
highlighted_current_sort_foreground_color
highlighted_current_sort_background_color
current_sort_foreground_color
current_sort_background_color
highlighted_drop_target_foreground_color
highlighted_drop_target_background_color
drop_target_foreground_color
drop_target_background_color
highlighted_selected_inactive_foreground_color
highlighted_selected_inactive_background_color
selected_inactive_foreground_color
selected_inactive_background_color
highlighted_selected_foreground_color
highlighted_selected_background_color
selected_foreground_color
selected_background_color
highlighted_foreground_color
highlighted_background_color
normal_foreground_color

The item's color. Set to #RRGGBB, RR is the red hex value, GG is the green hex value and BB is the blue hex value. Set to an empty string to use the default color.

current_sort_alternate_row_highlighted_bold
current_sort_alternate_row_bold
alternate_row_highlighted_bold
alternate_row_bold
mouseover_current_sort_highlighted_bold
current_sort_mouseover_bold
mouseover_highlighted_bold
mouseover_bold
highlighted_current_sort_bold
current_sort_bold
highlighted_drop_target_bold
drop_target_bold
highlighted_selected_inactive_bold
selected_inactive_bold
highlighted_selected_bold
selected_bold
highlighted_bold
normal_bold

The item's style, can be one of the following values:

Style	Description
0	Normal
1	Default style

Hidden settings

These settings are not available in the [Everything UI](#).

These settings are only available in the Everything.ini.

allow_http_server

Set to 1 to allow the HTTP server to be enabled.

Set to 0 to disable the HTTP server and remove it from the interface.

This option is only available in the Everything.ini in the same location as your Everything.exe.

See [http_server_enabled](#) to enable the HTTP server.

allow_etp_server

Set to 1 to allow the ETP server to be enabled.

Set to 0 to disable the ETP server and remove it from the interface.

This option is only available in the Everything.ini in the same location as your Everything.exe.

See [etp_server_enabled](#) to enable the ETP server.

allow_multiple_instances

Allow a new Everything process when one is already running. If set to 1, a new process is created every time you run Everything (Not recommended, use [named instances](#) instead). If set to 0 and an Everything process already exists, the commands line options are sent to the existing process. If set to 0 and there is no existing Everything process, a new Everything process is created.

beta_updates

Include beta updates when checking for updates. If set to 1, beta updates are included. If set to 0, only stable releases are included.

window_title_format

taskbar_notification_title_format

Window title or taskbar notification.

See [Change the Everything window caption](#) for more information.

instance_name

The instance name.

See [Multiple Instances](#) for more information.

max_threads

Limit the number of threads Everything uses. Set to 0 to use all available threads. Set to 1 to use only one thread.

reuse_threads

Reuse threads. Set to 1 to put threads to sleep when they complete. The thread is reused when a new thread is required. Set to 0 to terminate threads when they complete. Disabling can cause memory leaks.

max_recv_size

The maximum packet size to receive over TCP-IP in bytes.

search_history_always_suggest_extend_toolbar

Always extend the toolbar height when always suggestion search history is enabled.
Set to 1 to extend the toolbar. Set to 0 to disable.

search_history_visible_count_max

Set the maximum number of visible search history items.

search_history_always_suggest_visible_count_max

Set the maximum number of visible search history items when always show search suggestions is enabled.

search_history_show_all_max

Set the maximum number of search history results when showing all search history.

search_history_suggestion_max

Set the maximum number of search history results when suggesting a search.

search_history_show_all_sort

Set the sort order for search history when showing all search history.
Can be one of the following values:

Value	Description
0	Sort by search
1	Sort by search count
2	Sort by search date

search_history_suggestion_sort

Set the sort order for search history when showing search suggestions.
Can be one of the following values:

Value	Description
0	Sort by search
1	Sort by search count
2	Sort by search date

search_history_show_above

Show the search history drop down list above or below the search edit. Set to 1 to show above. Set to 0 to show below.

debug

Show the debug console. Set to 1 to show the debug console when Everything starts. Set to 0 to not show the debug console when Everything starts.

debug_log

Log debug messages to %TEMP%\Everything Debug Log.txt
Set to 1 to enable debugging mode and debug logging. Set to 0 to disable.

verbose

Enable verbose debugging.

min_zoom

The minimum pixel offset of the zoom.

max_zoom

The maximum pixel offset of the zoom.

single_parent_context_menu

Use context menus for single parents only.

Set to 1 to enable single parents only.

Set to 0 to disable single parents only.

If enabled, context menu items of files in different paths will only work for items in the same location as the focused item (or first selected item).

context_menu_type

The maximum context menu type to use. Can be one of the following settings:

Type	Description
0	Use highest context menu interface type possible.
1	Only IID_IContextMenu types.
2	Support IID_IContextMenu or IID_IContextMenu2 types.
3	Support IID_IContextMenu, IID_IContextMenu2 or IID_IContextMenu3 types.

auto_size_1

auto_size_2

auto_size_3

The window height in pixels to use when auto sizing.

auto_size_aspect_ratio_x

auto_size_aspect_ratio_y

The aspect ratio x:y to use when auto sizing.

auto_size_width_only

Set to 1 to enable, set to 0 to disable.

When enabled the height is unchanged when auto-sizing.

auto_size_path_x

auto_size_path_y

The ratio x:y to use for auto sizing the name and path column.

sticky_vscroll_bottom

When the vertical scrollbar is scrolled to the bottom, it will remain at the bottom when resizing the window or when the results change.

draw_focus_rect

Disable the focus rect drawn around result list items when they have focus.

date_format

The format to display dates. This is the same format `GetDateFormat` uses. Leave blank to use the default date format.

See [customizing#date_format](#) for more information.

time_format

The format to display times. This is the same format `GetTimeFormat` uses. Leave blank to use the default time format.

See [customizing#time_format](#) for more information.

listview_item_high

The height of a result list item in pixels. Set to 0 to use the default height.

update_display_after_mask

Force the display to update immediately after changing the selection mask. Set to 1 to update immediately. Set to 0 to invalidate and update later. Setting this to 1 can improve performance when the mouse updates faster than 40hz.

rename_move_caret_to_selection_end

Move the caret to the selection end or keep the selection when renaming. Set to 1 to move the caret to the end of the selection. Set to 0 to move the caret from the current position.

search_edit_move_caret_to_selection_end

Move the caret to the selection end or keep the selection when focusing the search edit. Set to 1 to move the caret to the end of the selection. Set to 0 to move the caret from the current position.

reset_vscroll_on_search

Scroll the results to the top when the search changes.

header_high

Custom header height in pixels. Set to 0 to use the default header height.

hide_on_close

Instead of closing the window, hide it temporarily. when a new search window is requested this window is made visible instead. Useful for keeping extended file information cached.

winmm

Automatically plays the focused item with Windows Multimedia API `mciSendString`. Set to 1 to enable. Set to 0 to disable.

menu_escape_amp

Escape & characters in menu items. Set to 0 to define custom keyboard shortcuts for bookmarks and filters. Custom keyboard shortcuts can be defined by prefixing a letter with the & character.

menu_folders

Enable submenus by including \ in your macro and bookmark names.
Set to 1 to enable. Set to 0 to disable.

For example, to use the bookmark name "Beatles" in a submenu called "Music", call your bookmark Music\Beatles

menu_folder_separator

Change the sub menu separator \ to the specified string.

menu_items_per_column

Split menus into columns and limit each column to the specified number of items. set to 0 to disable.

new_inherit

Inherit search settings from an existing window.

Set to 1 to enable. Set to 0 to disable.

When enabled, creating a new window from an existing window will create a new window with the same settings otherwise the default settings are used.

rtl_listview_edit

Use a right to left layout edit when renaming a file that contains a right to left character. Set to 1 to use a right to left edit. Set to 0 to always use a left to right edit.

force_path_ltr_order

Force the path to be shown in Left to Right order.

Set to 1 to enable. Set to 0 to disable.

force_path_left_align

Force the path to be left aligned.

Set to 1 to enable. Set to 0 to disable.

date_time_order

Change the order of date and time.

Can be one of the following values:

Value	Description
0	Date Time for LTR localizations, Time Date for RTL localizations
1	Date Time
2	Time Date
3	Time Date for LTR localizations, Date Time for RTL localizations

date_time_align

Change the alignment of date and time text.

Can be one of the following values:

Value	Description
-------	-------------

0	Left aligned for LTR localizations, Right aligned for RTL localizations
1	Left aligned
2	Right aligned
3	Right aligned for LTR localizations, Left aligned for RTL localizations

size_align

Change the alignment of size text.

Can be one of the following values:

Value	Description
0	Left aligned for LTR localizations, Right aligned for RTL localizations
1	Left aligned
2	Right aligned
3	Right aligned for LTR localizations, Left aligned for RTL localizations

update_layout_on_input_language_change

Automatically change the layout to match the current input language.

Set to 1 to enable. Set to 0 to disable.

control_shift_action

Set the action when pressing the Control and Shift keys.

Can be one of the following values:

Value	Description
0	Do nothing
1	Change input
2	Change layout
3	Change input and layout

change_search rtl_reading_action

Change the search edit Right to left reading order action

Can be one of the following values:

Value	Description
0	Do nothing
1	Change input
2	Change layout
3	Change input and layout

invert_layout_action

Change the invert layout action
Can be one of the following values:

Value	Description
0	Do nothing
1	Change input
2	Change layout
3	Change input and layout

file_list_relative_paths

Use relative paths with file lists.
Set to 1 to enable. Set to 0 to disable.

advanced_search_warnings

Show warning icons on advanced search fields that could perform slowly.

display_full_path_name

Display the full path and filename in the name column. Set to 1 to enable. Set to 0 to disable.

themed_toolbar

Use visual themes or windows classic style to render the toolbar. Set to 1 to use visual themes. Set to 0 to use Windows classic style.

copy_path_folder_append_backslash

Append a backslash to copied folder names. Set to 1 to append a backslash. Set to 0 to not append a backslash.

custom_verb01

custom_verb02

custom_verb03

custom_verb04

custom_verb05

custom_verb06

custom_verb07

custom_verb08

custom_verb09

custom_verb10

custom_verb11

custom_verb12

Define a custom verb for the custom verb keyboard shortcuts.

filters_tab_stop

Include the filter bar when cycling through the control focus with TAB. Set to 1 to include the filter bar. Set to 0 to exclude the filter bar.

preview_tab_stop

Allow TAB to focus the preview pane.

translucent_selection_rectangle_background_color

The color of the translucent selection rectangle. Set to #RRGGBB, RR is the red hex value, GG is the green hex value and BB is the blue hex value. Set to an empty string to use the default color.

translucent_selection_rectangle_border_color

The color of the translucent selection rectangle border. Set to #RRGGBB, RR is the red hex value, GG is the green hex value and BB is the blue hex value. Set to an empty string to use the default color.

size_tiny

size_small

size_medium

size_large

size_huge

The size to use when searching for size:small, size:medium, size:large, size:huge or size:gigantic.

always_keep_sort

Always keep the sort. Set to 1 to enable. Set to 0 to only keep fast sorts. Enabling this will drastically reduce performance.

select_search_on_focus_mode

Select all the search text when the search edit gains focus.

Can be one of the following modes:

Mode	Description
0	Do nothing.
1	Select all text and move the caret to the end.
2	Deselect all text and move the caret to the end.

select_search_on_set_mode

Can be one of the following modes:

Mode	Description
0	Do nothing.
1	Select all text and move the caret to the end.
2	Deselect all text and move the caret to the end.

db_update_thread_priority

The thread priority to update the database.

Can be one of the following values:

Value	Description
-15	Idle priority.
-1	Below normal priority.
0	Normal priority.

1	Above normal priority.
2	High priority.

refs_file_id_extd_directory_info_buffer_size

The read buffer size in bytes for reading ReFS folders. Set to 0 to use the default buffer size (64k).

folder_update_thread_mode_background

Set the folder monitor thread to have low IO priority.

Set to 1 to enable. Set to 0 to disable.

folder_update_rescan_asap

Rescan folders when starting Everything if a scheduled folder rescan was missed.

Set to 1 to rescan immediately. Set to 0 to rescan at the next scheduled time.

monitor_thread_mode_background

Set the USN monitor thread to have low IO priority.

Set to 1 to enable. Set to 0 to disable.

monitor_retry_delay

The retry delay in milliseconds before attempting to reopen an offline volume for monitoring.

monitor_update_delay

The delay in milliseconds between reading USN journal changes.

monitor_pause

Pause folder indexes monitoring, file list changes monitoring and USN change journal monitors.

Set to 1 to pause. Set to 0 to allow monitors to start.

Monitors can be pause and resumed with the search commands: /monitor_pause and /monitor_resume.

usn_record_filter

Include list of USN reasons flags. USN Reasons that are not included in this list are ignored.

Set to 0xffffffff to include all USN reasons.

Can be zero or more of the follow flags in hexadecimal:

Flag	Description
0x00000001	USN_REASON_DATA_OVERWRITE
0x00000002	USN_REASON_DATA_EXTEND
0x00000004	USN_REASON_DATA_TRUNCATION
0x00000010	USN_REASON_NAMED_DATA_OVERWRITE
0x00000020	USN_REASON_NAMED_DATA_EXTEND
0x00000040	USN_REASON_NAMED_DATA_TRUNCATION
0x00000100	USN_REASON_FILE_CREATE
0x00000200	USN_REASON_FILE_DELETE

0x00000400	USN_REASON_EA_CHANGE
0x00000800	USN_REASON_SECURITY_CHANGE
0x00001000	USN_REASON_RENAME_OLD_NAME
0x00002000	USN_REASON_RENAME_NEW_NAME
0x00004000	USN_REASON_INDEXABLE_CHANGE
0x00008000	USN_REASON_BASIC_INFO_CHANGE
0x00010000	USN_REASON_HARD_LINK_CHANGE
0x00020000	USN_REASON_COMPRESSION_CHANGE
0x00040000	USN_REASON_ENCRYPTION_CHANGE
0x00080000	USN_REASON_OBJECT_ID_CHANGE
0x00100000	USN_REASON_REPARSE_POINT_CHANGE
0x00200000	USN_REASON_STREAM_CHANGE
0x80000000	USN_REASON_CLOSE

cancel_delay

The delay in milliseconds to wait before showing the cancel dialog when the Everything database is busy.

http_title_format

The http title format.

See [Change the Everything window caption](#) for more information.

http_server_show_drive_labels

Show drive labels for root folders.

Set to 1 to enable. Set to 0 to disable.

http_server_strings

Customize the built in http server.

See [Custom Strings](#) for more information.

alpha

Set the transparency value of the Everything.

0 = transparent. 128 = 50% transparent. 192 = 25% transparent. 255 = opaque.

snap

Snap the Everything search window to the desktop edges. Set to 1 to enable snapping. Set to 0 to disable snapping.

snaplen

The maximum number of pixels from a desktop edge before the Everything search window will snap to it.

thumbnail_fill

Set to 1 to resize the thumbnail to fill the full thumbnail size. Set to 0 to limit the resizing to 1:1.

min_thumbnail_size

Set the minimum thumbnail size in pixels. Must be greater than 0.

max_thumbnail_size

Set the maximum thumbnail size in pixels. Must be greater than 0. Anything larger than [thumbnail_load_size](#) will be stretched.

medium_thumbnail_size

Set the medium thumbnail size in pixels.

large_thumbnail_size

Set the large thumbnail size in pixels.

extra_large_thumbnail_size

Set the extra large thumbnail size in pixels.

thumbnail_load_size

Set the internal thumbnail size. The largest supported thumbnail size is 1024x1024.
Set to 0 to use the default internal size (256x256).

show_in_taskbar

Set to 1 to show the Everything taskbar button.
Set to 0 to prevent the Everything search window from showing in the taskbar.

toggle_window_from_tray_icon

Set to 1 to toggle the search window from the tray icon. The taskbar must be ontop, there must not be any topmost windows.
Set to 0 to show or create a new search window from the tray icon.

show_focus_on_search

Set to 1 to brings the result list item focus into view after each search or sort.
Set to 0 to allow the result list item focus to go off screen after a search or sort.

statusbar_selected_item_format

The following string substitutions can be used:

String	Substitution
\$ \$	A single \$
\$ f	Full path and file name
\$ n	Name
\$ p	Path
\$ s	Size

\$e	Extension
\$t	Type
\$m	Date modified
\$c	Date created
\$a	Date accessed
\$b	Attributes
\$d	Date recently changed
\$r	Run count
\$u	Date last run
\$l	File list file name

Set to an empty string to use the text from each visible column (showing path last)

minimize_to_tray

When enabled, the Everything Search window will not show up in the task bar when minimized.
The search window can be restored from the tray icon.
Set to 1 to enable. Set to 0 to disable.

auto_remove_moved_ntfs_volumes

Set to 1 to enable.
Set to 0 to disable.
Automatically remove old NTFS volumes that have moved to a new drive letter.

auto_remove_moved_refs_volumes

Set to 1 to enable.
Set to 0 to disable.
Automatically remove old ReFS volumes that have moved to a new drive letter.

find_mount_points_on_removable_volumes

Look for mount points on removable volumes.
Set to 1 to enable.
Set to 0 to disable.
Can caused undesired access to removable volumes if enabled.

find_subfolders_and_files_max_threads

Set to the number of threads to use.
Set to 0 to use the default number of threads. The default number of threads to use is $\text{max_thread} / 2$ (at least one thread is used).

statusbar_size_format

Set the size format for the total selected size in the status bar.
Must be one of the following values:

Value	Description
0	Auto (Bytes/KB/MB/GB depending on size).
1	Always Bytes
2	Always KB.
3	Always MB.

lvm

Enable or disable listview messages. Set to 1 to enable. Set to 0 to disable.

ipc

Enable or disable IPC. Set to 1 to enable. Set to 0 to disable.

rename_nav

Enable or disable rename navigation or inline renaming. Set to 1 to enable. Set to 0 to disable.

tray_show_command_line

The command line to execute when running Everything from the tray icon.

For example, set to -home to load the home settings when clicking on the tray icon.

dpi

The Dots Per Inch (DPI) multiplier. Use to scale Everything.

Set to 0 to disable.

96 = 100%, 120=125%, 144=150%, 168=175%, 192=200%

This setting is multiplied by the current system DPI.

Please note that some widgets always use the system DPI.

ctrl_mouse_wheel_action

The action to perform when using the mouse wheel with the control held down.

Can be one of the following values:

Value	Description
0	None.
1	Change view
2	Change zoom

allow_open

Allow executing results.

Set to 1 to allow. Set to 0 to disallow.

allow_context_menu

Allow right clicking on results.

Set to 1 to allow. Set to 0 to disallow.

allow_delete

Allow deleting results.
Set to 1 to allow. Set to 0 to disallow.

allow_rename
Allow renaming results.
Set to 1 to allow. Set to 0 to disallow.

allow_cut
Allow cutting/moving results.
Set to 1 to allow. Set to 0 to disallow.

allow_copy
Allow copying results.
Set to 1 to allow. Set to 0 to disallow.

allow_paste
Allow pasting results.
Set to 1 to allow. Set to 0 to disallow.

allow_drag_drop
Allow dragging results and allow dropping on results.
Set to 1 to allow. Set to 0 to disallow.

auto_column_widths
Auto adjust column widths when resizing the search window.
Set to 1 to enable. Set to 0 to disable.

hotkey_explorer_path_search
Set the search to the current path in Windows Explorer when opening Everything from a hotkey.
Set to 1 to enable. Set to 0 to disable.

get_key_name_text
Use the keyboard driver for keyboard button names.
Set to 1 to enable. Set to 0 to disable.

paste_new_line_op
Convert new lines when pasting multi-line text into the search box.
Can be one of the following values:

Value	Description
0	AND (convert new lines to spaces)
1	OR (convert new lines to)

always_update_query_on_search_parameter_change
Update the query when search as you type is disabled and when you change a search option or filter.
Set to 1 to enable. Set to 0 to disable.

preview_mag_filter

Set the magnify filter for images in the preview window.
Only works for images.

Must be one of the following values:

Value	Description
0	Use default setting.
3	Color on color. (Performance)
4	Half tone (Quality)

preview_min_filter

Set the mipmap filter for images in the preview window.
Only works for images.

Must be one of the following values:

Value	Description
0	Use default setting.
3	Color on color. (Performance)
4	Half tone (Quality)

preview_fill

Set to 1 to resize the preview image to fill the full preview window. Set to 0 to limit the resizing to 1:1.
Only works for images.

show_preview_handlers_in_preview_pane

Allow system preview handlers in the preview pane. Everything has a built in preview handler for images.
Set to 0 to use the system setting. Set to 1 to allow. Set to 2 to disallow. The system setting can be changed from Windows Explorer -> Tools -> Folder Options... -> View -> Show preview handlers in preview pane.

preview_load_size

The single value used for both width and height in pixels of the image preview load buffer. Set to 0 to use the default of 1024.

service_pipe_name

Name of the service pipe.
Must be in the form of \\.\pipe\pipename
Leave blank to use the default pipe name: \\.\pipe\Everything Service

thumbnail_mouseover_border_color

Set the thumbnail mouse over border color.
Use the following color format: #RRGGBB

preview_background_color

Set the preview background color.
Use the following color format: #RRGGBB

etp_client_rewrite_patterns

etp_client_rewrite_substitutions

A comma (,) separated list of ETP link rewrites.

For example, on the ETP server you may have a folder D:\folder1 which is shared as \\ETPserver\folder1.

Set the pattern to: "D:\\folder1"

set the substitution to: "\\\\ETPserver\\folder1"

icons_only

Always display icons, never thumbnails.

Must be one of the following values:

Value	Description
0	Use system setting.
1	Icons only
2	Thumbnails

The system setting can be changed in Windows Explorer -> Tools -> Folder Options... -> View -> Always show icons, never thumbnails.

auto_scroll_repeat_delay

The delay to wait in milliseconds before auto scrolling the result list.

auto_scroll_repeat_rate

The repeat rate in milliseconds to auto scroll the result list.

open_many_files_warning_threshold

Show the open many files warning dialog when attempting to open more than the specified number of files.

search_edit_drag_accept_files

Allow files and folders to be dropped on the search edit. When a single folder is dropped, Everything will search inside this folder. When multiple files and/or folders are dropped, a file list search is generated to find the specified files and folders.

always_request_all_fileinfo

Always request file information, even when it is not displayed in the result list.

replace_exact_trailing_star_dot_star_with_star

Replace *.* with *

Only works when match_whole_filename_when_using_wildcards is enabled.

When enabled, *.* also matches files that do not contain a .

When disabled, *.* matches only files containing a .

Set to 1 to enable. Set to 0 to disable.

allow_exclamation_point_not

Allow ! to be used as a NOT operator.

Set to 1 to enable. Set to 0 to disable.

search_command_prefix

The search command prefix. If `convert_forward_slash_to_backslash` is enabled, you may wish to change this to `>` or some other prefix.

auto_complete_search_command

Allow TAB to auto complete search commands. The search must currently start with the search command prefix.

Set to 1 to enable. Set to 0 to disable.

listview_tooltips

Show or don't show tooltips in the result list. Set to 1 to show tooltips. Set to 0 to not show tooltips.

show_detailed_listview_tooltips

Show detailed tooltips in thumbnail view.

Set to 1 to show text for all visible column types. Set to 0 to show the name only.

Backup

To make a backup of all settings:

- In **Everything**, type in the following search and press ENTER:

```
/config_save
```

- Choose a filename and destination and click **Save**.

Restore

To restore all settings:

- In **Everything**, type in the following search and press ENTER:

```
/config_load
```

- Select a Everything.ini and click **Open**.

Restore default settings

To restore all the default settings in Everything:

- Exit "**Everything**"
- Delete the Everything.ini file in %APPDATA%\Everything
- Restart Everything.

Custom filename

Use Everything.exe `-config` <Everything.ini> to specify the filename of the Everything ini file.

See also

- [Options](#)

Keyboard Shortcuts

A list of all the default keyboard shortcuts for Everything and how to customize them.

Search edit shortcuts

Keyboard shortcuts that can be used when the search edit has focus.

Shortcut key	Action
Ctrl + A	Select all text.
Ctrl + Backspace	Delete previous word.
Ctrl + Space	Complete search. Requires search history to be enabled.
Enter	Focus result list and select the item with the highest run count.
Up Arrow Down Arrow	Focus result list.
Alt + Up Arrow Alt + Down Arrow	Show search history.
Double Click	Select word
Triple Click	Select all

Result list shortcuts

Keyboard shortcuts that can be used when the result list has focus.

Shortcut key	Action
F2	Rename the focused item.
Delete	Move selected items to the recycle bin.
Shift + Delete	Permanently delete selected items.
Left Arrow	Scroll left.
Right Arrow	Scroll right.
Ctrl + Left Arrow	Scroll left one page.
Ctrl + Right Arrow	Scroll right one page.
Enter	Open the selected items.
Ctrl + Enter	Open the path of the selected item.

Alt + Enter	Display properties of the selected items.
Ctrl + +	auto adjust column widths.
Up Arrow Down Arrow Page Up Page Down Home End	Navigate item focus and selection.
Shift + Up Arrow Shift + Down Arrow Shift + Page Up Shift + Page Down Shift + Home Shift + End	Extend item selection.
Ctrl + Up Arrow Ctrl + Down Arrow Ctrl + Page Up Ctrl + Page Down Ctrl + Home Ctrl + End	Navigate item focus.
Ctrl + Shift + Up Arrow Ctrl + Shift + Down Arrow Ctrl + Shift + Page Up Ctrl + Shift + Page Down Ctrl + Shift + Home Ctrl + Shift + End	Extend item selection, keeping the current selection.
Space	Select focus.
Ctrl + Space	Toggle item selection.
Escape	Cancel any current actions.
Ctrl + A	Selected all items.
Ctrl + C Ctrl + Insert	Copy selected items.
Ctrl + E	Read the file size, date and attributes of the current selection. May be useful before exporting all sizes, dates and attributes.
Ctrl + V Shift + Insert	Paste items from the clipboard into the selected folder.
Ctrl + X	Cut selected items.
Ctrl + Shift + C	Copy the full path and name of the current selection to the clipboard.
Shift + F10	Display item context menu for the current selection. Displays the result list context menu if no selection.
Any character	Adds a character to the jump to text.
Backspace	Remove the last character in the jump to text.

Global keyboard shortcuts

Keyboard shortcuts that can be used when an Everything search window has focus.

Shortcut key	Action
Escape Ctrl + W	Close the Everything window.
F1	Show the Everything help.
F3 Ctrl + F Alt + D	Focus and highlight the search edit.
F5	Reload icons, file sizes, dates and attributes.
F11	Toggle fullscreen.
Ctrl + Tab Ctrl + Shift + Tab	Cycle between open windows.
Tab Shift + Tab	Cycle between search edit and results view.
Escape	Close the current window.
Alt + 1	Resize the window to 512 x 398.
Alt + 2	Resize the window to 640 x 497.
Alt + 3	Resize the window to 768 x 597.
Alt + 4	Auto size the window.
Alt + P	Toggle the preview pane.
Ctrl + Shift + 1	Toggle extra large thumbnails.
Ctrl + Shift + 2	Toggle large thumbnails.
Ctrl + Shift + 3	Toggle medium thumbnails.
Ctrl + Shift + 6	Toggle detail view.
Ctrl + Alt + +	Increase thumbnail size.
Ctrl + Alt + -	Decrease thumbnail size.
Ctrl + `	Toggle debug console.
Ctrl + F1	Show About Everything.
Ctrl + 1	Sort by name.
Ctrl + 2	Sort by path.
Ctrl + 3	Sort by size.
Ctrl + 4	Sort by extension.
Ctrl + 5	Sort by type.
Ctrl + 6	Sort by date modified.

Ctrl + 7	Sort by date created.
Ctrl + 8	Sort by attributes.
Ctrl + 9	Sort by date recently changed.
Ctrl + B	Toggle match whole word.
Ctrl + D	Bookmark the current search.
Ctrl + I	Toggle match case.
Ctrl + M	Toggle match diacritics.
Ctrl + N	Open a new search window.
Ctrl + O	Open an Everything file list.
Ctrl + P	Show the Options window.
Ctrl + Q	Exit Everything.
Ctrl + R	Toggle Regex.
Ctrl + S	Export the current results to an Everything file list, csv or txt file.
Ctrl + T	Toggle always on top.
Ctrl + U	Toggle match path.
Ctrl + +	Increase text size.
Ctrl + -	Decrease text size.
Ctrl + 0	Reset text size to Normal.
Ctrl + Mouse Wheel Up Ctrl + Mouse Wheel Down	Change view.
Alt + Home	Go to the home search.
Alt + Left Arrow Back Mouse back button	Go back to the previous search.
Alt + Right Arrow Forward Mouse forward button	Go forward to the next search.
Ctrl + Shift + F	Organize filters.
Ctrl + Shift + B	Organize bookmarks.
Ctrl + H Ctrl + Shift + H	Show all search history.

Hotkeys

The Everything search window can be shown by pressing a hotkey.
Hotkeys can be used anywhere in Windows.

There are three different methods for opening the Everything window with a hotkey:

Method	Description
New window	Always show a new Everything search window.
Show window	Show the Everything search window if one exists, otherwise create a new Everything search window.
Toggle window	Hide the Everything search window if it is currently in the foreground, otherwise show the Everything search window if one exists, otherwise create a new Everything search window.

To define a Hotkey to open Everything:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Keyboard** tab.
- Choose a hotkey method.
- Press a new key combination.
- Click OK.

Note: Everything must be running in the background for hotkeys to work.

Note: You can not override any existing hotkey.

To disable a Windows hotkey, such as Win + F:

- From the Start menu, open regedit
- In the Registry Editor, navigate to:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
```

- Create a new String Value called **DisabledHotkeys**
- Set the DisabledHotkeys data to the letter you would like freed, for example: **F**
- Restart your computer
- Apply the (now free) hotkey Win + F in Everything.

Customizing

To customize keyboard shortcuts:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Keyboard** tab.
- Use **Show commands containing:** to search for a command
- Click **Add...** to add a new keyboard shortcut or click **Edit...** to edit an existing one.
- Select a location to use the shortcut in:

Location	Description
Global	The shortcut will work anywhere in the Everything search window.
Search Edit	The shortcut will only work when pressed in the search edit.
Result List	The shortcut will only work when pressed in the result list.

- Press a new keyboard shortcut for shortcut key.
- Check shortcut key currently used by, this existing shortcut (if any) will be removed if you click OK.
- Click **OK**.
- Click **OK**.

KNOWNFOLDERID

Use with the `shell:<name>` search function.

Replace `<name>` with one of the following physical folders:

Use double quotes to escape spaces in the shell name, eg: `shell:"Common Start Menu"`

Name	Availability	Example Path
AccountPictures	Windows 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Windows\AccountPictures
Administrative Tools	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Administrative Tools
AppData	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming
Application Shortcuts	Windows 8, 10	C:\Users\User\AppData\Local\Microsoft\Windows\Application Shortcuts
Cache	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\XXXXXXXXX
Camera Roll	Windows 8, 10	C:\Users\User\Pictures\Camera Roll
CD Burning	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Local\Microsoft\Windows\Burn\Burn
Common Administrative Tools	Windows XP, Vista, 7, 8, 10	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools
Common AppData	Windows XP, Vista, 7, 8, 10	C:\ProgramData
Common Desktop	Windows XP, Vista, 7, 8, 10	C:\Users\Public\Desktop
Common Documents	Windows XP, Vista, 7, 8, 10	C:\Users\Public\Documents
Common Programs	Windows XP, Vista, 7, 8, 10	C:\ProgramData\Microsoft\Windows\Start Menu\Programs
Common Start Menu	Windows XP, Vista, 7, 8, 10	C:\ProgramData\Microsoft\Windows\Start Menu
Common Start Menu Places	Windows 10	C:\ProgramData\Microsoft\Windows\Start Menu Places
Common Startup	Windows XP, Vista, 7, 8, 10	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup
Common Templates	Windows XP, Vista, 7, 8, 10	C:\ProgramData\Microsoft\Windows\Templates
CommonDownloads	Windows Vista, 7, 8, 10	C:\Users\Public\Downloads

CommonMusic	Windows Vista, 7, 8, 10	C:\Users\Public\Music
CommonPictures	Windows XP, Vista, 7, 8, 10	C:\Users\Public\Pictures
CommonRingtones	Windows 7, 8, 10	C:\ProgramData\Microsoft\Windows\Ringtones
CommonVideo	Windows XP, Vista, 7, 8, 10	C:\Users\Public\Videos
Contacts	Windows Vista, 7, 8, 10	C:\Users\User\Contacts
Cookies	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Windows\Cookies
CredentialManager	Windows Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Credentials
CryptoKeys	Windows Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Crypto
Default Gadgets	Windows Vista, 7	C:\Program Files (x86)\Windows Sidebar\Gadgets
Desktop	Windows XP, Vista, 7, 8, 10	C:\Users\User\Desktop
Device Metadata Store	Windows 7, 8, 10	C:\ProgramData\Microsoft\Windows\DeviceMetadataStore
DocumentsLibrary	Windows 7, 8, 10	Libraries\Documents
Downloads	Windows Vista, 7, 8, 10	C:\Users\User\Downloads
DpapiKeys	Windows Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Protect
Favorites	Windows XP, Vista, 7, 8, 10	C:\Users\User\Favorites
Fonts	Windows XP, Vista, 7, 8, 10	C:\Windows\Fonts
Gadgets	Windows Vista, 7	C:\Users\User\AppData\Local\Microsoft\Windows Sidebar\Gadgets
GameTasks	Windows Vista, 7, 8, 10	C:\Users\User\AppData\Local\Microsoft\Windows\GameExplorer
History	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Local\Microsoft\Windows\History
ImplicitAppShortcuts	Windows 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned\ImplicitAppShortcuts
Links	Windows Vista, 7, 8, 10	C:\Users\User\Links

Local AppData	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Local
Local Documents	Windows 10	C:\Users\User\Documents
Local Downloads	Windows 10	C:\Users\User\Downloads
Local Music	Windows 10	C:\Users\User\Music
Local Pictures	Windows 10	C:\Users\User\Pictures
Local Videos	Windows 10	C:\Users\User\Videos
LocalAppDataLow	Windows Vista, 7, 8, 10	C:\Users\User\AppData\LocalLow
LocalizedResourcesDir	Windows XP, Vista	C:\Windows\Resources\<LanguageID>
My Music	Windows XP, Vista, 7, 8, 10	C:\Users\User\Music
My Pictures	Windows XP, Vista, 7, 8, 10	C:\Users\User\Pictures
My Video	Windows XP, Vista, 7, 8, 10	C:\Users\User\Videos
NetHood	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft \Windows\Network Shortcuts
OneDrive	Windows 8, 10	C:\Users\User\OneDrive
Personal	Windows XP, Vista, 7, 8, 10	C:\Users\User\Documents
PrintHood	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft \Windows\Printer Shortcuts
Profile	Windows XP, Vista, 7, 8, 10	C:\Users\User
ProgramFiles	Windows XP, Vista, 7, 8, 10	C:\Program Files
ProgramFilesCommon	Windows Vista, 7, 8, 10	C:\Program Files\Common Files
ProgramFilesCommonX64	Windows 10	C:\Program Files\Common Files
ProgramFilesCommonX86	Windows Vista, 7, 8, 10	C:\Program Files (x86)\Common Files
ProgramFilesX64	Windows 10	C:\Program Files
ProgramFilesX86	Windows Vista, 7, 8, 10	C:\Program Files (x86)
Programs	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft \Windows\Start Menu\Programs
Public	Windows Vista, 7, 8, 10	C:\Users\Public

PublicAccountPictures	Windows 8, 10	C:\Users\Public\AccountPictures
PublicGameTasks	Windows Vista, 7, 8, 10	C:\ProgramData\Microsoft\Windows\GameExplorer
PublicLibraries	Windows 8, 10	C:\Users\Public\Libraries
Quick Launch	Windows Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch
Recent	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Windows\Recent
ResourceDir	Windows XP, Vista, 7, 8, 10	C:\Windows\Resources
Ringtones	Windows 7, 8, 10	C:\Users\User\AppData\Local\Microsoft\Windows\Ringtones
Roaming Tiles	Windows 8, 10	C:\Users\User\AppData\Local\Microsoft\Windows\RoamingTiles
SampleMusic	Windows Vista, 7	C:\Users\Public\Music\Sample Music
SamplePictures	Windows Vista, 7	C:\Users\Public\Pictures\Sample Pictures
SampleVideos	Windows Vista, 7	C:\Users\Public\Videos\Sample Videos
SavedGames	Windows Vista, 7, 8, 10	C:\Users\User\Saved Games
SavedPictures	Windows 10	C:\Users\User\Pictures\Saved Pictures
Searches	Windows Vista, 7, 8, 10	C:\Users\User\Searches
SendTo	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Windows\SendTo
SkyDriveCameraRoll	Windows 8	C:\Users\User\SkyDrive\Pictures\Camera Roll
SkyDriveDocuments	Windows 8	C:\Users\User\SkyDrive\Documents
SkyDriveMusic	Windows 8	C:\Users\User\SkyDrive\Music
SkyDrivePictures	Windows 8	C:\Users\User\SkyDrive\Pictures
Start Menu	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Windows\Start Menu
Startup	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
System	Windows XP, Vista, 7, 8, 10	C:\Windows\System32
SystemCertificates	Windows Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\SystemCertificates

SystemX86	Windows XP, Vista, 7, 8, 10	C:\Windows\SysWOW64
Templates	Windows XP, Vista, 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Windows\Templates
ThisDeviceFolder	Windows 8, 10	C:\Users\Public
User Pinned	Windows 7, 8, 10	C:\Users\User\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned
UserProfiles	Windows Vista, 7, 8, 10	C:\Users
UserProgramFiles	Windows 7, 8, 10	C:\Users\User\AppData\Local\Programs
UserProgramFilesCommon	Windows 7, 8, 10	C:\Users\User\AppData\Local\Programs\Common
UsersFilesFolder	Windows Vista, 7, 8, 10	C:\Users\User
Windows	Windows XP, Vista, 7, 8, 10	C:\Windows

Language IDs

A list of Windows language Identifiers

Hexadecimal code	Decimal code	Language
0x0000	0	Language Neutral
0x007f	127	The language for the invariant locale (LOCALE_INVARIANT). See MAKELCID.
0x0400	1024	Process or User Default Language
0x0800	2048	System Default Language
0x0436	1078	Afrikaans
0x041c	1052	Albanian
0x0401	1025	Arabic (Saudi Arabia)
0x0801	2049	Arabic (Iraq)
0x0c01	3073	Arabic (Egypt)
0x1001	4097	Arabic (Libya)
0x1401	5121	Arabic (Algeria)
0x1801	6145	Arabic (Morocco)
0x1c01	7169	Arabic (Tunisia)
0x2001	8193	Arabic (Oman)

0x2401	9217	Arabic (Yemen)
0x2801	10241	Arabic (Syria)
0x2c01	11265	Arabic (Jordan)
0x3001	12289	Arabic (Lebanon)
0x3401	13313	Arabic (Kuwait)
0x3801	14337	Arabic (U.A.E.)
0x3c01	15361	Arabic (Bahrain)
0x4001	16385	Arabic (Qatar)
0x042b	1067	Windows 2000/XP: Armenian. This is Unicode only.
0x042c	1068	Azeri (Latin)
0x082c	2092	Azeri (Cyrillic)
0x042d	1069	Basque
0x0423	1059	Belarusian
0x0445	1093	Bengali (India)
0x141a	5146	Bosnian (Bosnia and Herzegovina)
0x0402	1026	Bulgarian
0x0455	1109	Burmese
0x0403	1027	Catalan
0x0404	1028	Chinese (Taiwan)
0x0804	2052	Chinese (PRC)
0x0c04	3076	Chinese (Hong Kong SAR, PRC)
0x1004	4100	Chinese (Singapore)
0x1404	5124	Windows 98/Me, Windows 2000/XP: Chinese (Macao SAR)
0x041a	1050	Croatian
0x101a	4122	Croatian (Bosnia and Herzegovina)
0x0405	1029	Czech
0x0406	1030	Danish
0x0465	1125	Windows XP: Divehi. This is Unicode only.
0x0413	1043	Dutch (Netherlands)
0x0813	2067	Dutch (Belgium)
0x0409	1033	English (United States)
0x0809	2057	English (United Kingdom)

0x0c09	3081	English (Australian)
0x1009	4105	English (Canadian)
0x1409	5129	English (New Zealand)
0x1809	6153	English (Ireland)
0x1c09	7177	English (South Africa)
0x2009	8201	English (Jamaica)
0x2409	9225	English (Caribbean)
0x2809	10249	English (Belize)
0x2c09	11273	English (Trinidad)
0x3009	12297	Windows 98/Me, Windows 2000/XP: English (Zimbabwe)
0x3409	13321	Windows 98/Me, Windows 2000/XP: English (Philippines)
0x0425	1061	Estonian
0x0438	1080	Faeroese
0x0429	1065	Farsi
0x040b	1035	Finnish
0x040c	1036	French (Standard)
0x080c	2060	French (Belgian)
0x0c0c	3084	French (Canadian)
0x100c	4108	French (Switzerland)
0x140c	5132	French (Luxembourg)
0x180c	6156	Windows 98/Me, Windows 2000/XP: French (Monaco)
0x0456	1110	Windows XP: Galician
0x0437	1079	Windows 2000/XP: Georgian. This is Unicode only.
0x0407	1031	German (Standard)
0x0807	2055	German (Switzerland)
0x0c07	3079	German (Austria)
0x1007	4103	German (Luxembourg)
0x1407	5127	German (Liechtenstein)
0x0408	1032	Greek
0x0447	1095	Windows XP: Gujarati. This is Unicode only.
0x040d	1037	Hebrew
0x0439	1081	Windows 2000/XP: Hindi. This is Unicode only.

0x040e	1038	Hungarian
0x040f	1039	Icelandic
0x0421	1057	Indonesian
0x0434	1076	isiXhosa/Xhosa (South Africa)
0x0435	1077	isiZulu/Zulu (South Africa)
0x0410	1040	Italian (Standard)
0x0810	2064	Italian (Switzerland)
0x0411	1041	Japanese
0x044b	1099	Windows XP: Kannada. This is Unicode only.
0x0457	1111	Windows 2000/XP: Konkani. This is Unicode only.
0x0412	1042	Korean
0x0812	2066	Windows 95, Windows NT 4.0 only: Korean (Johab)
0x0440	1088	Windows XP: Kyrgyz.
0x0426	1062	Latvian
0x0427	1063	Lithuanian
0x0827	2087	Windows 98 only: Lithuanian (Classic)
0x042f	1071	Macedonian (FYROM)
0x043e	1086	Malay (Malaysian)
0x083e	2110	Malay (Brunei Darussalam)
0x044c	1100	Malayalam (India)
0x0481	1153	Maori (New Zealand)
0x043a	1082	Maltese (Malta)
0x044e	1102	Windows 2000/XP: Marathi. This is Unicode only.
0x0450	1104	Windows XP: Mongolian
0x0414	1044	Norwegian (Bokmal)
0x0814	2068	Norwegian (Nynorsk)
0x0415	1045	Polish
0x0416	1046	Portuguese (Brazil)
0x0816	2070	Portuguese (Portugal)
0x0446	1094	Windows XP: Punjabi. This is Unicode only.
0x046b	1131	Quechua (Bolivia)
0x086b	2155	Quechua (Ecuador)

0x0c6b	3179	Quechua (Peru)
0x0418	1048	Romanian
0x0419	1049	Russian
0x044f	1103	Windows 2000/XP: Sanskrit. This is Unicode only.
0x043b	1083	Sami, Northern (Norway)
0x083b	2107	Sami, Northern (Sweden)
0x0c3b	3131	Sami, Northern (Finland)
0x103b	4155	Sami, Lule (Norway)
0x143b	5179	Sami, Lule (Sweden)
0x183b	6203	Sami, Southern (Norway)
0x1c3b	7227	Sami, Southern (Sweden)
0x203b	8251	Sami, Skolt (Finland)
0x243b	9275	Sami, Inari (Finland)
0x0c1a	3098	Serbian (Cyrillic)
0x1c1a	7194	Serbian (Cyrillic, Bosnia, and Herzegovina)
0x081a	2074	Serbian (Latin)
0x181a	6170	Serbian (Latin, Bosnia, and Herzegovina)
0x046c	1132	Sesotho sa Leboa/Northern Sotho (South Africa)
0x0432	1074	Setswana/Tswana (South Africa)
0x041b	1051	Slovak
0x0424	1060	Slovenian
0x040a	1034	Spanish (Spain, Traditional Sort)
0x080a	2058	Spanish (Mexican)
0x0c0a	3082	Spanish (Spain, Modern Sort)
0x100a	4106	Spanish (Guatemala)
0x140a	5130	Spanish (Costa Rica)
0x180a	6154	Spanish (Panama)
0x1c0a	7178	Spanish (Dominican Republic)
0x200a	8202	Spanish (Venezuela)
0x240a	9226	Spanish (Colombia)
0x280a	10250	Spanish (Peru)
0x2c0a	11274	Spanish (Argentina)

0x300a	12298	Spanish (Ecuador)
0x340a	13322	Spanish (Chile)
0x380a	14346	Spanish (Uruguay)
0x3c0a	15370	Spanish (Paraguay)
0x400a	16394	Spanish (Bolivia)
0x440a	17418	Spanish (El Salvador)
0x480a	18442	Spanish (Honduras)
0x4c0a	19466	Spanish (Nicaragua)
0x500a	20490	Spanish (Puerto Rico)
0x0430	1072	Sutu
0x0441	1089	Swahili (Kenya)
0x041d	1053	Swedish
0x081d	2077	Swedish (Finland)
0x045a	1114	Windows XP: Syriac. This is Unicode only.
0x0449	1097	Windows 2000/XP: Tamil. This is Unicode only.
0x0444	1092	Tatar (Tatarstan)
0x044a	1098	Windows XP: Telugu. This is Unicode only.
0x041e	1054	Thai
0x041f	1055	Turkish
0x0422	1058	Ukrainian
0x0420	1056	Windows 98/Me, Windows 2000/XP: Urdu (Pakistan)
0x0820	2080	Urdu (India)
0x0443	1091	Uzbek (Latin)
0x0843	2115	Uzbek (Cyrillic)
0x042a	1066	Windows 98/Me, Windows NT 4.0 and later: Vietnamese
0x0452	1106	Welsh (United Kingdom)

makevoidtools

makevoidtools is a program to compile your own voidtools.com website.

All text can be customized for localization.

makevoidtools Guide

- Extract [makevoidtools.zip](#) to a new folder, eg: c:\makevoidtools
- Copy the en-us.txt file and name it with the [language code](#) for your language and region.
- This is the main translation template file for the website.
- For example: en-us.txt
- Copy the en-us subfolder and name it with the [language code](#) for your language and region.
- This is the main translation folder for the website.
- For example: en-us
- open your main translation file, eg: en-us.txt
- Read the notes inside this translation file for help.
- translate all strings.
- translate all pages in the main translation folder, eg: en-us*.txt
- I recommend leaving the filenames in English (Optional).

Requirements

Only the files directly in your main translation folder (eg: en-us) need to be translated.

Translating the support wiki is not required. -not all pages need translating as some are quite technical.

translating the following support wiki pages are encouraged:

- 01-Installing Everything.txt
- 02-Using Everything.txt
- Searching.txt

Offline help

The Everything.chm offline help will use the same support wiki templates.

Testing

Run makevoidtools.exe

makevoidtools will compile *.txt in the same location as makevoidtools.exe

These txt files will be translation templates for each language and region.

```
makevoidtools.exe <output folder> <url root>
```

Example:

```
makevoidtools.exe c:\makevoidtools\voidtools file:///c:/makevoidtools/voidtools
```

Open c:\makevoidtools\voidtools\index.html

Wiki Pages

Make sure you open and save wiki pages as UTF-8

Wiki pages in the support folder use basic [wiki markup language](#).

Prefixing a wiki page template with a number are shown above all other wiki pages in sequence. For example, 01-Installing Everything.txt is the first wiki page listed and 02-Using Everything.txt is the second wiki page listed and so on.

wiki page templates starting with a underscore (_) are hidden. For example: <http://www.voidtools.com/support>

[/everything/db.](#)

Wiki syntax

The voidtools wiki syntax is very similar to [Wikipedia's markup](#) with some limitations:

Text

```
A paragraph of text.
```

A paragraph of text.

Headers

```
=Main Header=
```

Main Header

```
==Sub Header==
```

Sub Header

Italic

```
' 'Italic'
```

Italic

Bold

```
'''Bold'''
```

bold

Italic and Bold

```
''''Italic and Bold''''
```

Italic and Bold

Links

Relative URLs are relative to <http://www.voidtools.com/en-us>
where en-us is your language-region code.

The [text part in [[url|text]] can be omitted, in which case the url is used as the text and URLs paths are striped when no protocol is present.

[[http://en.wikipedia.org/wiki/]]

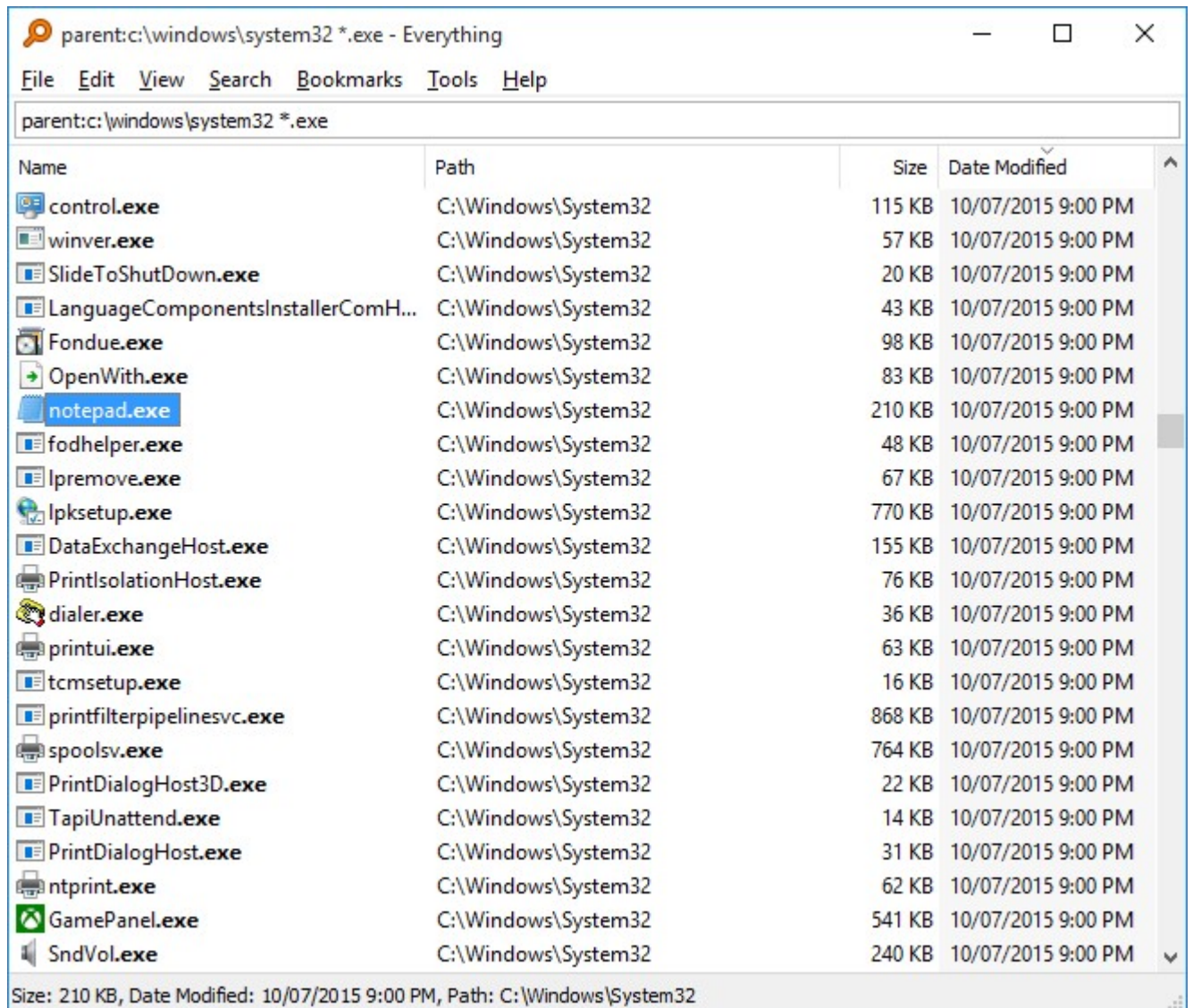
<http://en.wikipedia.org/wiki/>

[[faq#what_is_everything|What is Everything?]]

What is Everything?

Images

[[File:support/everything/Everything.Search.Window.png|alt=Everything Search Window]]



Preformatted text

```
<pre>
int i;
for (i=0;i<10;i++)
{
    printf("hello %d\n",i);
}
```

```
</pre>
```

```
int i;
for (i=0;i<10;i++)
{
    printf("hello %d\n",i);
}
```

Unordered lists

```
<ul>
*item1
*item2
<ul>
*subitem1
*subitem2
*subitem3
</ul>
*item3
</ul>
```

- item1
- item2
 - subitem1
 - subitem2
 - subitem3
- item3

Tables

```
{|
! header1
! header2
! header3
|-
| row1, data1
| row1, data2
| row1, data3
|-
| row2, abc1
| row2, abc2
| row2, abc3
|}
```

header1	header2	header3
row1, data1	row1, data2	row1, data3
row2, abc1	row2, abc2	row2, abc3

Discription lists

```
<dl>
;'item1''
text1
;'item2''
```

```
text2
;''item3''
text3
</dl>
```

item1
text1

item2
text2

item3
text3

Macros

```
{ { DOWNLOAD_EVERYTHING } }
```

Download Everything 1.4.1.1002

[Download Installer](#)[Download Installer 64-bit](#)[Download Portable Zip](#)[Download Portable Zip 64-bit](#)[Download Lite Installer](#)[Download Lite Installer 64-bit](#)

[Changes](#) ▪ [What's New](#) ▪ [Older Versions](#) ▪ [License](#) ▪ [SHA256](#) ▪ [Supported Languages](#) ▪ [Help](#)

Help

If you need any assistance, please [contact David Carpenter](#).

Multiple Instances

By default there is only one instance of Everything.

When you run Everything.exe it will check for an existing instance. If an existing instance is found the command line options are sent and processed by the existing instance. If no existing instance is found a new one is created.

To run multiple instances of Everything use [Named instances](#).

Named instances

By default Everything uses the unnamed instance.

Assign unique names to Everything instances to use multiple instances.

There can only be one instance for each unique name.

To assign a name to the instance:

Use the [-instance](#) command line option to specify the instance name when starting Everything.exe.

For example:

```
Everything.exe -instance "ETP Server"
```

Named instances use unique settings and data.

For example, the named instance "ETP Server" will use Everything-ETP SERVER.ini as the settings file.

The name of the instance is located in the Everything search window caption and in the tooltip when hovering over the tray icon with the cursor.

Service

By default Everything will install a service for each instance.

To use the same service for all instances, use the [-service-pipe-name](#) command line option or [service_pipe_name](#) ini option to specify the same service pipe name.

The default unnamed service pipe name is:

```
\\\\.\\PIPE\\Everything Service
```

For example, to connect to the default service pipe with a different instance name:

```
Everything.exe -instance "ETP Server" -service-pipe-name "\\\\.\\PIPE\\Everything Se
```

Options

The Everything Options contains settings that modify the system or Everything.

To open the Everything Options window:

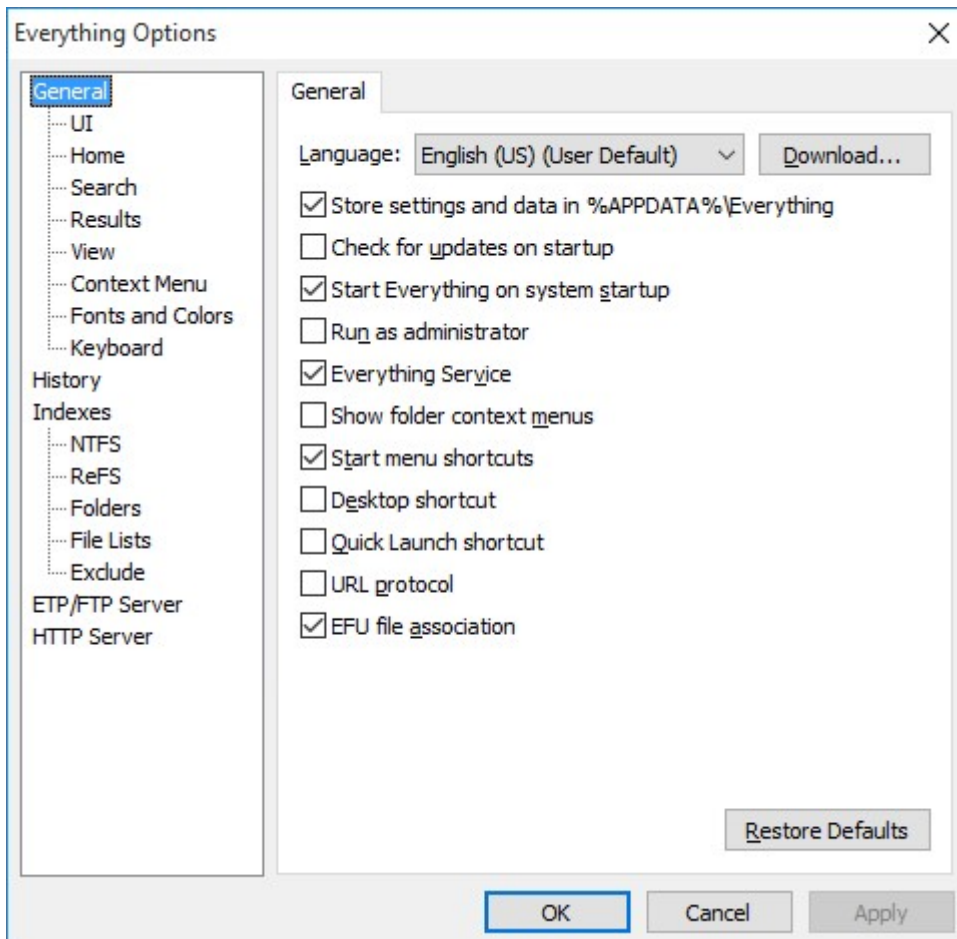
- In **Everything**, from the **Tools** menu, click **Options**.

Select one of the *"tabs"* below:

- [General](#)
 - [UI](#)
 - [Home](#)
 - [Search](#)
 - [Results](#)
 - [View](#)
 - [Context Menu](#)
 - [Fonts and Colors](#)
 - [Keyboard](#)
- [History](#)
- [Indexes](#)
 - [NTFS](#)

- ReFS
- Folders
- File Lists
- Exclude
- ETP/FTP Server
- HTTP Server

General



Contains settings for configuring the installation of *Everything*.

Language

Selects the language used for displaying text in Everything. If set to User Default the language is determined by the current users language for displaying text, (User Default) is displayed next to this language in the language selection combo.

Click the download button to the right of languages to open the download page for the Everything language pack in your web browser.

Store settings and data in %APPDATA%\Everything

Choose the location for storing settings and data. If enabled, settings and data are stored in %APPDATA%\Everything. If disabled, settings and data are stored in the same location as the Everything.exe.

This option is enabled in the installer by default and disabled in the portable version by default. Changing this setting will overwrite any existing settings.

Check for updates on startup

Check www.voidtools.com for Everything updates when starting Everything.
To include beta updates, see [Beta updates](#).

Start Everything on system startup

Start Everything when Windows starts. When Everything starts from Windows startup, the tray icon is shown and the database is loaded and updated, no Everything search window is shown.

Run as administrator

If enabled and Everything is run with limited privileges, Everything will relaunch with administrative privileges. The Everything service is not required when enabling this option.
When disabling this option you must manually restart Everything.
When enabled, applications run from Everything will also run as administrator.
Mapped network drives will also need to be created for the administrator account.

Everything service

Install the Everything service. The Everything service will allow the Everything search window to read and monitor the USN Journal on all NTFS volumes. Enabling this will allow Everything to run as a standard user.

Show folder context menus

Add the **Search Everything...** item to folder context menus. The folder context menu is displayed when you right click a folder in Windows. Clicking on the folder context menu item **Search Everything...** will launch Everything and populate the search with the selected folder.

Start menu shortcuts

Add **Search Everything** shortcut (and **Uninstall Everything** shortcut if uninstall.exe is present) to the start menu under Programs \ Everything.

Desktop shortcut

Add **Search Everything** shortcut to the Desktop.

Quick launch shortcut

Add **Search Everything** shortcut to the Quick launch toolbar.
To show the Quick Launch toolbar in Windows Vista or later:

- Right click an empty spot on the taskbar.
- From the **Toolbars** submenu, click **New toolbar...**
- type in:

`%userprofile%\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch`

- Click Select Folder.

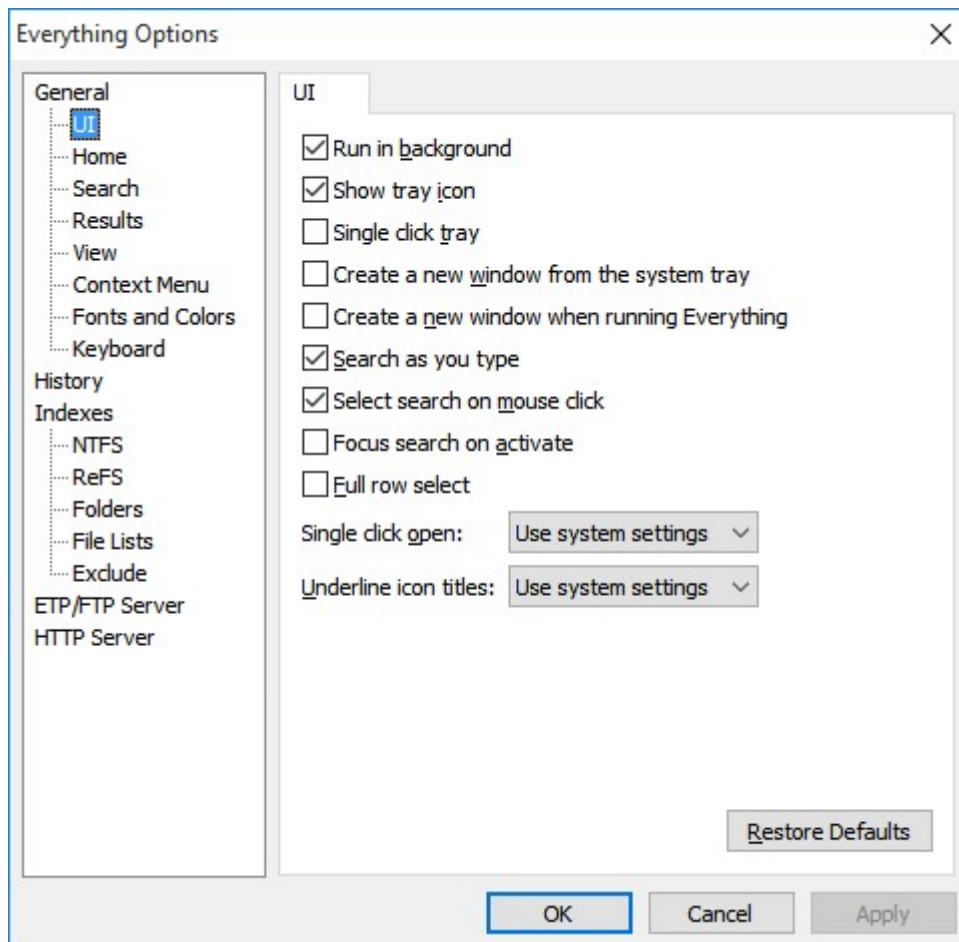
URL protocol

Open Everything from es: urls.
For example, when clicking on [es:abc123](#) Everything will launch and populate the search with abc123.

EFU file association

Open EFU files with Everything. EFU files are Everything file lists, for more information please see [File Lists](#).

UI



Contains settings for configuring the user interface.

Run in background

Keep Everything running in the background when all search windows are closed. If disabled and all search windows are closed, Everything will exit completely.

Show tray icon

Show the Everything tray icon.

Single click tray

If enabled, Everything will open from the tray icon with a single mouse click. If disabled, Everything will open from the tray icon with a double click.

Create a new window from the system tray

If enabled, always create a new search window from the system tray. If disabled, opening the search window from the tray will show an existing search window before creating a new search window.

Create a new window when running Everything

If enabled, always create a new search window when running Everything.exe or an Everything shortcut. If disabled, running Everything.exe or an Everything shortcut will show an existing search window before creating a new search window.

Search as you type

If enabled, each keystroke you press will update the search results. If disabled, you must press enter to update the search results.

Select search on mouse click

Select all the search edit text when the search edit gains focus from a mouse click.

Focus search on activate

Focus the search edit, when Everything is activated from alt + tab or clicking on an Everything search window with the mouse.

Full row select

The whole row is selectable, otherwise only the name part is selectable.

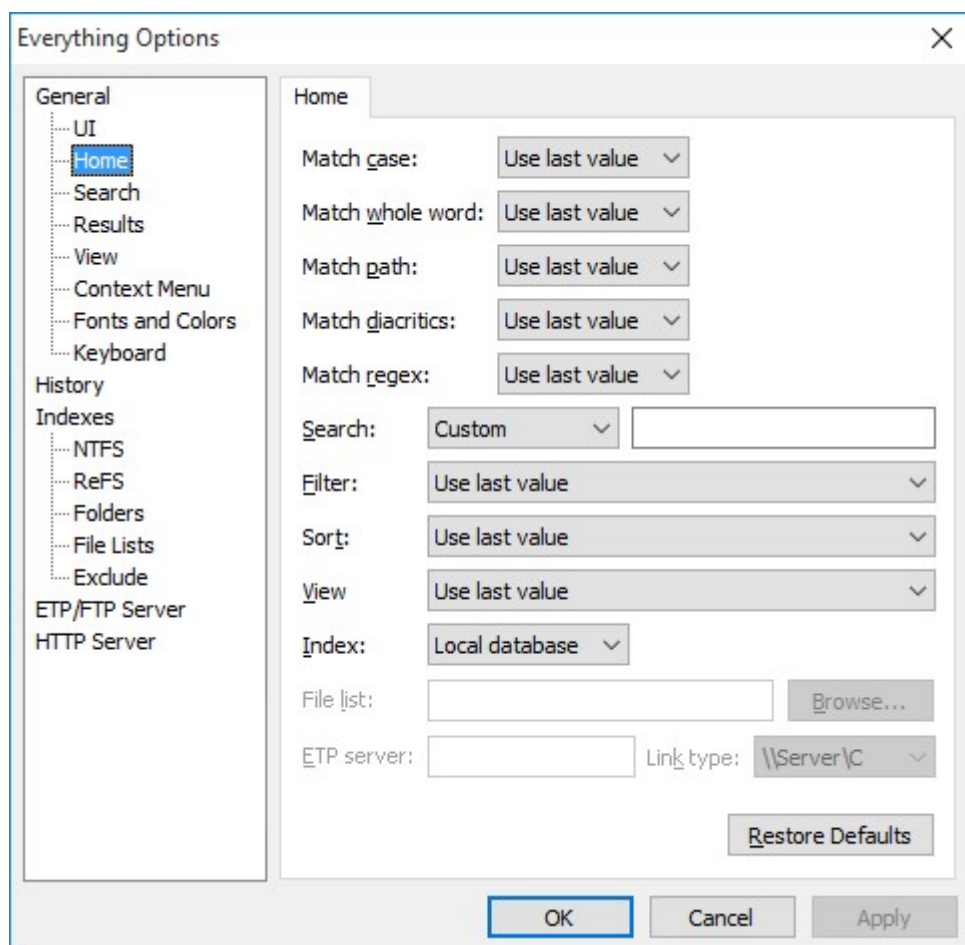
Single click open

Open a file or folder with a single click or double click. To select a file or folder when single click is active hover over the item with the cursor.

Underline icon titles

Choose when filenames are underlined in single click mode.

Home



Contains settings for configuring the first search when a new Everything search window is shown.

Match case

The default match case setting. If set to use last value, the match case setting will be restored from the

previously closed Everything search window.

Match whole word

The default match whole word setting. If set to use last value, the match whole word setting will be restored from the previously closed Everything search window.

Match path

The default match path setting. If set to use last value, the match path setting will be restored from the previously closed Everything search window.

Match diacritics

The default match diacritics setting. If set to use last value, the match diacritics setting will be restored from the previously closed Everything search window.

Match Regex

The default match regex setting. If set to use last value, the match regex setting will be restored from the previously closed Everything search window.

Search

The default search. If set to use last value, the search will be restored from the previously closed Everything search window. Set to custom to define your own default search.

Filter

The default filter. If set to use last value, the filter will be restored from the previously closed Everything search window.

Sort

The default sort. If set to use last value, the sort will be restored from the previously closed Everything search window if it was a fast sort. A fast sort is a sort by name, run count, date run or date recent changed.

To always keep the sort, see [Always keep sort](#).

Index

The default index. If set to use last value, the index will be restored from the previously closed Everything search window.

File list

The filename of the file list when the index is set to a file list.

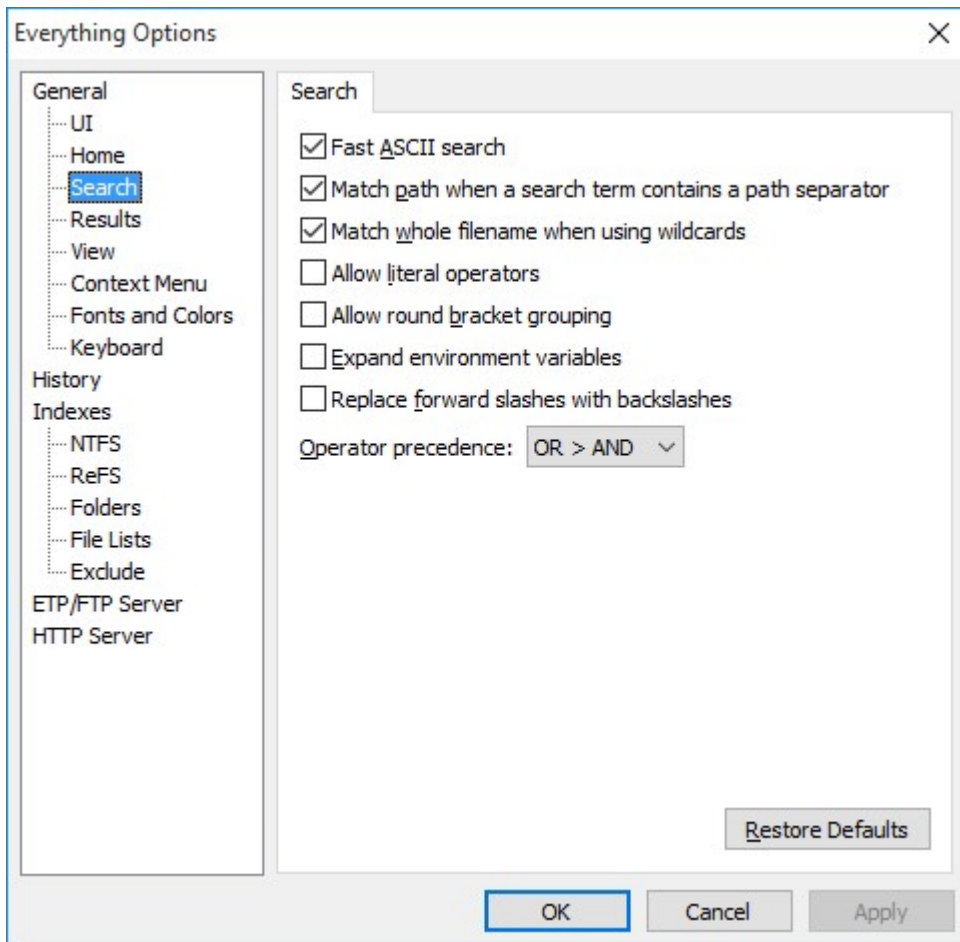
ETP server

The username, password, host and port of the ETP server when the index is set to a ETP server.

Link type

The link type of the ETP server when the index is set to an ETP server.

Search



Contains settings for configuring the search.

Fast ASCII search

Use single byte comparisons for lower case ASCII characters.

Enabling this will greatly improve searching performance when match diacritics is enabled.

Disable this to allow Turkish I with dot to match lowercase i.

Match path when a search term contains a path separator

Match the full path and filename when a search term contains a backslash (\).

Disable this to find files with backslashes in the filename.

Match whole filename when using wildcards

If enabled, using wildcards must match the *entire* filename.

If disabled, wildcard matches can be found anywhere in the filename.

Allow literal operators

Allow AND, OR and NOT as operators.

Can cause issues when searching for filenames that include the string AND, OR or NOT.

Allow round bracket grouping

Allow (and) for grouping operators. Can cause issues when searching for filenames with the text (and).

Expand environment variables

Expand environment variables inside % quotes.

For example: %APPDATA% will expand to C:\Users\<Current User>\AppData\Roaming.

Can cause issues when searching for filenames that include a % character.

Replace forward slashes with backslashes

Forward slashes (/) are replaced with backslashes (\).

Can cause issues when searching for filenames that include a / character. File lists can contain URLs with forward slashes (/).

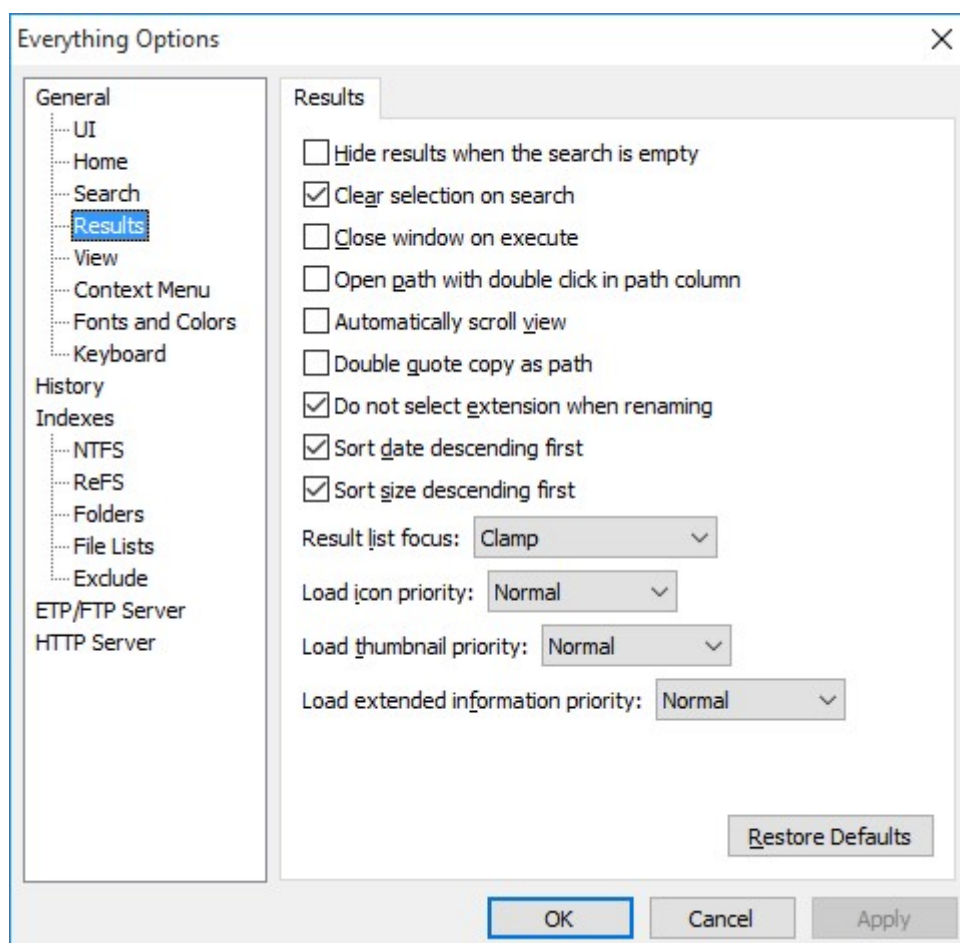
Operator precedence

Set the order in which operators are evaluated.

Set to OR > AND to evaluate OR operations first, for example, searching for A B | C D is the same as A (B | C) D.

Set to AND > OR to evaluate AND operations first, for example, searching for A B | C D is the same as (A B) | (C D).

Results



Contains settings for configuring the result list.

Hide results when the search is empty

The results are not shown until you type in a search.

Clear selection on search

Clear the selection each time the search changes. Disable to preserve the selection between searches.

Close window on execute

The Everything search window is closed immediately after opening an item.

Open path with double click

Paths in the path column can be opened with a double click.

Auto scroll view

Periodically scroll the result list when the mouse is dragged outside the visible area.

Double quote copy as path

Include double quotes when copying filenames to the clipboard.

Do not select extension when renaming

Enable to select the file part only, excluding the extension when renaming a file.

Sort date descending first

When sorting by date, display newest dates first.

Sort size descending first

When sorting by size, display the largest files first.

Result list focus

Can be one of the following:

Value	Description
Clamp	The result list focus will be clamped to the top and bottom items.
Repeat	The result list focus will wrap around the top and bottom items.
Repeat with search	The result list focus will wrap around the top and bottom items to the search edit.

Load icon priority

The thread priority for loading icons.

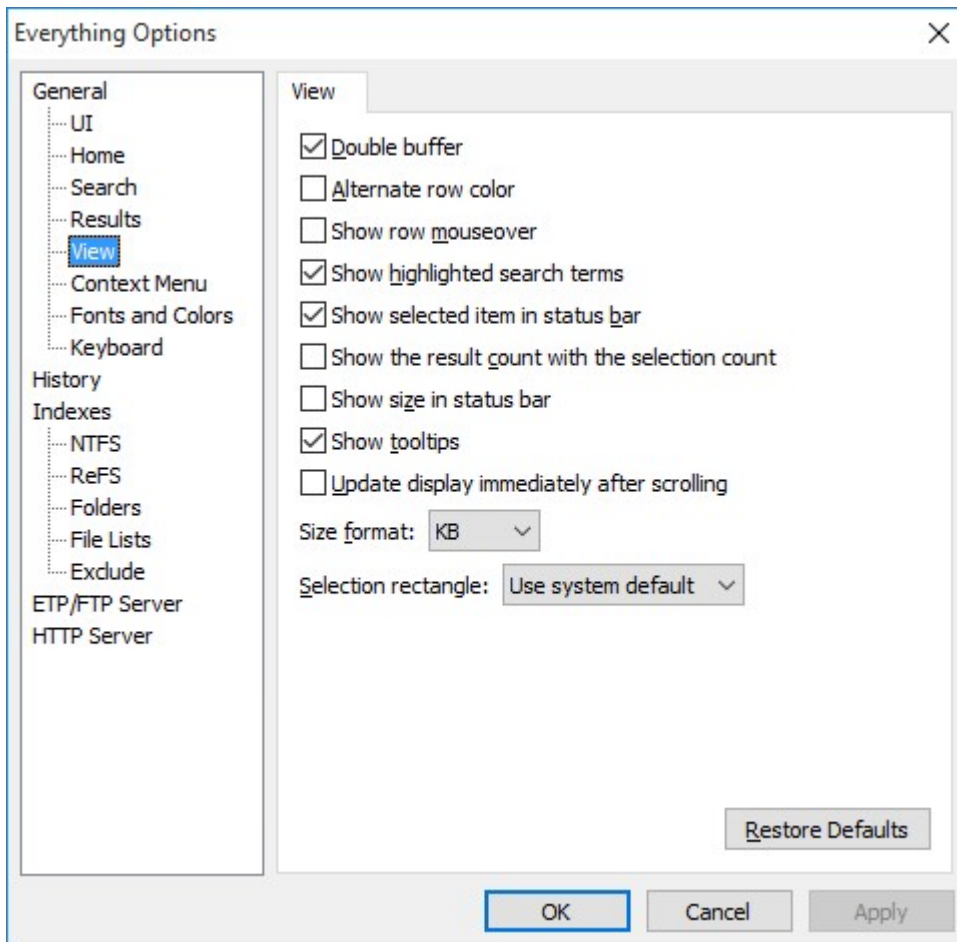
Load thumbnail priority

The thread priority for loading thumbnails.

Load extended information priority

The thread priority for loading file dates, sizes and attributes.

View



Contains settings for configuring the view.

Double buffer

Use an off screen buffer when rendering the result list. Enable to avoid flickering when rendering. Disable to draw directly to the screen.

Alternate row color

Show odd and even rows in a different color.

Show row mouseover

Show the row the mouse is currently over in a different color.

Show highlighted search terms

Show highlighted search terms.

Show selected item in status bar

Show the full path and filename of the selected file in the status bar.

The status bar format for the selected item can be customized with the [statusbar_selected_item_format](#) ini option.

Show the result count with the selection count

Show the total number of results with the selection count in the status bar.

Show size in status bar

Shows the total file selection size in the status bar.

Folder are not included.

If there is no selection the total file size is shown.

Show tooltips

Show helpful tooltips in the Everything Options window.

Update display immediately after scrolling

Force the display to update immediately after it is scrolled. Enabling this can cause performance issues for mice with high refresh rates.

Size format

The size format. Can be one of the following values:

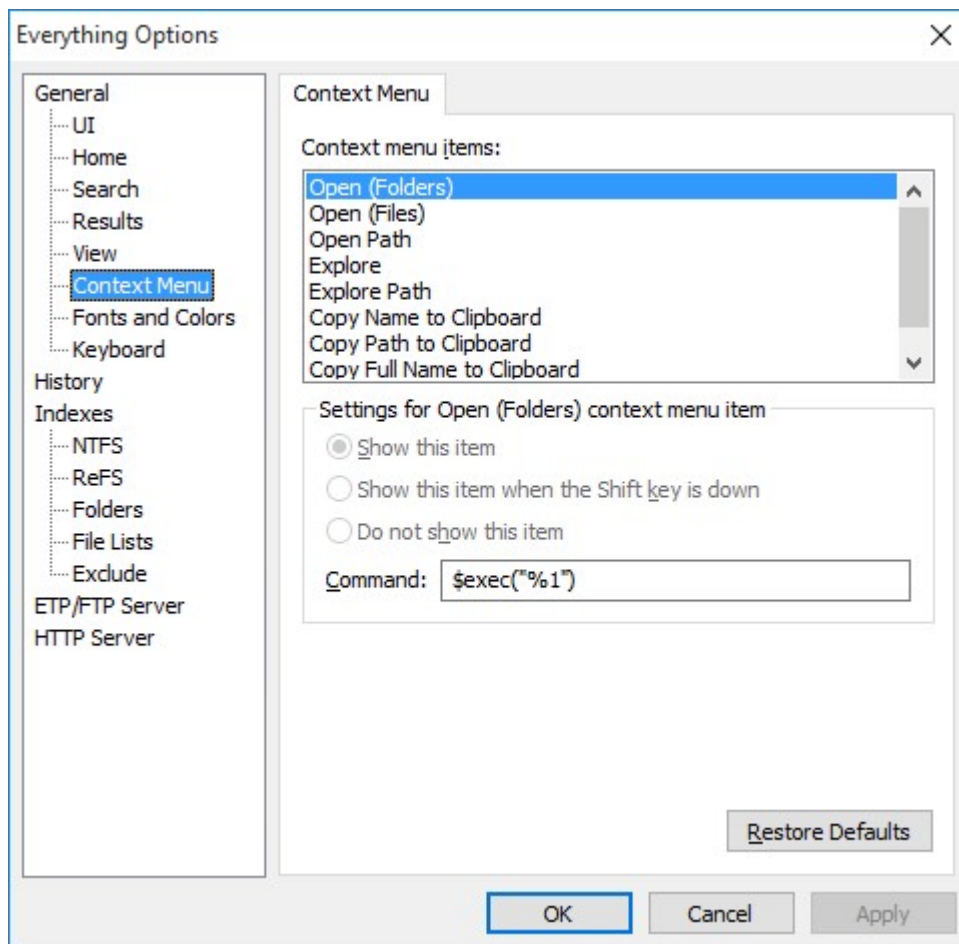
Value	Description
Auto	Use the best format depending on the size.
Bytes	Always display sizes in bytes.
KB	Always display sizes in KB.

Selection rectangle

The selection rectangle display type. Can be one of the following values:

Value	Description
Use system default.	Use the same setting as Windows Explorer.
Translucent mask	Show a transparent selection mask. Translucency and color can be customized .
XOR mask	Display every other pixel as an inverted pixel.

Context Menu



Contains settings for configuring the context menu. The context menu is displayed when right clicking a file or folder in the Everything result list.

Show this item

Show this item when the Shift key is down

Do not show this item

Choose the visibility of this context menu item.

Command

The command for the selected context menu item.

The following command syntax can be used:

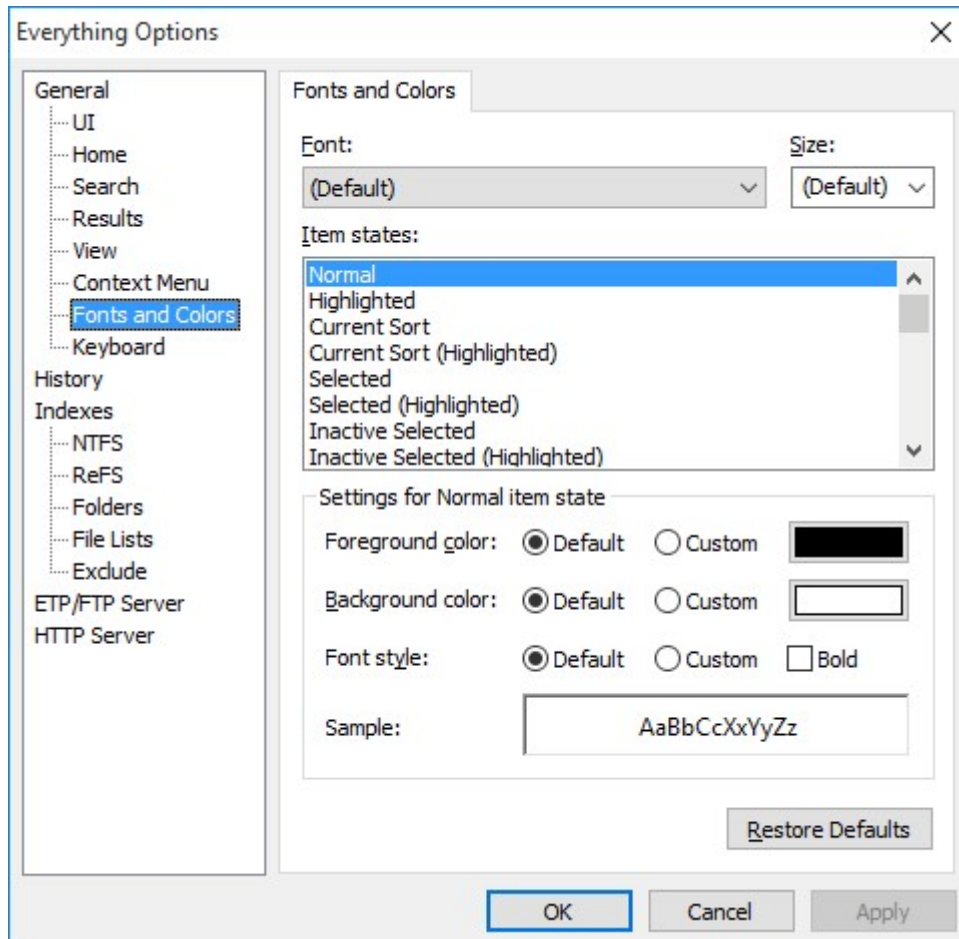
command	Description
\$exec(filename)	Executes the specified filename.
\$openpath(filename)	Opens the folder containing the specified filename and selects the specified filename.
\$parent(filename)	The parent folder of the specified filename
\$pathpart(filename)	The path part of filename
\$namepart(filename)	The name and extension part of filename
%1	The selected filename.
\$\$	A single literal \$
\$(A single literal (

\$)

A single literal)

See [Customizing](#) for more information.

Fonts and Colors



Contains settings for configuring the font and colors in *Everything*.

Font

Size

The result list font family and size. Use (Default) to use the same settings as the Windows icon font. To customize other fonts, please see [Everything.ini](#).

Foreground color

The text color for the selected item. Select a custom color by clicking on the color button.

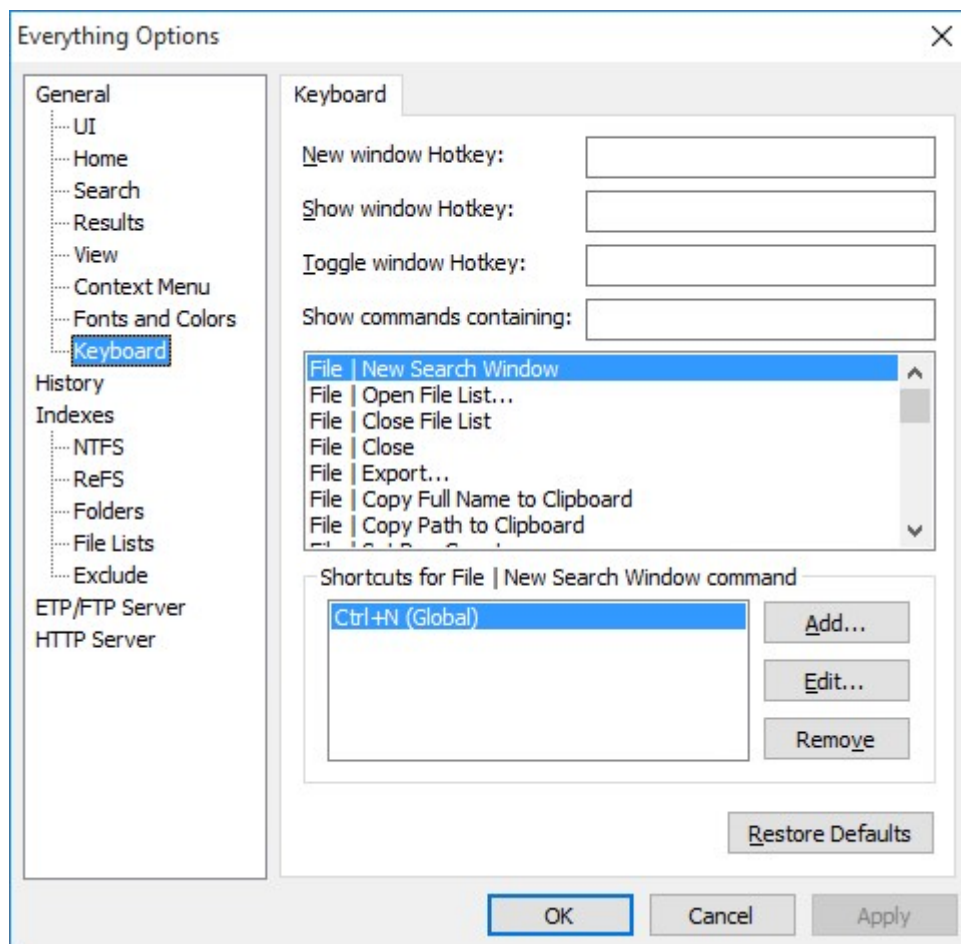
Background color

The background color for the selected item. Select a custom color by clicking on the color button.

Font style

The font style. Can be bold or normal.

Keyboard



Contains settings for configuring keyboard shortcuts and hotkeys.

For a list of all the default keyboard shortcuts, please see [Keyboard Shortcuts](#).

New window hotkey

Show window hotkey

Toggle window hotkey

New window will always show a new Everything search window. Show window will show an existing window if one exists, otherwise a new search window is created. Toggle window will hide any existing window that is in the foreground, otherwise an existing window is shown if one exists, otherwise a new search window is created.

Press a new keyboard shortcut to set the hotkey.

Show commands containing

Type in a search to filter the command list by name.

Shortcuts for the selected command

To add a new keyboard shortcut to the selected command or edit an existing keyboard shortcut for the selected command:

- Click **Add...** or **Edit...**
- Choose a location to use the shortcut. Can be one of the following values:

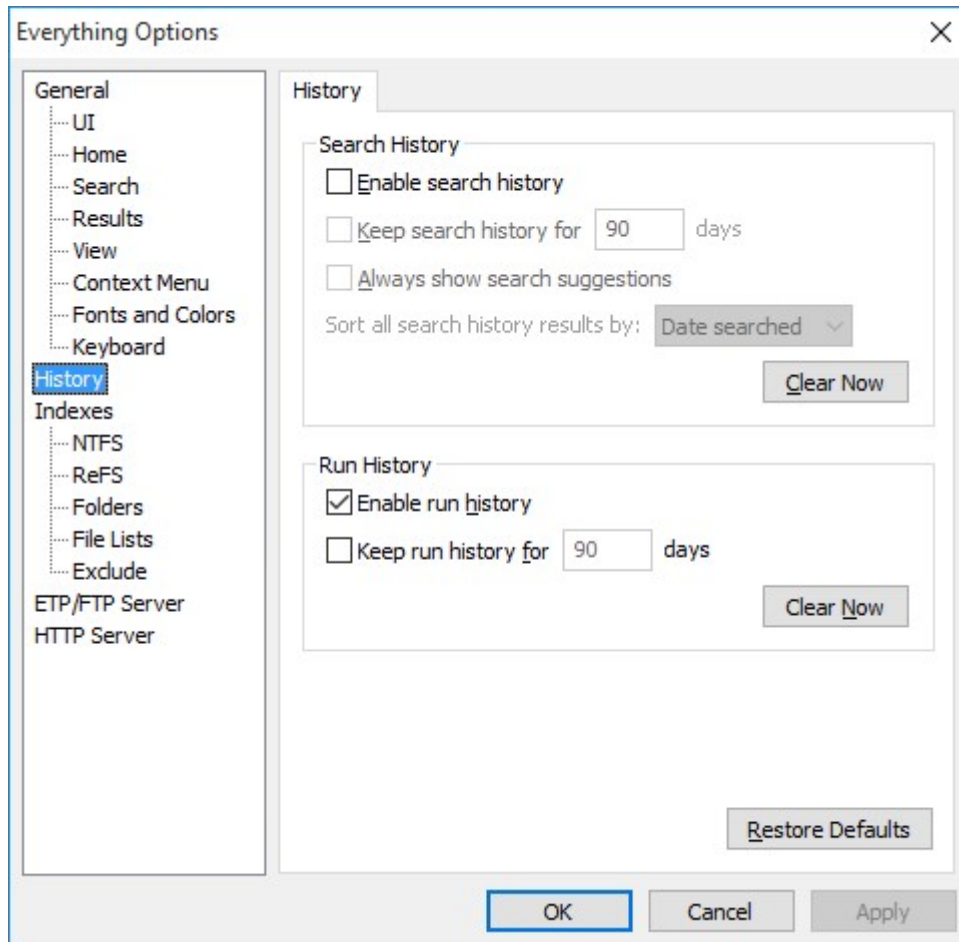
Value	Description
Global	The shortcut can be used anywhere in <i>Everything</i> .
Search Edit	The shortcut can only be used in the search edit.
Result List	The shortcut can only be used in the result list.

- Press a new shortcut key for shortcut key.
- Take note of the shortcut key currently used by list, These existing keyboard shortcuts will be removed when clicking OK.
- Click **OK**.

To remove a keyboard shortcut:

- Select the keyboard shortcut in shortcuts for the selected command.
- Click **Remove**.

History



Contains settings for configuring search history and run history.

Enable search history

Remember search history. A search is added to the search history when you press enter from the search edit or when you act on a file or folder in the result list. Disabling search history does not clear the search history. Click Clear Now to permanently delete search history.

Press Ctrl + Space to complete a search using the search history, if there's more than one way to complete the search a suggestion list is shown.

Press the drop down arrow on the right to show search suggestions.

See [Search History](#) for more information.

Keep search history

Specify the number of days to keep search history. When the search history is loaded, searches that occurred longer than the specified number of days will be removed.

if disabled search history is kept forever.

Always show search suggestions

Show the suggestion list after every change to the search.

Consider disabling [Search as you type](#).

See [Everything.ini](#) for more options.

Enable run history

Remember the number of times each file or folder has been run. When you execute an item in the result list, its run count is incremented by one. You can set the run count for an item in the result list by right clicking it and selecting Set run count.... Disabling run history will not delete run history. Click Clear Now to permanently delete run history.

Press enter from the search edit will focus the result with the highest run count.

Results can be sorted by run count.

See [runcount:](#) and [daterun:](#) to search run history.

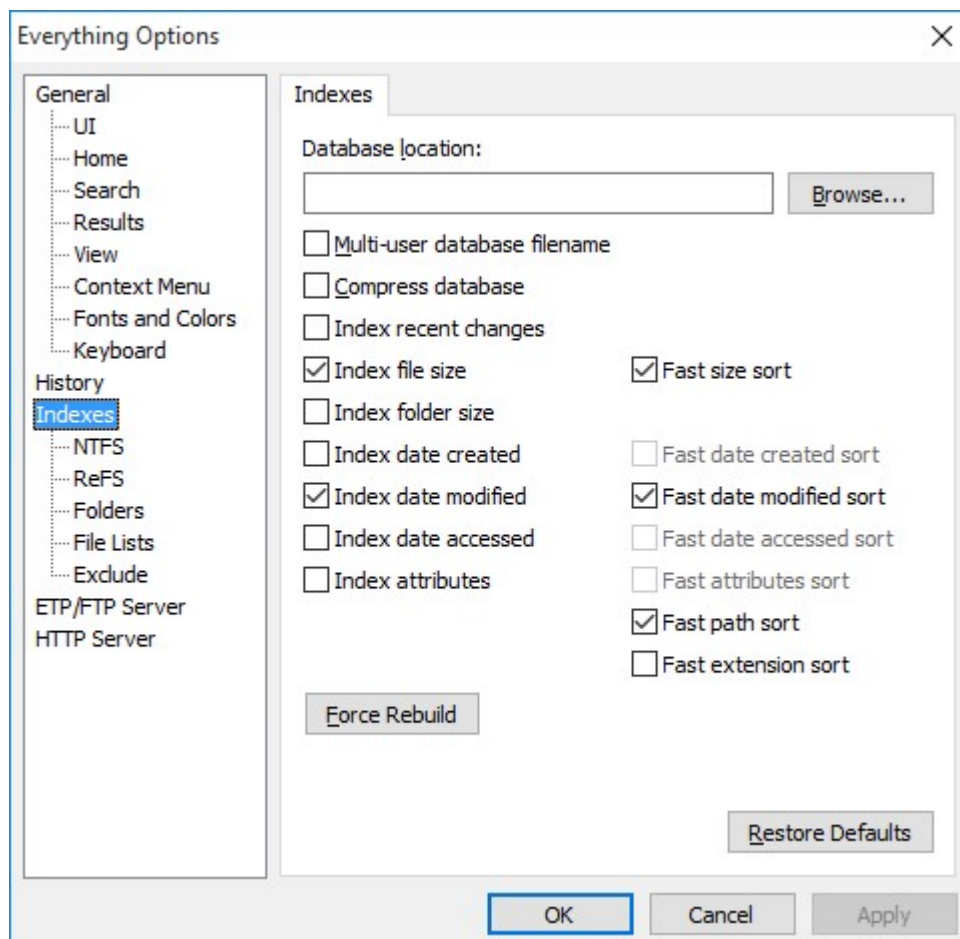
See [Run History](#) for more information.

Keep run history

Specify the number of days to keep run history. When the run history is loaded, searches that occurred longer than the specified number of days will be removed.

if disabled run history is kept forever.

Indexes



Contains settings for configuring the indexes.

Database location

Override the default location for the database. The default location is:

%LOCALAPPDATA%\Everything

If [Store settings and data in %APPDATA%\Everything](#) is disabled, the database location is the same as the Everything.exe location.

Click **Browse...** to browse for a database location.

Multi-user database filename

Normally Everything uses the filename Everything.db for the database. If enabled Everything will use the unique filename Everything.<ComputerName>.<UserName>.db. This might be useful when using the portable version of Everything on a thumb drive on different computers.

Compress database

Compress the database with Bzip 2. Can improve loading performance on very slow hard drives at the cost of higher CPU usage when loading and saving.

Index recent changes

Maintain a separate recent change sorted database of all file system changes that can be searched with [rc](#) or sorted by date recently changed.

Index file size

- Include file size information in the index.
- Size information is maintained in real time.
- Enable to search for size information instantly.
- Greatly improves sorting by size performance.
- Requires an additional 8 bytes of memory per file.

fast size sort

- Maintain a separate size sorted database of files.
- Enable to sort by size instantly.
- Enable to make Everything remember this sort between searches.
- Requires an additional 4-8 bytes of memory per file.

Index folder size

- Include folder size information in the index.
- Size information is maintained in real time.
- Enable to search for folder size information instantly.
- Requires an additional 8 bytes of memory per folder.
- Folder size will not include excluded files.

Index file date created

- Include date created information in the index.
- Date created information is maintained in real time.
- Enable to search for date created information instantly.
- Greatly improves sorting by date created performance.
- Requires an additional 8 bytes of memory per file and folder.

fast date created sort

- Maintain a separate date created sorted database of files and folders.
- Enable to sort by date created instantly.
- Enable to make Everything remember this sort between searches.
- Requires an additional 4-8 bytes of memory per file.

Index file date modified

- Include date modified information in the index.
- Date modified information is maintained in real time.
- Enable to search for date modified information instantly.
- Greatly improves sorting by date modified performance.
- Requires an additional 8 bytes of memory per file and folder.

fast date modified sort

- Maintain a separate date modified sorted database of files and folders.
- Enable to sort by date modified instantly.
- Enable to make Everything remember this sort between searches.
- Requires an additional 4-8 bytes of memory per file.

Index file date accessed

- Include date accessed information in the index.
- Date accessed information is maintained in real time.
- Enable to search for date accessed information instantly.
- Greatly improves sorting by date accessed performance.
- Requires an additional 8 bytes of memory per file and folder.

fast date accessed sort

- Maintain a separate date accessed sorted database of files and folders.
- Enable to sort by date accessed instantly.
- Enable to make Everything remember this sort between searches.
- Requires an additional 4-8 bytes of memory per file.

Index file attributes

- Include attributes information in the index.
- Attributes information is maintained in real time.
- Enable to search for attributes information instantly.
- Greatly improves sorting by attributes performance.
- Requires an additional 4 bytes of memory per file and folder.

fast attributes sort

- Maintain a separate attributes sorted database of files and folders.
- Enable to sort by attributes instantly.
- Enable to make Everything remember this sort between searches.
- Requires an additional 4-8 bytes of memory per file.

fast path sort

- Maintain a separate path sorted database of files and folders.
- Enable to sort by path instantly.
- Enable to make Everything remember this sort between searches.
- Requires an additional 4-8 bytes of memory per file.

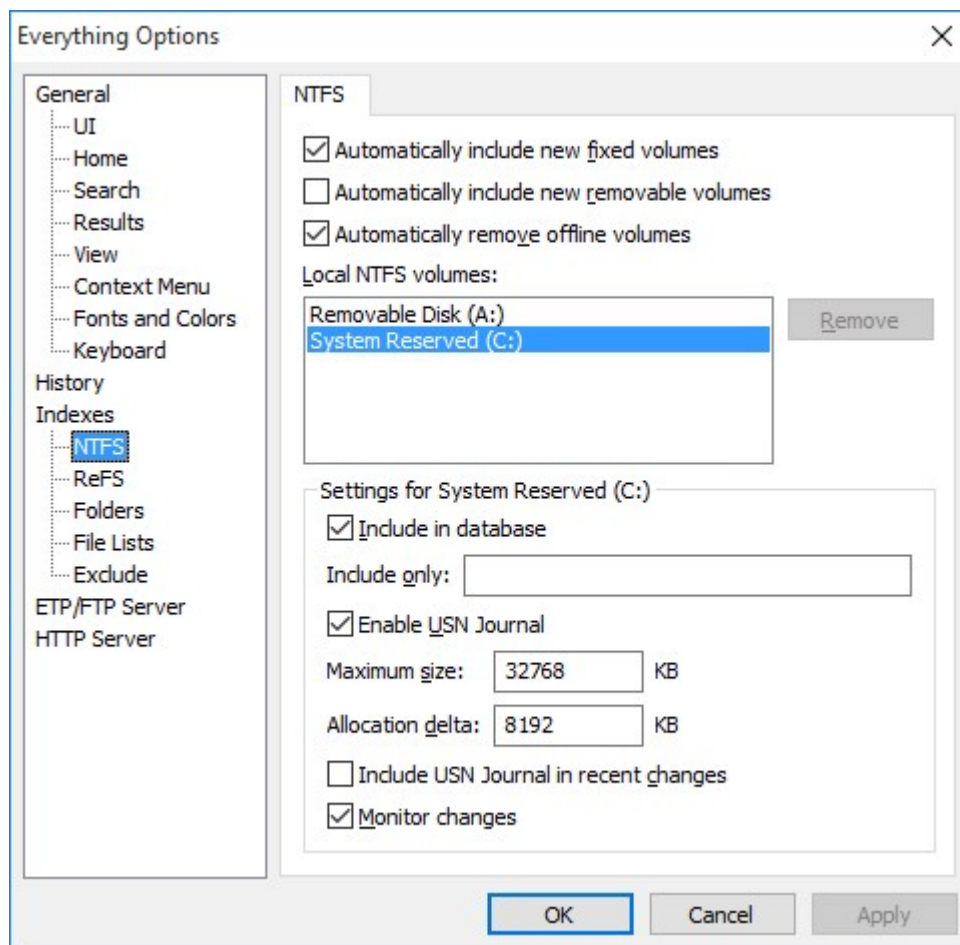
fast extension sort

- Maintain a separate extension sorted database of files and folders.
- Enable to sort by extension instantly.
- Enable to make Everything remember this sort between searches.
- Requires an additional 4-8 bytes of memory per file.

Force Rebuild

- Rebuild the Everything.db database.

NTFS



Contains settings for configuring NTFS volumes and NTFS indexing.

Auto include new fixed volumes

- Automatically include new fixed volumes in Everything.

Auto include new removable volumes

- Automatically include new removable volumes in Everything.
- Everything must rebuild the database when a volume path, volume guid, the USN Journal ID changes or the volume goes offline.

Enable USN Journal

- Enable USN Journal logging on the selected NTFS volume. Everything uses the USN Change journal to index

and monitor changes to NTFS volumes.

Maximum size

Set the maximum size of the USN Journal in KB. If Everything is rebuilding the database after restarting Windows, try increasing this size to at least 32768 KB.

If the journal grows larger than this value it is truncated by removing the specified allocation delta from the beginning of the log.

Allocation delta

Specify how much is removed from the start of the USN Journal in bytes when it has reached its maximum size.

Include in database

Include this NTFS volume in the Everything index.

Include USN Journal in recent changes

Load the entire USN Journal for this volume into the recent changes list. See [Recent changes](#) for more information.

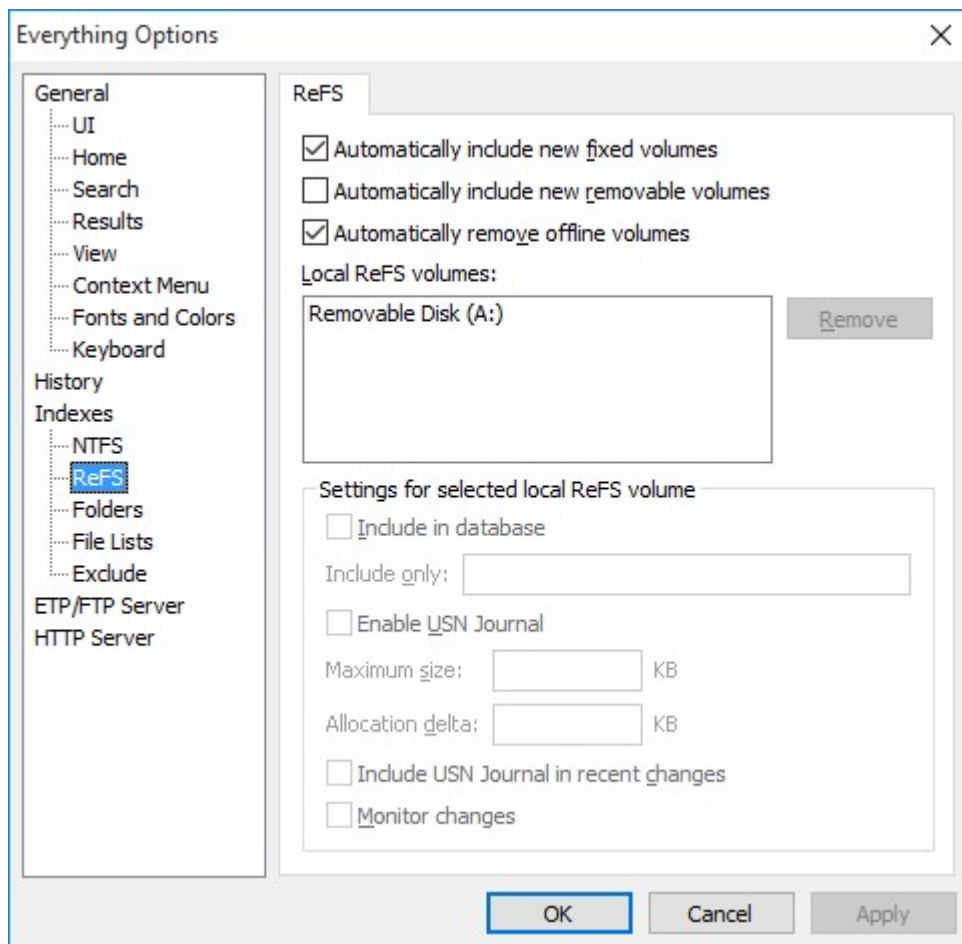
Include only

semicolon delimited list of folders to include on this NTFS volume. Leave empty to include all folders.

Monitor changes

Keep the NTFS index of this NTFS volume up to date.

ReFS



Contains settings for configuring ReFS volumes and ReFS indexing.

Auto include new fixed volumes

Automatically include new fixed volumes in Everything.

Auto include new removable volumes

Automatically include new removable volumes in Everything.

Everything must rebuild the database when a volume path, volume guid, the USN Journal ID changes or the volume goes offline.

Enable USN Journal

Enable USN Journal logging on the selected ReFS volume. Everything uses the USN Change journal to index and monitor changes to ReFS volumes.

Maximum size

Set the maximum size of the USN Journal in KB. If Everything is rebuilding the database after restarting Windows, try increasing this size to at least 32768 KB.

If the journal grows larger than this value it is truncated by removing the specified allocation delta from the beginning of the log.

Allocation delta

Specify how much is removed from the start of the USN Journal in bytes when it has reached its maximum size. Allocation delta is currently ignored for ReFS volumes.

Include in database

Include this ReFS volume in the Everything index.

Include USN Journal in recent changes

Load the entire USN Journal for this volume into the recent changes list. See [Recent changes](#) for more information.

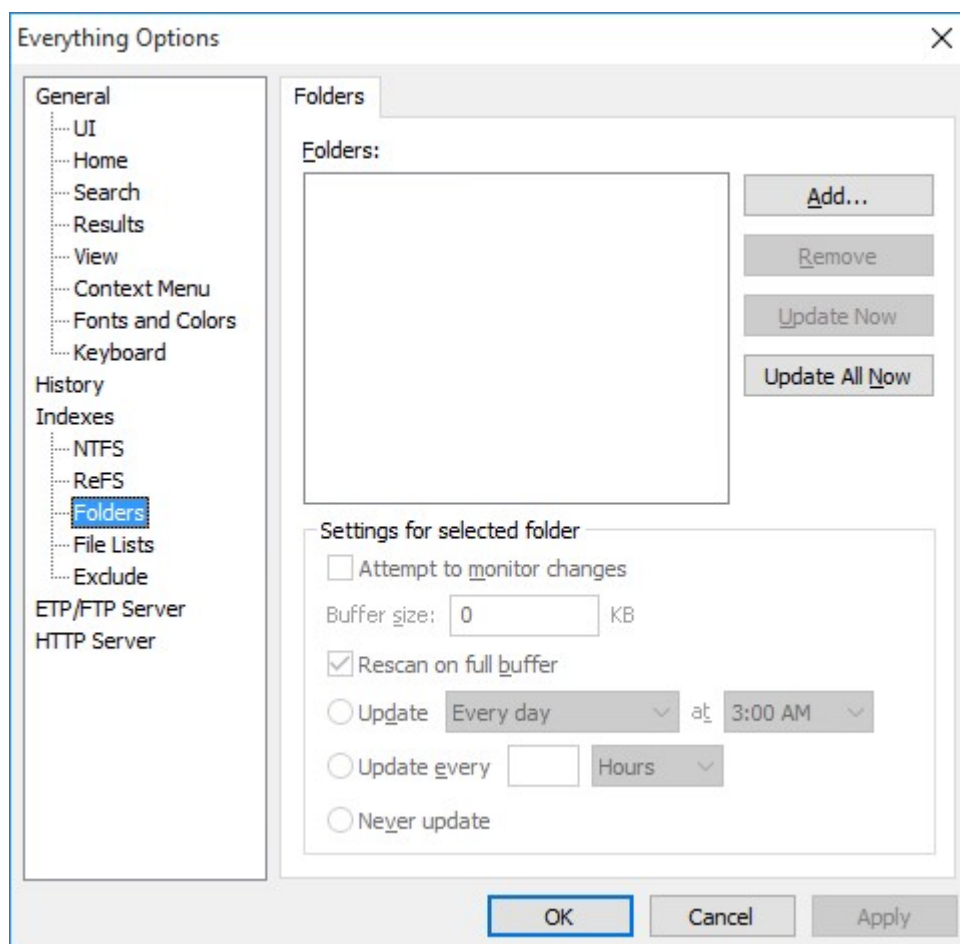
Include only

semicolon delimited list of folders to include on this ReFS volume. Leave empty to include all folders.

Monitor changes

Keep the ReFS index of this ReFS volume up to date.

Folders



Contains settings for configuring folder indexes.

Folders

The list of folders to include in the Everything index. Can be used to index Window shares, NAS, CDRoms, DVDROMs, Fat and Fat32 volumes.

To add a folder to the list, click Add....

To remove a folder from the list click Remove....

To update a folder in the list now, click Update Now.

To update all folders in the list now, click Update All Now

Attempt to monitor changes

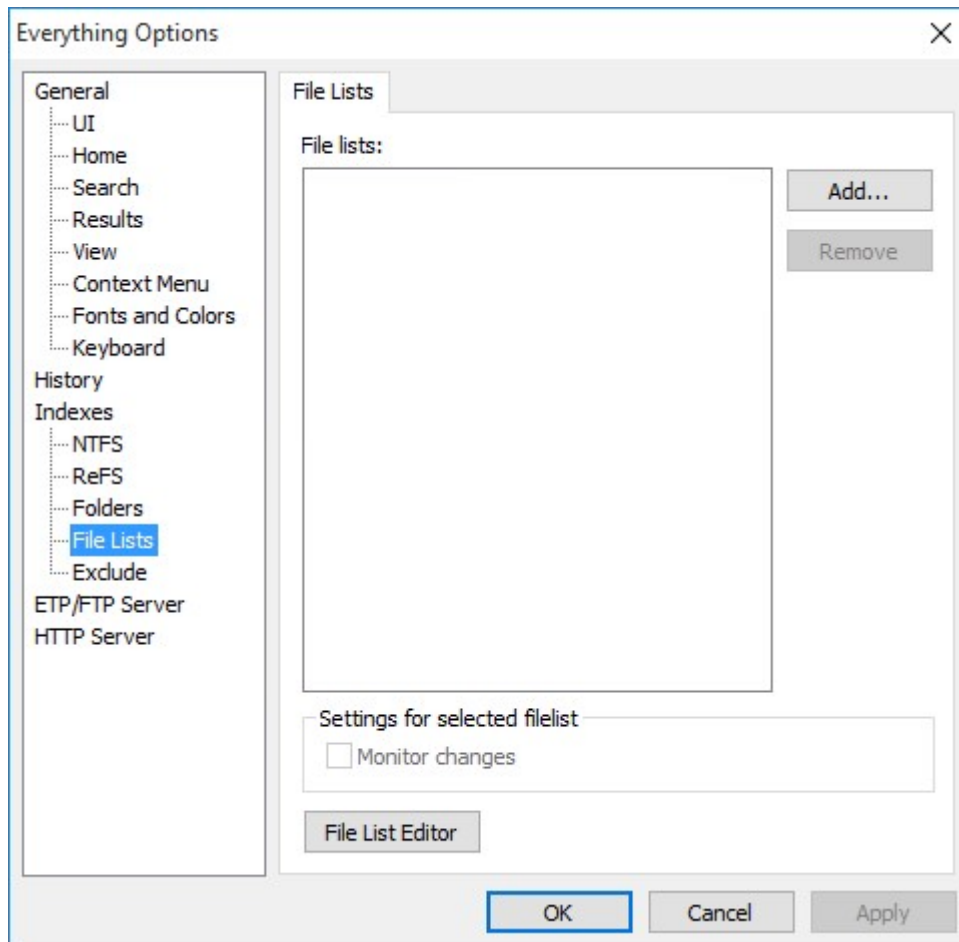
Attempt to monitor changes to the selected folder.

Update

Choose an option for when to update the selected folder.

See [Folder Indexing](#) for more information.

File Lists



Contains settings for configuring file lists indexes.

File lists

The list of file lists to include in the Everything index.

To add a file list to the list, click Add....

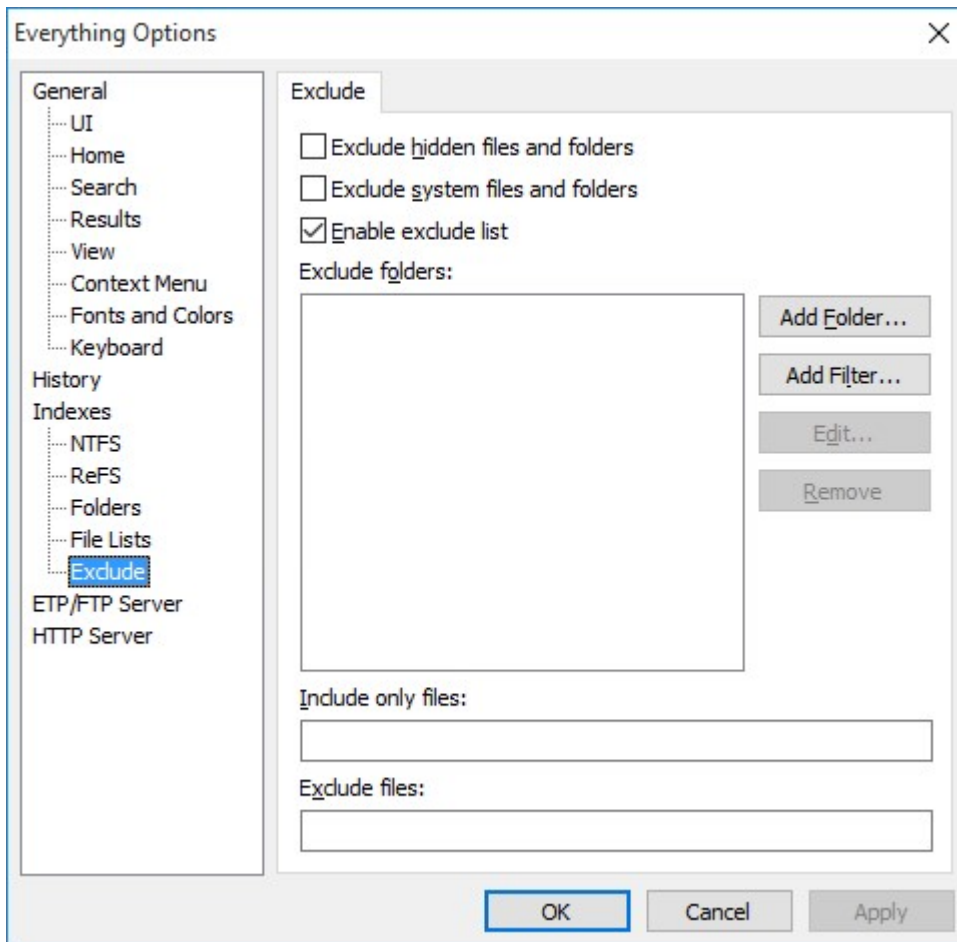
To remove a file list from the list, click Remove.

Click the File List Editor button to open the file lists editor.

The file list editor can create a new file list or edit an existing one.

See [File lists](#) for more information.

Exclude



Contains settings for exclude files and folders from the index.

Exclude hidden files and folders

Exclude files and folders that have the hidden attribute set.

Exclude system files and folders

Exclude files and folders that have the system attribute set.

Exclude folders

A list of folders to exclude from the index.

To add a folder to the list, click Add Folder....

To add a wildcard filter to the list, click Add Filter....

To add a regex filter to the list, click Add Filter.... and prefix your filter with regex:

To remove a folder or filter from the list, click Remove.

For example, to exclude the windows folders:

- Click Add Folder....
- Type in:

C:\Windows

- Click OK.

Include only files

A semicolon delimited wildcard filter list of files to include. Leave empty to include all files.

For example, include only mp3 and mkv files:

.mp3;.mkv

Prefix a filter with regex: to enable regex.

For example, to include only files starting with A-M:

regex:^[A-M]

Exclude files

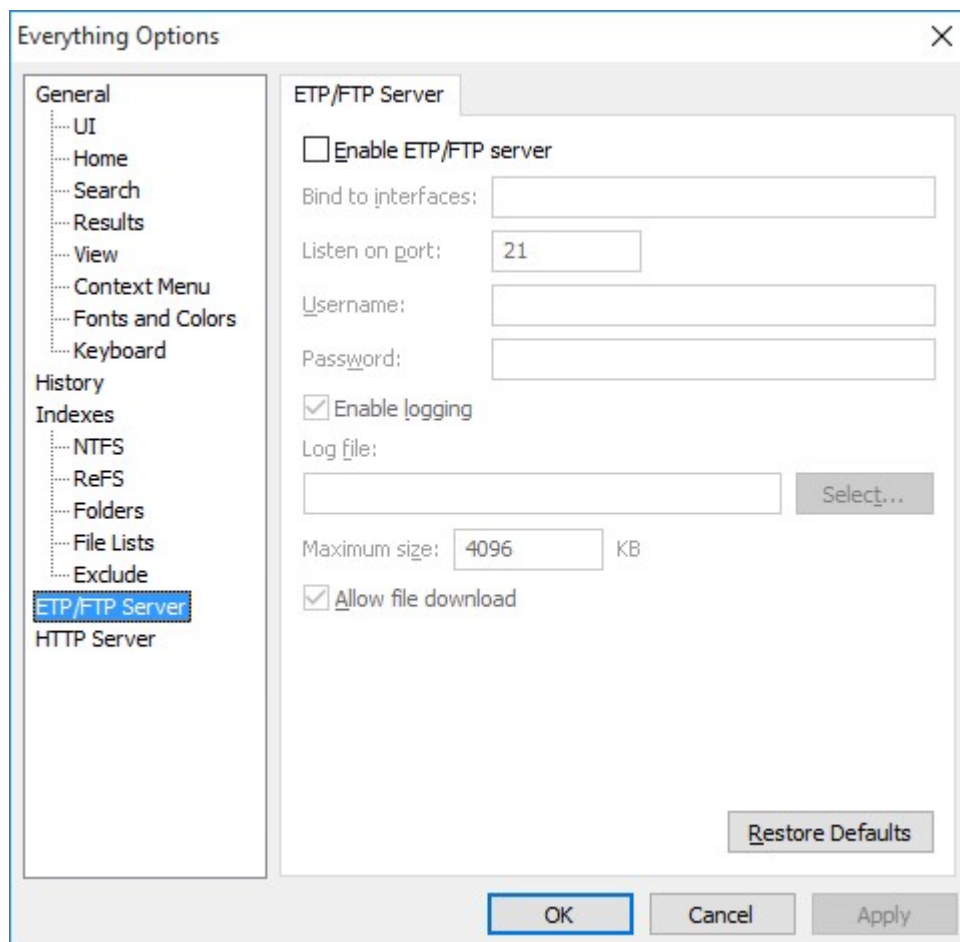
A semicolon delimited wildcard filter list of files to exclude. Leave empty to include all files.

For example, exclude db files

*.db

Prefix a filter with regex: to enable regex.

ETP/FTP Server



Contains settings for configuring a ETP/FTP server to access your files remotely from Everything or a FTP client. For more information see [ETP/FTP Server](#).

Enable ETP/FTP server

Enable the ETP/FTP server.

Bind to interfaces

Bind the listening socket to the specified interfaces. Use 127.0.0.1 to bind to the loopback adaptor. Only local connections can be made to the loopback adaptor. Leave empty to bind to all interfaces.

Listen on port

Set the listening socket port. The ETP/FTP client must also connect on this port.

Username

The ETP/FTP username. The user must enter the correct username and password before performing an action. Leave empty to ignore the username. The username is sent as raw text.

Password

The ETP/FTP password. The user must enter the correct username and password before performing an action. Leave empty to ignore the password. The password is sent as raw text.

Enable logging

Log all ETP/FTP events to a log file.

Log file

The filename of the log file. Click Select to choose the log filename.

Maximum size

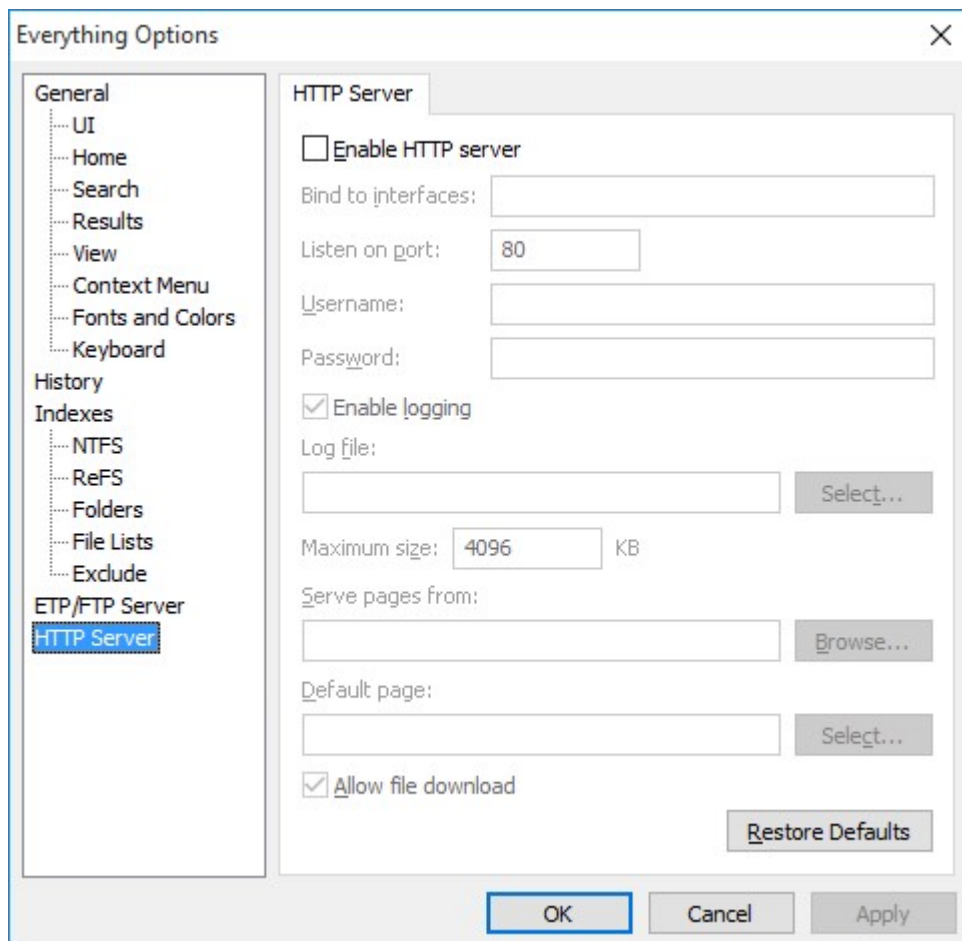
The maximum size of the log file in KB. The log file is trimmed from the start when it is larger than this size.

Allow file download

Allow files to be downloaded from the ETP/FTP server. Disable to only allow browsing.

Use [multiple instances](#) to create an ETP/FTP server with a custom index.

HTTP Server



Contains settings for configuring a HTTP server to access your files remotely from a web browser. For more information see [HTTP Server](#).

Enable HTTP server

Enable the HTTP server.

Bind to interfaces

Bind the listening socket to the specified interfaces. Use 127.0.0.1 to bind to the loopback adaptor. Only local connections can be made to the loopback adaptor. Leave empty to bind to all interfaces.

Listen on port

Set the listening socket port. The web browser must also connect on this port.

Username

The HTTP username. The user must enter the correct username and password before performing an action. Leave empty to ignore the username. The username is sent as raw text.

Password

The HTTP password. The user must enter the correct username and password before performing an action. Leave empty to ignore the password. The password is sent as raw text.

Enable logging

Log all HTTP events to a log file.

Log file

The filename of the log file. Click Select to choose the log filename.

Maximum size

The maximum size of the log file in KB. The log file is trimmed from the start when it is larger than this size.

Serve pages from

Select the folder to load HTTP files from. These files are not required to be in the index. Leave empty to server files from %APPDATA%\Everything\HTTP Server.

Default page

Select the default page. Leave blank to show a list of volumes.

Allow file download

Allow files to be downloaded from the HTTP server. Disable to only allow browsing.

Use [multiple instances](#) to create an HTTP server with a custom index.

Where are settings stored

Settings are stored in your [Everything.ini](#).

Backup

To make a backup of all settings:

- In **Everything**, type in the following search and press ENTER:

```
/config_save
```

- Choose a filename and destination and click **Save**.

Restore

To restore all settings:

- In **Everything**, type in the following search and press ENTER:

```
/config_load
```

- Select a Everything.ini and click **Open**.

Load defaults

Click the Restore defaults button in the bottom right to load the defaults for the current page.

To reset all settings to their default values, see [Everything.ini](#)

See also

- [Everything.ini](#)

Previous Versions

A list of all the previous versions of Everything.

Previous stable releases:

Everything-1.4.1.986.x86-Setup.exe	2020	Installer	x86	1520 KB
Everything-1.4.1.986.x64-Setup.exe	2020	Installer	x64	1614 KB
Everything-1.4.1.986.x86.zip	2020	Portable	x86	1553 KB
Everything-1.4.1.986.x64.zip	2020	Portable	x64	1424 KB
Everything-1.4.1.969.x86-Setup.exe	2020	Installer	x86	1494 KB
Everything-1.4.1.969.x64-Setup.exe	2020	Installer	x64	1614 KB
Everything-1.4.1.969.x86.zip	2020	Portable	x86	1387 KB
Everything-1.4.1.969.x64.zip	2020	Portable	x64	1549 KB
Everything-1.4.1.935.x86-Setup.exe	2018	Installer	x86	1290 KB
Everything-1.4.1.935.x64-Setup.exe	2018	Installer	x64	1410 KB
Everything-1.4.1.935.x86.zip	2018	Portable	x86	1198 KB
Everything-1.4.1.935.x64.zip	2018	Portable	x64	1359 KB
Everything-1.4.1.877.x86-Setup.exe	2017	Installer	x86	1290 KB
Everything-1.4.1.877.x64-Setup.exe	2017	Installer	x64	1410 KB
Everything-1.4.1.877.x86.zip	2017	Portable	x86	1198 KB
Everything-1.4.1.877.x64.zip	2017	Portable	x64	1359 KB
Everything-1.3.4.686.x86-Setup.exe	2014	Installer	x86	463 KB
Everything-1.3.4.686.x64-Setup.exe	2014	Installer	x64	546 KB
Everything-1.3.4.686.x86.zip	2014	Portable	x86	433 KB
Everything-1.3.4.686.x64.zip	2014	Portable	x64	547 KB
Everything-1.2.1.371.exe	2009	Installer	x86	589 KB
Everything-1.2.1.371.zip	2009	Portable	x86	272 KB
Everything-1.1.4.301.exe	2008	Installer	x86	293 KB
Everything-1.0.0.0.exe	2008	Installer	x86	237 KB

Nightly builds

[Everything 1.4.1 Beta Nightly Builds](#)

Friday, 18 February 2019: Version 1.4.1.935

- fixed an issue with really long filenames
- fixed an issue when adding or removing an "include only" exclude.
- fixed an issue with missing changes to excluded files.

[Everything-1.4.1.935.x86-Setup.exe](#)

[Everything-1.4.1.935.x64-Setup.exe](#)

[Everything-1.4.1.935.x86.zip](#)

[Everything-1.4.1.935.x64.zip](#)

Friday, 26 January 2019: Version 1.4.1.932

- fixed a crash when gathering NTFS file information
- fixed an issue with updating folder information

[Everything-1.4.1.932.x86-Setup.exe](#)

[Everything-1.4.1.932.x64-Setup.exe](#)

[Everything-1.4.1.932.x86.zip](#)

[Everything-1.4.1.932.x64.zip](#)

Friday, 11 January 2019: Version 1.4.1.928

- fixed a crash with ntdll.dll
- fixed a crash when renaming a folder with excluded subfolders
- fixed an issue with calculating folder children counts when using folder excludes

[Everything-1.4.1.928.x86-Setup.exe](#)

[Everything-1.4.1.928.x64-Setup.exe](#)

[Everything-1.4.1.928.x86.zip](#)

[Everything-1.4.1.928.x64.zip](#)

Saturday, 15 December 2018: Version 1.4.1.924

- improved thread yielding
- fixed an issue with Everything detecting renamed files.
- fixed an issue when restarting Everything would restart as administrator.
- fixed an icon overlay display issue.
- fixed an issue with detecting renamed files on ReFS volumes.
- fixed an issue with deleting or renaming ReFS folders.

[Everything-1.4.1.924.x86-Setup.exe](#)

[Everything-1.4.1.924.x64-Setup.exe](#)

[Everything-1.4.1.924.x86.zip](#)

[Everything-1.4.1.924.x64.zip](#)

Friday, 9 February 2018: Version 1.4.1.895

- added .jpeg extension to the image information search functions.

- added support for cluster sizes larger than 64k.
- added preview_context ini setting.
- updated localization
- fixed a crash when clearing search history.
- fixed a security issue when running a FTP server.
- fixed an issue preventing Everything from closing when pressing ESC
- fixed an issue where Everything would lose the current sort after reindexing.
- fixed an issue when using invalid characters to rename a file.
- fixed a crash when rendering tall or wide images.
- fixed an issue when shutting down Everything would cause it to miss changes.
- fixed a memory alignment issue.
- fixed database corruption when renaming a folder when folder size indexing is enabled or when there is a folder exclude.
- fixed an issue with circular NTFS/ReFS folder parents.

[Everything-1.4.1.895.x86-Setup.exe](#)

[Everything-1.4.1.895.x64-Setup.exe](#)

[Everything-1.4.1.895.x86.zip](#)

[Everything-1.4.1.895.x64.zip](#)

Wednesday, 7 June 2017: Version 1.4.1.877

- added option to index size, dates and attributes
- added option to enable fast sorting of size, dates, attributes, path and extension
- added thumbnail view
- added preview pane
- added REFS support
- added advanced searching
- added multi-file renaming
- added content searching
- added search history organizer
- added show total size in status bar option
- added single click open option
- added full row select option
- added perl compatible regex
- added new search functions
- added / search commands
- added filter functions
- added new Everything icon
- improved folder deletion performance
- improved indexing when using file and folder excludes
- improved search term highlighting
- improved database blocking
- improved indexes updating
- improved folder index monitoring
- improved NTFS indexing
- improved re-indexing performance when the existing indexes are up to date
- improved IPC
- fixed an issue causing Everything to hang onto a volume handle when trying to safely remove a device
- fixed an issue with excludes being ignored when rebuilding the indexes
- fixed a crash when renaming a folder that had subfolders that should be excluded
- fixed an issue with sorting by date created ascending and date accessed ascending

- fixed a crash when accessing file information cache
- fixed an issue with ETP url links and special characters
- fixed a focus issue with UAC prompts
- fixed an issue when opening a file as admin
- fixed a lock in the Everything service
- fixed an issue with a race condition and initializing COM
- fixed a hang when deleting a registry key
- fixed an issue with very large sorts
- fixed an issue with !parent:, !child: and !c:\path
- fixed an issue with filter modifiers
- fixed an issue with file:!
- fixed an issue with indexing hidden or system folders/files
- fixed an issue with previous navigation searches
- fixed an issue when searching for a \path\
- fixed a crash when cutting an item multiple times

[Everything-1.4.1.877.x86-Setup.exe](#)

[Everything-1.4.1.877.x64-Setup.exe](#)

[Everything-1.4.1.877.x86.zip](#)

[Everything-1.4.1.877.x64.zip](#)

Saturday, 13 May 2017: Version 1.4.1.873b

- fixed an issue with count: and descending sorts.
- fixed an issue with namepartdupe:
- added -monitor-pause -monitor-resume

[Everything-1.4.1.873b.x86-Setup.exe](#)

[Everything-1.4.1.873b.x64-Setup.exe](#)

[Everything-1.4.1.873b.x86.zip](#)

[Everything-1.4.1.873b.x64.zip](#)

Tuesday, 11 April 2017: Version 1.4.1.867b

- fixed an issue with excludes being ignored when rebuilding the indexes
- fixed a crash when renaming a folder that had subfolders that should be excluded
- fixed an issue with sorting by date created ascending and date accessed ascending
- fixed an issue with gathering file info from fragmented attribute lists
- fixed an issue with unnecessary extended file information gathering with IPC requests
- fixed a crash when accessing file information cache
- fixed an issue with ETP url links and special characters
- fixed an issue with vertical scrolling thumbnails by page
- fixed a focus issue with UAC prompts
- fixed an issue when opening a file as admin
- fixed an issue with the preview not updating when changing the focus
- fixed a lock in the Everything service
- fixed an issue with a race condition and initializing COM
- fixed a hang when deleting a registry key
- fixed a hang when the service allocated too much memory
- fixed a crash when stopping service monitors
- fixed an issue causing Everything to hang onto a volume handle when trying to safely remove a device.

- fixed a crash when reading directory changes.
- fixed an issue that would miss NTFS volumes.
- fixed an issue with very large sorts.
- fixed an issue that would cause the folder monitors to stop.
- fixed an issue with automatically removing offline custom volumes.
- fixed a hang when indexing a remote folder.
- fixed an issue with !parent:, !child: and !c:\path
- improved folder deletion performance
- improved indexing when using file and folder excludes
- added -create-file-list exclude options
- added Unicode Properties to Regex
- added folder sizes to HTTP server
- added count, frn, id3, flac tags and image dimensions search functions.
- added support for ReFS 3.x

[Everything-1.4.1.867b.x86-Setup.exe](#)

[Everything-1.4.1.867b.x64-Setup.exe](#)

[Everything-1.4.1.867b.x86.zip](#)

[Everything-1.4.1.867b.x64.zip](#)

Monday, 21 November 2016: Version 1.4.1.809b

- fixed a synchronization issue with the Everything service
- fixed an issue with filter modifiers
- fixed an issue with file:!
- added / search commands
- added filter functions

[Everything-1.4.1.809b.x86-Setup.exe](#)

[Everything-1.4.1.809b.x64-Setup.exe](#)

[Everything-1.4.1.809b.x86.zip](#)

[Everything-1.4.1.809b.x64.zip](#)

Tuesday, 25 October 2016: Version 1.4.1.782b

- fixed an issue with namepartdupe:
- fixed exclude hidden files
- fixed a dual display search history positioning issue
- fixed an issue with file:
- fixed an issue with indexing hidden or system folders/files
- fixed an issue when re-indexing an existing NTFS/ReFS index when "Include Only" is set
- added shell: function support
- added millisecond support to time format

[Everything-1.4.1.772b.x86-Setup.exe](#)

[Everything-1.4.1.772b.x64-Setup.exe](#)

[Everything-1.4.1.772b.x86.zip](#)

[Everything-1.4.1.772b.x64.zip](#)

Friday, 14 October 2016: Version 1.4.1.771b

- fixed a display issue with the installer

[Everything-1.4.1.771b.x86-Setup.exe](#)

[Everything-1.4.1.771b.x64-Setup.exe](#)

[Everything-1.4.1.771b.x86.zip](#)

[Everything-1.4.1.771b.x64.zip](#)

Friday, 14 October 2016: Version 1.4.1.770b

- fixed a display issue with tooltips
- fixed a crash when closing Everything
- fixed an issue when excluding hidden files and changing the attributes of a file
- fixed an issue when searching content in empty files
- fixed a selection issue in full row select mode
- fixed a display issue with dates
- fixed an issue with previous navigation searches
- fixed a hang when installing over a running instance
- fixed a blocking issues with search as your type disabled
- fixed a crash when displaying thumbnails
- fixed an indexing issue with small refs volumes
- fixed an issue when selecting with ctrl in full row select mode
- fixed an issue when searching for a \path\
- fixed a crash when cutting an item multiple times
- added perl compatible regex
- added option to never show thumbnails, always icons
- added iso 8601 syntax for dates and time
- added option to disable recent changes
- improved search term highlighting
- improved database blocking
- improved indexes updating
- improved advanced search
- improved multi-file renaming
- improved folder index monitoring

[Everything-1.4.1.770b.x86-Setup.exe](#)

[Everything-1.4.1.770b.x64-Setup.exe](#)

[Everything-1.4.1.770b.x86.zip](#)

[Everything-1.4.1.770b.x64.zip](#)

Tuesday, 24 November 2015: Version 1.4.0.713b

- fixed an issue with indexing deleted attributes in ReFS volumes
- fixed an issue with indexing very large ReFS volumes
- fixed an issue with indexing subst volumes
- fixed an issue with inline renaming and focus wrapping
- added option to remember search in bookmarks
- added option to remember view in bookmarks
- improved unicode normalization
- improved open path command

- improved tooltips

[Everything-1.4.0.713b.x86-Setup.exe](#)

[Everything-1.4.0.713b.x64-Setup.exe](#)

[Everything-1.4.0.713b.x86.zip](#)

[Everything-1.4.0.713b.x64.zip](#)

Sunday, 11 October 2015: Version 1.4.0.705b

- fixed an issue when saving a refs index
- fixed an issue with multiple service instances

[Everything-1.4.0.705b.x86.zip](#)

[Everything-1.4.0.705b.x64.zip](#)

Sunday, 11 October 2015: Version 1.4.0.704b

- fixed an issue with displaying wrong sizes
- fixed an issue with displaying corrupt file names
- fixed an issue with missing sizes and dates
- fixed an issue with unnecessary accessing floppy disks
- fixed an issue when renaming multi-files with non-ascii characters
- fixed a localization issue with some values in the config file
- fixed an issue with indexing substituted volumes
- fixed an crash after performing an IPC query
- fixed an issue with install command line options not working if Everything was already running
- added thumbnail load priority ui option
- added service name customization by changing the instance name
- improved preview support for images on xp
- improved inherited options when creating a new window
- improved -update -exit

[Everything-1.4.0.704b.x86.zip](#)

[Everything-1.4.0.704b.x64.zip](#)

Friday, 2 October 2015: Version 1.4.0.703b

- fixed an issue with Everything.db being stored in the wrong location
- fixed an issue with MSAA and SysListView32 emulation
- improved debug console support

[Everything-1.4.0.703b.x86.zip](#)

[Everything-1.4.0.703b.x64.zip](#)

Wednesday, 30 September 2015: Version 1.4.0.702b

- added option to index size, dates and attributes
- added option to enable fast sorting of size, dates, attributes, path and extension
- added thumbnail view

- added preview pane
- added REFS support
- improved NTFS indexing
- improved indexing performance
- added advanced searching
- added multi-file renaming
- added content searching
- improved re-indexing performance when the existing indexes are up to date
- added search history organizer
- added show total size in status bar option
- added single click open option
- added full row select option
- improved IPC

[Everything-1.4.0.702b.x86.zip](#)

[Everything-1.4.0.702b.x64.zip](#)

Monday, 4 August 2014: Version 1.3.4.685b

- fixed an issue with folders not being removed from a folder index
- added -install-config command line option

[Everything-1.3.4.685b.x86.zip](#)

[Everything-1.3.4.685b.x64.zip](#)

Thursday, 31 July 2014: Version 1.3.4.684b

- added service debugging option

[Everything-1.3.4.684b.x86.zip](#)

[Everything-1.3.4.684b.x64.zip](#)

Saturday, 26 July 2014: Version 1.3.4.683b

- fixed an issue with path ordering
- fixed a crash when an error occurred in a uncompressed stream

[Everything-1.3.4.683b.x86.zip](#)

[Everything-1.3.4.683b.x64.zip](#)

Friday, 25 July 2014: Version 1.3.4.682b

- fixed an issue with path ordering

[Everything-1.3.4.682b.x86.zip](#)

[Everything-1.3.4.682b.x64.zip](#)

Thursday, 24 July 2014: Version 1.3.4.681b

- fixed a crash when removing a folder in a folder index

[Everything-1.3.4.681b.x86.zip](#)

[Everything-1.3.4.681b.x64.zip](#)

Sunday, 20 July 2014: Version 1.3.4.680b

- fixed a crash when dragging a file or folder over Everything

[Everything-1.3.4.680b.x86.zip](#)

[Everything-1.3.4.680b.x64.zip](#)

Thursday, 17 July 2014: Version 1.3.4.679b

- fixed an issue that would cause a connection failure to the service on startup
- fixed an issue that would cause deleting everything.exe to fail when uninstalling

[Everything-1.3.4.679b.x86.zip](#)

[Everything-1.3.4.679b.x64.zip](#)

Monday, 14 July 2014: Version 1.3.4.678b

- fixed an issue with ext: being garbled

[Everything-1.3.4.678b.x86.zip](#)

[Everything-1.3.4.678b.x64.zip](#)

Saturday, 12 July 2014: Version 1.3.4.677b

- fixed an issue when displaying tooltips with &s

[Everything-1.3.4.677b.x86.zip](#)

[Everything-1.3.4.677b.x64.zip](#)

Wednesday, 9 July 2014: Version 1.3.4.676b

- added ini setting menu_escape_amp to use custom & shortcuts in menus
- fixed an issue when renaming a file and losing focus

[Everything-1.3.4.676b.x86.zip](#)

[Everything-1.3.4.676b.x64.zip](#)

Monday, 7 July 2014: Version 1.3.4.673b

- fixed an issue with paths in bookmarks

[Everything-1.3.4.673b.x86.zip](#)

[Everything-1.3.4.673b.x64.zip](#)

Monday, 7 July 2014: Version 1.3.4.672b

- fixed an issue when installing the x64 version over an existing x86 install
- added ini option keep_missing_volumes

[Everything-1.3.4.672b.x86.zip](#)

[Everything-1.3.4.672b.x64.zip](#)

Friday, 4 July 2014: Version 1.3.4.671b

- fixed an issue with duplicated everything filters
- added xbutton support
- added browser back/forward key support

[Everything-1.3.4.671b.x86.zip](#)

[Everything-1.3.4.671b.x64.zip](#)

Monday, 30 June 2014: Version 1.3.4.670b

- fixed a display issue with menu commands
- improved -uninstall command line option
- updated default filters

[Everything-1.3.4.670b.x86.zip](#)

[Everything-1.3.4.670b.x64.zip](#)

Saturday, 28 June 2014: Version 1.3.4.669b

- fixed vertical spacing of icons in listview
- fixed minor layout and localization issues in installer
- fixed an issue that prevented rebuilding the indexes after installing the service

[Everything-1.3.4.669b.x86.zip](#)

[Everything-1.3.4.669b.x64.zip](#)

Wednesday, 25 June 2014: Version 1.3.4.667b

- added previous language selection to installer
- fixed a hang when uninstalling the service
- fixed an issue with multiple dialogs
- fixed an issue with environment string expansion
- fixed an issue with Everything not getting focus after the runas admin dialog

[Everything-1.3.4.667b.x86.zip](#)

[Everything-1.3.4.667b.x64.zip](#)

Tuesday, 24 June 2014: Version 1.3.4.666b

- fixed an issue with English shortcuts being created after non-English language is selected

[Everything-1.3.4.666b.x86.zip](#)

[Everything-1.3.4.666b.x64.zip](#)

Monday, 23 June 2014: Version 1.3.4.665b

- fixed an issue with the x64 installer installing to x86

[Everything-1.3.4.665b.x86.zip](#)

[Everything-1.3.4.665b.x64.zip](#)

Monday, 23 June 2014: Version 1.3.4.664b

- added non-admin options to portable version
- added installer localization
- added MIT license

[Everything-1.3.4.664b.x86.zip](#)

[Everything-1.3.4.664b.x64.zip](#)

Tuesday, 17 June 2014: Version 1.3.4.663b

- fixed an issue with updating the db

[Everything-1.3.4.663b.x86.zip](#)

[Everything-1.3.4.663b.x64.zip](#)

Saturday, 7 June 2014: Version 1.3.4.662b

- added run as admin option
- added wildcards anywhere option
- added run date column/sort/search
- added support for accessible MSAA
- added option to toggle left or right filter bar
- added etp connection history
- improved efficiency of the Everything service
- improved RTL support
- improved multi-monitor support
- improved performance when renaming/deleting folders
- improved run history after renaming/deleting an item
- fixed an issue with monitors stopping and not starting again
- fixed an issue with Everything not getting focus on startup
- fixed a crash when closing a window

- fixed a crash when adding an item to an empty array.
- fixed an issue with whole words and non ascii chars
- fixed a GDI object leak
- fixed an issue when searching for paths using wildcards

[Everything-1.3.4.662b.x86.zip](#)

[Everything-1.3.4.662b.x64.zip](#)

Wednesday, 26 June 2013: Version 1.3.3.658b

- fixed an issue with monitoring icons and file info cache
- fixed a crash when canceling a sort
- fixed an issue with threads live locking

[Everything-1.3.3.658b.x86.zip](#)

[Everything-1.3.3.658b.x64.zip](#)

Monday, 24 June 2013: Version 1.3.3.657b

- fixed an issue when installing the service
- fixed an issue with the IPC and x64
- fixed an issue with search as you type
- fixed an issue with unnecessary db rebuilds
- improved type name sorting

[Everything-1.3.3.657b.x86.zip](#)

[Everything-1.3.3.657b.x64.zip](#)

Thursday, 20 June 2013: Version 1.3.3.656b

- fixed an issue with updating icons.
- fixed a crash when sorting folders by size.
- fixed a crash when updating folder indexes with an excluded folder.
- improved sorting

[Everything-1.3.3.656b.x86.zip](#)

[Everything-1.3.3.656b.x64.zip](#)

Sunday, 16 June 2013: Version 1.3.3.653b

- added recent changes
- added run history
- added home page
- added x64 build
- fixed a crash when deleting or renaming a folder with a short file name

[Everything-1.3.3.653b.x86.zip](#)

[Everything-1.3.3.653b.x64.zip](#)

Tuesday, 7 May 2013: Version 1.3.2.649b

- fixed an issue with matching whole words
- fixed a crash when requesting type information
- added client service
- fixed a crash when requesting admin privileges
- fixed a crash when displaying corrupt icons

[Everything-1.3.2.649b.zip](#)

Saturday, 4 May 2013: Version 1.3.2.647b

- fixed an issue with creating threads
- fixed an issue when sorting files with invalid sizes
- fixed an issue with file lists duplicates
- fixed an issue with the everything filter
- moved etp/ftp/http servers to options

[Everything-1.3.2.647b.zip](#)

Thursday, 18 April 2013: Version 1.3.2.645b

- fixed a crash when reading extended file info
- fixed an issue with sending http files

[Everything-1.3.2.645b.zip](#)

Saturday, 13 April 2013: Version 1.3.2.644b

- fixed a crash when the index contains duplicated folders

[Everything-1.3.2.644b.zip](#)

Thursday, 11 April 2013: Version 1.3.2.643b

- fixed a crash when sorting

[Everything-1.3.2.643b.zip](#)

Thursday, 11 April 2013: Version 1.3.2.642b

- added ETP/FTP server
- added HTTP server
- added always ontop option
- fixed a crash when renaming a file or folder
- reset keyboard shortcuts

[Everything-1.3.2.642b.zip](#)

Wednesday, 20 February 2013: Version 1.3.1.636b

- added %APPDATA% option
- added Everything service option
- added search history option
- added the option to cancel locked queries after one second
- added context menu customization
- added tooltips for hidden items
- added option to not compress database
- fixed an issue with right to left text rendering
- fixed an issue parsing literal ops
- fixed an issue expanding filter searches
- fixed an issue parsing macros
- fixed an issue when sorting by size with a selection
- fixed a db event issue
- fixed an issue with wildcard exclude filters
- improved db locking mechanism

[Everything-1.3.1.636b.zip](#)

Thursday, 10 January 2013: Version 1.3.0.632b

- fixed non-unicode clipboard issue with non-ansi locale.
- fixed a bug causing the db to block while typing a search.
- fixed a bug with context menus and opening sub menus.
- fixed wrong mutex name.
- fixed shift + esc hotkey
- added custom filters

[Everything-1.3.0.632b.zip](#)

Sunday, 6 January 2013: Version 1.3.0.631b

- fixed a leak with ext:
- added hotkey option to toggle search window
- fixed an issue with bookmark macros and diacritics:

[Everything-1.3.0.631b.zip](#)

Sunday, 6 January 2013: Version 1.3.0.630b

- fixed an issue with the -instance <name> command line option

[Everything-1.3.0.630b.zip](#)

Saturday, 5 January 2013: Version 1.3.0.629b

- added filelists
- added folder indexing
- added optional diacritical mark matching
- added bookmarks
- added new search functions and macros
- removed etp/ftp
- removed http
- removed custom filters
- removed search history
- removed run history
- removed recent changes
- improved performance of many search functions
- fixed many bugs

[Everything-1.3.0.629b.zip](#)

Friday, 24 November 2009: Version 1.2.1.451a

Early test release.

[Everything-1.2.1.451a.zip](#)

Friday, 13 March 2009: Version 1.2.1.371

- added title customization
- added NOT operator (!)
- added special case wildcards
- added authentication support for QTWeb
- added the option to exclude hidden and system files
- added a basic text export option
- added search only mode for HTTP/FTP servers
- added shortcut keys for search options
- added special case wildcards
- added utf8 support for the FTP server
- added keyboard shortcut Ctrl + Enter to open path
- added extension and type columns
- added support for external file managers via the ini file
- added exclude paths to etp/ftp/http servers
- added unicode support
- added multiple language support
- added removable NTFS volume support
- added exclude folder filter
- added option to check media on removable volumes (floppy drives are disabled by default)
- added custom icon support
- added optional http server user name and password
- added option to use FTP links for ETP connections
- added option to choose which drives are included in the installer
- added fullscreen mode (F11)

- added portability
- improved rendering performance
- improved search parsing
- improved multiple instances option
- improved command line parsing
- improved option window navigation with keyboard
- improved tooltip help for some items
- improved searching performance with wildcards by upto 5 times
- improved searching performance by upto 30%
- improved ETP/FTP/HTTP performance
- improved text rendering output and performance
- improved rendering performance when updating the UI due to a large amount of changes to the file system.
- improved memory usage when database is unload by clearing icon and file info cache
- improved volume management
- improved sorting performance
- improved ftp command LIST
- fixed an issue with HTTP url parsing
- fixed default icons for windows 7 beta
- fixed user handle leak in context menus
- fixed a problem with cut/copy/paste
- fixed cut with files and folders in different folders
- fixed start menus for Vista
- fixed opening a folder that had the same name as an executable
- fixed the FTP REST position
- fixed unicode IPC queries
- fixed sorting by descending size
- fixed characters being displayed incorrectly with the HTTP server
- fixed copying files and folders in different folders.
- fixed drag drop shortcuts
- fixed an issue when searching for wildcards with match path enabled
- fixed delete for files and folders in different folders
- fixed an issue with long file names
- fixed the folder and file selection after sorting and changing search parameters
- fixed folder and file renaming using invalid destination
- fixed mouse wheel scrolling using the wrong number of lines to scroll
- fixed icons for files with the extension lnk and url
- fixed show shell context menus toggle in options window
- fixed a problem with menus blocking after a search was started
- fixed -filename command line option using the wrong path
- fixed new windows inheriting old windows settings
- fixed horizontal scroll bar using wrong width
- fixed a gdi leak when closing a window
- fixed the default large icon

[Everything-1.2.1.371.exe](#) (installer)

[Everything-1.2.1.371.zip](#) (portable)

Thursday, 5 March 2009: Version 1.2.1.369

- improved search parsing
- improved multiple instances option
- improved command line parsing

- improved option window navigation with keyboard
- fixed default icons for windows 7 beta
- added title customization

[Everything-1.2.1.369b.exe](#) (installer)

[Everything-1.2.1.369b.zip](#) (portable)

Saturday, 21 February 2009: Version 1.2.1.360

- added NOT operator (!)
- added localization to Context menus, Start menus, Quick Launch and the Desktop
- removed restart on language pack change
- fixed user handle leak in context menus
- added more special case wildcards

[Everything-1.2.1.360b.exe](#) (installer)

[Everything-1.2.1.360b.zip](#) (portable)

Sunday, 15 February 2009: Version 1.2.1.358

- added localization for export

[Everything-1.2.1.358b.exe](#) (installer)

[Everything-1.2.1.358b.zip](#) (portable)

Sunday, 15 February 2009: Version 1.2.1.356

- improved tooltip help for some items

[Everything-1.2.1.356b.exe](#) (installer)

[Everything-1.2.1.356b.zip](#) (portable)

Friday, 13 February 2009: Version 1.2.1.355

- added authentication support for QTWeb
- fixed a problem with cut/copy/paste
- fixed cut with files and folders in different folders
- fixed about box widget alignment

[Everything-1.2.1.355b.exe](#) (installer)

[Everything-1.2.1.355b.zip](#) (portable)

Tuesday, 10 February 2009: Version 1.2.1.352

- improved searching performance with wildcards by upto 5 times
- improved searching performance by upto 30%
- added the option to exclude hidden and system files
- improved the exclude list

- added a basic text export option
- improved ETP/FTP/HTTP performance
- improved text rendering output and performance
- improved rendering performance when updating the UI due to a large amount of changes to the file system.
- fixed start menus for Vista
- added search only mode for HTTP/FTP servers
- added shortcut keys for search options
- fixed opening a folder that had the same name as an executable
- fixed the FTP REST position
- fixed unicode IPC queries
- fixed sorting by descending size
- fixed characters being displayed incorrectly with the HTTP server
- fixed copying files and folders in different folders.
- fixed drag drop shortcuts
- added special case wildcards
- added utf8 support for the FTP server
- added keyboard shortcut Ctrl + Enter to open path

Thursday, 13 November 2008: Version 1.2.0.323

- fixed an issue when searching for wildcards with match path enabled
- fixed delete for files and folders in different folders
- fixed a portability issue with the default database and log locations
- fixed an issue with long file names
- fixed the folder and file selection after sorting and changing search parameters
- added extension and type columns
- added support for external file managers via the ini file

[Everything-1.2.0.323b.exe](#) (installer)

[Everything-1.2.0.323b.zip](#) (portable)

Saturday, 8 November 2008: Version 1.2.0.322

- fixed rendering of paths with highlighted search terms enabled

[Everything-1.2.0.322b.exe](#) (installer)

[Everything-1.2.0.322b.zip](#) (portable)

Saturday, 8 November 2008: Version 1.2.0.321

- fixed folder and file renaming using invalid destination
- fixed mouse wheel scrolling using the wrong number of lines to scroll
- fixed icons for files with the extension lnk and url
- improved unicode sorting order
- added exclude paths to etp/ftp/http servers

[Everything-1.2.0.321b.exe](#) (installer)

[Everything-1.2.0.321b.zip](#) (portable)

Tuesday, 28 October 2008: Version 1.2.0.318

- fixed http server security issue when a password was set
- fixed show shell context menus toggle in options window
- fixed fullscreen mode saving wrong window position rect when closing
- fixed a problem with menus blocking after a search was started
- improved memory usage when database is unload by clearing icon and file info cache

[Everything-1.2.0.318b.exe](#) (installer)

[Everything-1.2.0.318b.zip](#) (portable)

Saturday, 25 October 2008: Version 1.2.0.317

- added unicode support
- added multi-language support
- added removable NTFS volume support
- added exclude folder filter
- improved volume management
- fixed -filename command line option using the wrong path
- improved sorting performance
- added option to check media on removable volumes (floppy drives are disabled by default)
- improved ftp command LIST
- added custom icon support
- improved portability (if the http server files dont exist everything uses the embedded http server files)
- added optional http server user name and password
- fixed new windows inheriting old windows settings
- fixed horizontal scroll bar using wrong width
- fixed a gdi leak when closing a window
- fixed the default large icon
- improved rendering performance
- added option to use FTP links for ETP connections
- added option to choose which drives are included in the installer
- added fullscreen mode (F11)

[Everything-1.2.0.317b.exe](#) (installer)

[Everything-1.2.0.317b.zip](#) (portable)

Monday, 29 September 2008: Version 1.1.4.301

- added -search <text> command line option
- improved status of servers under the tool menu
- changed configuration file to Everything.ini
- fixed a bug that prevented hotkeys from being unregistered

[Everything-1.1.4.301.exe](#) (stable)

Recent Changes

Recent changes is a list of files and folders that have been changed since Everything was started.

Search recent changes

Use `rc:` or `recentchange:` with a date to search for recent changes:

```
rc:<date>
```

For example, to find recent changes today, search for:

```
rc:today
```

For example, to find recent changes yesterday, search for:

```
rc:yesterday
```

For example, to find recent changes in the last 2 hours, search for:

```
rc:last2hours
```

For example, to find recent changes in the last 10 minutes, search for:

```
rc:last10mins
```

For the full `rc:` search syntax, please see [searching](#).

Include the USN Journal in the recent changes list

By default, Everything will only keep a list of changes since Everything was started, including the USN change Journal in the recent changes list will allow you to quickly search for changes that can go back many days.

To include the USN Journals in the recent changes list:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **NTFS** tab.
- For each local NTFS volume:
 - Check **Include USN Journal in recent changes**.
- Click **OK**.

Sort by recent changes

Sorting by recent changes is instant and persistent.

Right click an empty spot in the result list, from the Sort by menu, click Date Recently changed.

Sort by recent changes to monitor changes to the Everything index in real-time.

See also

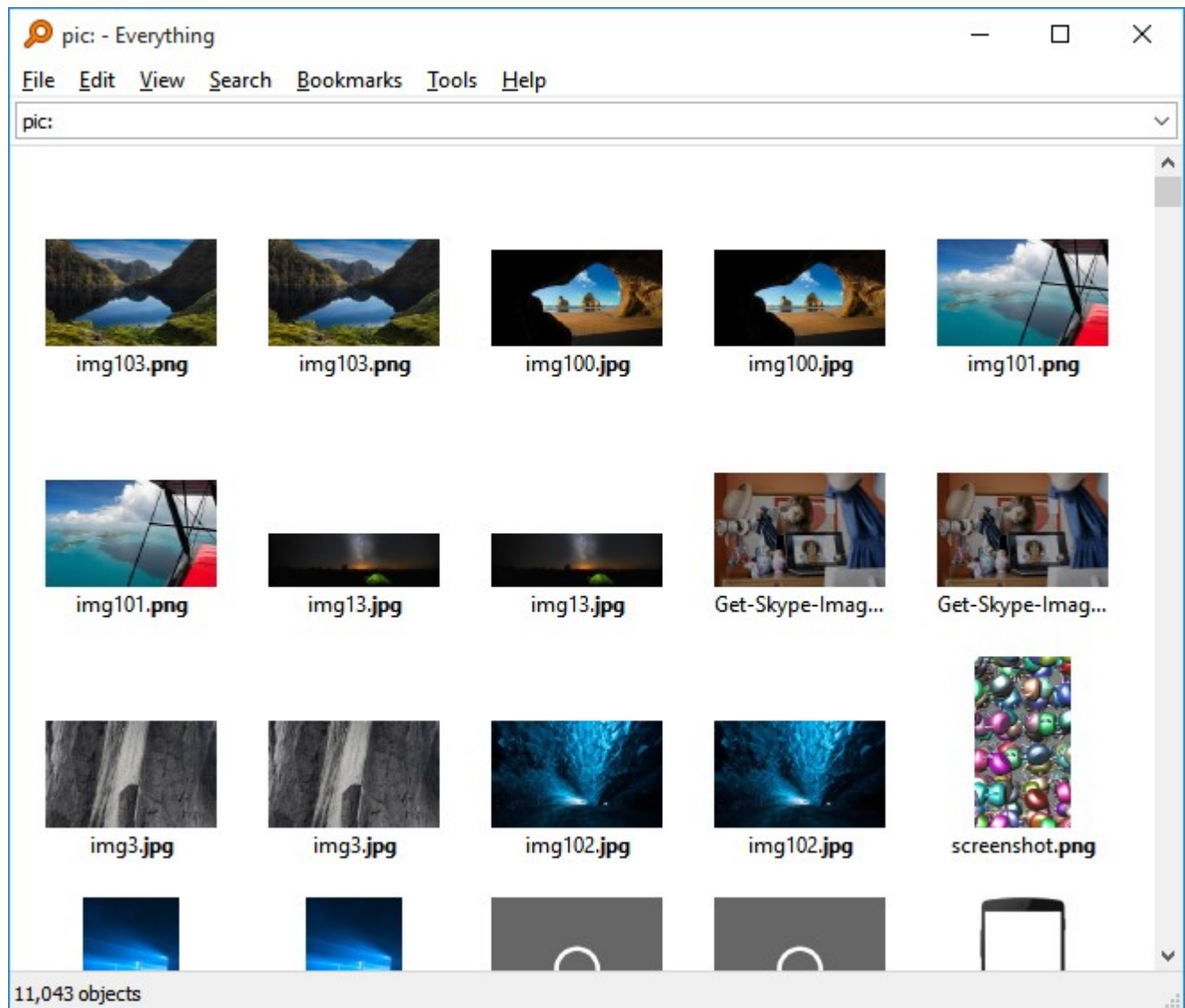
- [rc:](#) and [recentchange:](#) search functions.
- [Include USN Journal in recent changes](#) option.

Results

The result list shows the file and folders matches for the current search.

Right click a file or folder to perform an action, such as open, cut, copy, delete, rename or properties.

Thumbnails



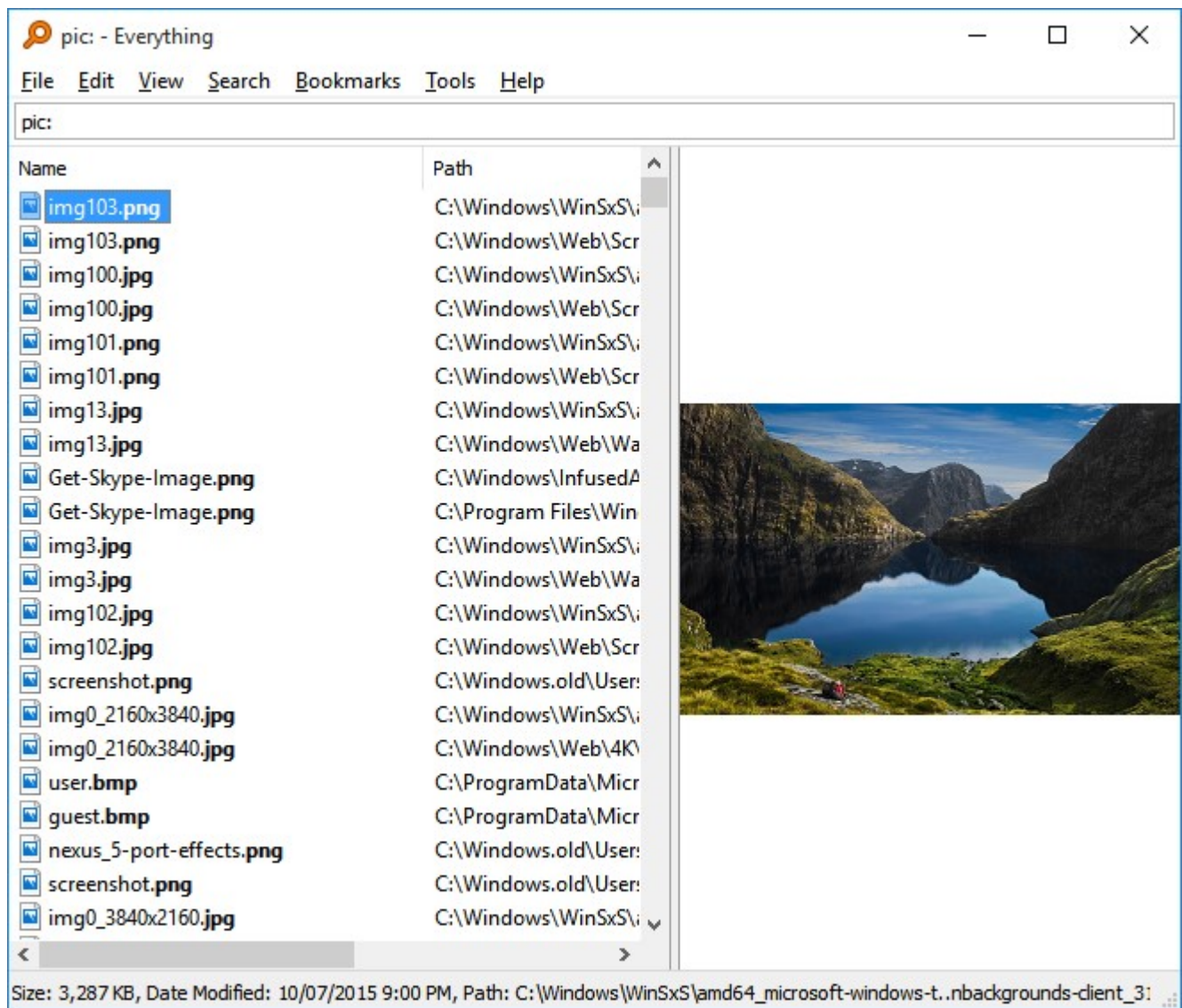
To show results in thumbnail view:

- In "**Everything**", from the **View** menu, click **Medium Thumbnails**, **Large Thumbnails** or **Extra Large Thumbnails**.

-or-

- In "**Everything**", Press and hold the Ctrl key and use the mouse wheel to change the view from Details to Extra Large Thumbnails.

Preview



To show the preview pane:

- In "**Everything**", from the **View** menu, check **Preview**.

Click and drag the divider to resize the preview pane.

Hide results when the search is empty

All files and folders are found when the search is empty. To hide the results when the search is empty:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Check **Hide results when the search is empty**.
- Click **OK**.

Keep selection

When you change the search in Everything, the current selection is cleared.

To keep the selection after changing the search:

- In "**Everything**", from the **Tools** menu, click **Options**.
- Click the **Results** tab.
- Uncheck the **Clear selection on search**.
- Click **OK**.

Jump to

Type in the start of a filename to jump to it.

The jump to text is displayed in the status bar.

Sorting results

By default the results are sorted by Name, then Path.

Sorting is reset back to the default each time the search parameters change.

At the top of the result list is a header. To sort a column click its header.

Note: Sorting can be slow when there are many results.

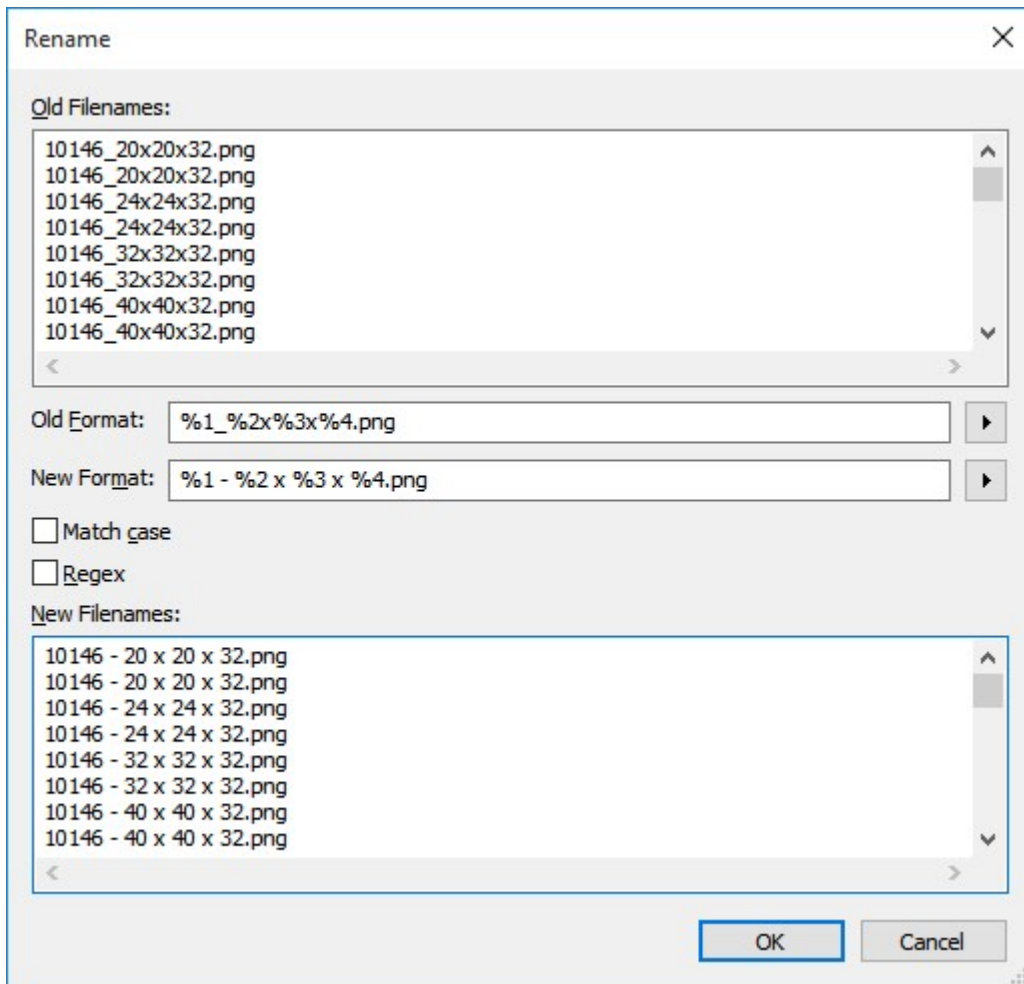
Sorting can be cancelled by selecting another sort or changing the search.

Highlight the row the mouse is over

To highlight the row the mouse is over:

- In "*Everything*", from the **Tools** menu, click **Options**.
- Select the **View** tab.
- Check **Show row mouseover**.
- Click **OK**.

Multi-file Renaming



To use multi-file renaming select multiple files in the result list and press F2.

With multi-file renaming you can:

- Change common text in all filenames.
- Change the extension of all files easily.
- Find and replace text.
- Edit all filenames in one edit box.

Note: multi-file renaming works best when the selected files share similar formats, for example: Artist - Album - Track - Song Title.mp3

Undo

To undo a move, copy or delete:

- Open Windows Explorer (Win + E)
- From the Edit menu.
- Click Undo.

See Also

- [Results Options](#)
- [Change the font and colors](#)

Run History

"Everything" "Run History" is a database of how many times you have run every file or folder from "Everything". Every time you open a file or folder its run count is incremented by one. Pressing Enter from the search edit will focus the file or folder result with the highest run count.

Set an items run count

- **Right click** the file or folder.
- Click **Set Run Count...**
- Enter in a new run count.

Disable Run History

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **History** tab.
- Uncheck **Enable run history**.
- Click **OK**.

Run History Location

Everything stores run history in:

```
%APPDATA%\Everything\Run History.csv
```

If [Store settings and data in %APPDATA%\Everything](#) is disabled, Run History.csv is stored in the same location as Everything.exe.

Search for run count

Use runcount: to search for a specific run count or run count range.

Search for a run count greater than 100:

```
runcount:>100
```

Run once or more:

```
runcount:
```

Search for a run count between 10 and 20:

```
runcount:10-20
```

Search for files you ran yesterday:

```
daterun:yesterday
```

Sort by run count by default

Sorting by run count is instant and persistent.

To sort by run count by default:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Home** tab.
- Change **sort** to **Run Count (Descending)**.
- Click **OK**.

SDK

The Everything SDK provides a DLL and Lib interface to Everything over [IPC](#).

It also provides methods for using WM_COPYDATA for [IPC](#).

Everything is required to run in the background.

Download

[Everything-SDK.zip](#)

Manipulating the search state

- [Everything_SetSearch](#)
- [Everything_SetMatchPath](#)
- [Everything_SetMatchCase](#)
- [Everything_SetMatchWholeWord](#)
- [Everything_SetRegex](#)
- [Everything_SetMax](#)
- [Everything_SetOffset](#)
- [Everything_SetReplyWindow](#)
- [Everything_SetReplyID](#)
- [Everything_SetSort](#)
- [Everything_SetRequestFlags](#)

Reading the search state

- [Everything_GetSearch](#)
- [Everything_GetMatchPath](#)
- [Everything_GetMatchCase](#)
- [Everything_GetMatchWholeWord](#)
- [Everything_GetRegex](#)
- [Everything_GetMax](#)
- [Everything_GetOffset](#)
- [Everything_GetReplyWindow](#)
- [Everything_GetReplyID](#)
- [Everything_GetLastError](#)
- [Everything_GetSort](#)
- [Everything_GetRequestFlags](#)

Executing the query

- [Everything_Query](#)

Check for query reply

- [Everything_IsQueryReply](#)

Manipulating results

- [Everything_SortResultsByPath](#)
- [Everything_Reset](#)

Reading results

- [Everything_GetNumFileResults](#)
- [Everything_GetNumFolderResults](#)
- [Everything_GetNumResults](#)
- [Everything_GetTotFileResults](#)
- [Everything_GetTotFolderResults](#)
- [Everything_GetTotResults](#)
- [Everything_IsVolumeResult](#)
- [Everything_IsFolderResult](#)
- [Everything_IsFileResult](#)
- [Everything_GetResultFileName](#)
- [Everything_GetResultPath](#)
- [Everything_GetResultFullPathName](#)
- [Everything_GetResultListSort](#)
- [Everything_GetResultListRequestFlags](#)
- [Everything_GetResultExtension](#)
- [Everything_GetResultSize](#)
- [Everything_GetResultDateCreated](#)
- [Everything_GetResultDateModified](#)
- [Everything_GetResultDateAccessed](#)
- [Everything_GetResultAttributes](#)
- [Everything_GetResultFileListFileName](#)
- [Everything_GetResultRunCount](#)
- [Everything_GetResultDateRun](#)
- [Everything_GetResultDateRecentlyChanged](#)
- [Everything_GetResultHighlightedFileName](#)
- [Everything_GetResultHighlightedPath](#)
- [Everything_GetResultHighlightedFullPathAndFileName](#)

General

- [Everything_Reset](#)
- [Everything_CleanUp](#)
- [Everything_GetMajorVersion](#)
- [Everything_GetMinorVersion](#)
- [Everything_GetRevision](#)

- [Everything_GetBuildNumber](#)
- [Everything_Exit](#)
- [Everything_IsDBLoaded](#)
- [Everything_IsAdmin](#)
- [Everything_IsAppData](#)
- [Everything_RebuildDB](#)
- [Everything_UpdateAllFolderIndexes](#)
- [Everything_SaveDB](#)
- [Everything_SaveRunHistory](#)
- [Everything_DeleteRunHistory](#)
- [Everything_GetTargetMachine](#)

Run History

- [Everything_GetRunCountFromFileName](#)
- [Everything_SetRunCountFromFileName](#)
- [Everything_IncRunCountFromFileName](#)

Examples

- [C/C++](#)
- [C#](#)
- [Clarion](#)
- [Visual Basic](#)
- [Python](#)

Notes

- The SDK is a basic [IPC](#) wrapper
- Requires "Everything" client to be running.
- ANSI / Unicode support.
- Thread safe.
- Support for blocking and non-blocking mode.
- x86 and x64 support.

See Also

- [IPC](#)

C

Make sure to link to Everything32.lib or Everything64.lib.

Copy Everything32.dll or Everything64.dll to your programs executable directory.

Note: requires Everything running in the background.

```
#include <stdio.h>
```

```

#include "Everything.h"

int main(int argc, char **argv)
{
    DWORD i;

    // Set the search string to abc
    Everything_SetSearch("abc");

    // Execute the query.
    Everything_Query(TRUE);

    // Display results.
    for(i=0;i<Everything_GetNumResults();i++)
    {
        printf("%s [%s]\n",Everything_GetResultFileName(i),Everything_GetRes

    }

    return 0;
}

```

C#

A C# example.

Note: requires Everything running in the background.

form1.Designer.cs

```

namespace WindowsApplication1
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false;
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }

            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.button1 = new System.Windows.Forms.Button();
            this.listBox1 = new System.Windows.Forms.ListBox();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.vScrollBar1 = new System.Windows.Forms.VScrollBar();

```

```

        this.SuspendLayout();
        //
        // button1
        //
        this.button1.Location = new System.Drawing.Point(250, 12);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(84, 20);
        this.button1.TabIndex = 0;
        this.button1.Text = "Search";
        this.button1.UseVisualStyleBackColor = true;
        this.button1.Click += new System.EventHandler(this.button1_C
        //
        // listBox1
        //
        this.listBox1.FormattingEnabled = true;
        this.listBox1.Location = new System.Drawing.Point(12, 38);
        this.listBox1.Name = "listBox1";
        this.listBox1.Size = new System.Drawing.Size(302, 199);
        this.listBox1.TabIndex = 2;
        //
        // textBox1
        //
        this.textBox1.Location = new System.Drawing.Point(12, 12);
        this.textBox1.Name = "textBox1";
        this.textBox1.Size = new System.Drawing.Size(232, 20);
        this.textBox1.TabIndex = 3;
        //
        // vScrollBar1
        //
        this.vScrollBar1.Location = new System.Drawing.Point(317, 38
        this.vScrollBar1.Name = "vScrollBar1";
        this.vScrollBar1.Size = new System.Drawing.Size(17, 199);
        this.vScrollBar1.TabIndex = 4;
        this.vScrollBar1.Scroll += new System.Windows.Forms.ScrollEv
        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F)
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        this.ClientSize = new System.Drawing.Size(344, 245);
        this.Controls.Add(this.vScrollBar1);
        this.Controls.Add(this.textBox1);
        this.Controls.Add(this.listBox1);
        this.Controls.Add(this.button1);
        this.Name = "Form1";
        this.Text = "Form1";
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    #endregion

    private System.Windows.Forms.Button button1;
    private System.Windows.Forms.ListBox listBox1;
    private System.Windows.Forms.TextBox textBox1;
    private System.Windows.Forms.VScrollBar vScrollBar1;

}

```

form1.cs

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        const int EVERYTHING_OK = 0;
        const int EVERYTHING_ERROR_MEMORY = 1;
        const int EVERYTHING_ERROR_IPC = 2;
        const int EVERYTHING_ERROR_REGISTERCLASSEX = 3;
        const int EVERYTHING_ERROR_CREATEWINDOW = 4;
        const int EVERYTHING_ERROR_CREATETHREAD = 5;
        const int EVERYTHING_ERROR_INVALIDINDEX = 6;
        const int EVERYTHING_ERROR_INVALIDCALL = 7;

        const int EVERYTHING_REQUEST_FILE_NAME = 0x00000001;
        const int EVERYTHING_REQUEST_PATH = 0x00000002;
        const int EVERYTHING_REQUEST_FULL_PATH_AND_FILE_NAME = 0x00000004;
        const int EVERYTHING_REQUEST_EXTENSION = 0x00000008;
        const int EVERYTHING_REQUEST_SIZE = 0x00000010;
        const int EVERYTHING_REQUEST_DATE_CREATED = 0x00000020;
        const int EVERYTHING_REQUEST_DATE_MODIFIED = 0x00000040;
        const int EVERYTHING_REQUEST_DATE_ACCESSED = 0x00000080;
        const int EVERYTHING_REQUEST_ATTRIBUTES = 0x00000100;
        const int EVERYTHING_REQUEST_FILE_LIST_FILE_NAME = 0x00000200;
        const int EVERYTHING_REQUEST_RUN_COUNT = 0x00000400;
        const int EVERYTHING_REQUEST_DATE_RUN = 0x00000800;
        const int EVERYTHING_REQUEST_DATE_RECENTLY_CHANGED = 0x00001000;
        const int EVERYTHING_REQUEST_HIGHLIGHTED_FILE_NAME = 0x00002000;
        const int EVERYTHING_REQUEST_HIGHLIGHTED_PATH = 0x00004000;
        const int EVERYTHING_REQUEST_HIGHLIGHTED_FULL_PATH_AND_FILE_NAME = 0x00008000;

        const int EVERYTHING_SORT_NAME_ASCENDING = 1;
        const int EVERYTHING_SORT_NAME_DESCENDING = 2;
        const int EVERYTHING_SORT_PATH_ASCENDING = 3;
        const int EVERYTHING_SORT_PATH_DESCENDING = 4;
        const int EVERYTHING_SORT_SIZE_ASCENDING = 5;
        const int EVERYTHING_SORT_SIZE_DESCENDING = 6;
        const int EVERYTHING_SORT_EXTENSION_ASCENDING = 7;
        const int EVERYTHING_SORT_EXTENSION_DESCENDING = 8;
        const int EVERYTHING_SORT_TYPE_NAME_ASCENDING = 9;
        const int EVERYTHING_SORT_TYPE_NAME_DESCENDING = 10;
        const int EVERYTHING_SORT_DATE_CREATED_ASCENDING = 11;
        const int EVERYTHING_SORT_DATE_CREATED_DESCENDING = 12;
        const int EVERYTHING_SORT_DATE_MODIFIED_ASCENDING = 13;
        const int EVERYTHING_SORT_DATE_MODIFIED_DESCENDING = 14;
        const int EVERYTHING_SORT_ATTRIBUTES_ASCENDING = 15;
        const int EVERYTHING_SORT_ATTRIBUTES_DESCENDING = 16;
        const int EVERYTHING_SORT_FILE_LIST_FILENAME_ASCENDING = 17;
        const int EVERYTHING_SORT_FILE_LIST_FILENAME_DESCENDING = 18;
        const int EVERYTHING_SORT_RUN_COUNT_ASCENDING = 19;
        const int EVERYTHING_SORT_RUN_COUNT_DESCENDING = 20;
        const int EVERYTHING_SORT_DATE_RECENTLY_CHANGED_ASCENDING = 21;
        const int EVERYTHING_SORT_DATE_RECENTLY_CHANGED_DESCENDING = 22;
        const int EVERYTHING_SORT_DATE_ACCESSED_ASCENDING = 23;
        const int EVERYTHING_SORT_DATE_ACCESSED_DESCENDING = 24;
        const int EVERYTHING_SORT_DATE_RUN_ASCENDING = 25;
        const int EVERYTHING_SORT_DATE_RUN_DESCENDING = 26;

        const int EVERYTHING_TARGET_MACHINE_X86 = 1;
        const int EVERYTHING_TARGET_MACHINE_X64 = 2;
        const int EVERYTHING_TARGET_MACHINE_ARM = 3;

        [DllImport("Everything64.dll", CharSet = CharSet.Unicode)]
    }
}

```

```

public static extern UInt32 Everything_SetSearchW(string lpSearchStr
[DllImport("Everything64.dll")]
public static extern void Everything_SetMatchPath(bool bEnable);
[DllImport("Everything64.dll")]
public static extern void Everything_SetMatchCase(bool bEnable);
[DllImport("Everything64.dll")]
public static extern void Everything_SetMatchWholeWord(bool bEnable);
[DllImport("Everything64.dll")]
public static extern void Everything_SetRegex(bool bEnable);
[DllImport("Everything64.dll")]
public static extern void Everything_SetMax(UInt32 dwMax);
[DllImport("Everything64.dll")]
public static extern void Everything_SetOffset(UInt32 dwOffset);

[DllImport("Everything64.dll")]
public static extern bool Everything_GetMatchPath();
[DllImport("Everything64.dll")]
public static extern bool Everything_GetMatchCase();
[DllImport("Everything64.dll")]
public static extern bool Everything_GetMatchWholeWord();
[DllImport("Everything64.dll")]
public static extern bool Everything_GetRegex();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetMax();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetOffset();
[DllImport("Everything64.dll")]
public static extern IntPtr Everything_GetSearchW();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetLastError();

[DllImport("Everything64.dll")]
public static extern bool Everything_QueryW(bool bWait);

[DllImport("Everything64.dll")]
public static extern void Everything_SortResultsByPath();

[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetNumFileResults();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetNumFolderResults();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetNumResults();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetTotFileResults();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetTotFolderResults();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetTotResults();
[DllImport("Everything64.dll")]
public static extern bool Everything_IsVolumeResult(UInt32 nIndex);
[DllImport("Everything64.dll")]
public static extern bool Everything_IsFolderResult(UInt32 nIndex);
[DllImport("Everything64.dll")]
public static extern bool Everything_IsFileResult(UInt32 nIndex);
[DllImport("Everything64.dll", CharSet = CharSet.Unicode)]
public static extern void Everything_GetResultFullPathName(UInt32 nIndex,
[DllImport("Everything64.dll")]
public static extern void Everything_Reset();

[DllImport("Everything64.dll", CharSet = CharSet.Unicode)]
public static extern IntPtr Everything_GetResultFileName(UInt32 nIndex);

// Everything 1.4
[DllImport("Everything64.dll")]
public static extern void Everything_SetSort(UInt32 dwSortType);
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetSort();
[DllImport("Everything64.dll")]

```



```

public static extern UInt32 Everything_GetResultListSort();
[DllImport("Everything64.dll")]
public static extern void Everything_SetRequestFlags(UInt32 dwReques
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetRequestFlags();
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetResultListRequestFlags();
[DllImport("Everything64.dll", CharSet = CharSet.Unicode)]
public static extern IntPtr Everything_GetResultExtension(UInt32 nIn
[DllImport("Everything64.dll")]
public static extern bool Everything_GetResultSize(UInt32 nIndex, ou
[DllImport("Everything64.dll")]
public static extern bool Everything_GetResultDateCreated(UInt32 nIn
[DllImport("Everything64.dll")]
public static extern bool Everything_GetResultDateModified(UInt32 nI
[DllImport("Everything64.dll")]
public static extern bool Everything_GetResultDateAccessed(UInt32 nI
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetResultAttributes(UInt32 nI
[DllImport("Everything64.dll", CharSet = CharSet.Unicode)]
public static extern IntPtr Everything_GetResultFileListFileName(UIIn
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetResultRunCount(UInt32 nInC
[DllImport("Everything64.dll")]
public static extern bool Everything_GetResultDateRun(UInt32 nIndex,
[DllImport("Everything64.dll")]
public static extern bool Everything_GetResultDateRecentlyChanged(UI
[DllImport("Everything64.dll", CharSet = CharSet.Unicode)]
public static extern IntPtr Everything_GetResultHighlightedFileName(
[DllImport("Everything64.dll", CharSet = CharSet.Unicode)]
public static extern IntPtr Everything_GetResultHighlightedPath(UInt
[DllImport("Everything64.dll", CharSet = CharSet.Unicode)]
public static extern IntPtr Everything_GetResultHighlightedFullPathA
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_GetRunCountFromFileName(string
[DllImport("Everything64.dll")]
public static extern bool Everything_SetRunCountFromFileName(string
[DllImport("Everything64.dll")]
public static extern UInt32 Everything_IncRunCountFromFileName(string

public Form1()
{
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
    UInt32 i;

    // set the search
    Everything_SetSearchW(textBox1.Text);

    // use our own custom scrollbar...
    // Everything_SetMax(listBox1.ClientRectangle.Height / listB
    // Everything_SetOffset(VerticalScrollBarPosition...);

    // request name and size
    Everything_SetRequestFlags(EVERYTHING_REQUEST_FILE_NAME | EV

    Everything_SetSort(13);

    // execute the query
    Everything_QueryW(true);

    // sort by path
    // Everything_SortResultsByPath();

    // clear the old list of results
    listBox1.Items.Clear();

```

```

        // set the window title
        Text = textBox1.Text + " - " + Everything_GetNumResults() +

        // loop through the results, adding each result to the list
        for (i = 0; i < Everything_GetNumResults(); i++)
        {
            long date_modified;
            long size;

            Everything_GetResultDateModified(i, out date_modified);
            Everything_GetResultSize(i, out size);

            // add it to the list box
            listBox1.Items.Insert((int)i, "Size: " + size.ToStri

        }
    }
}

```

Clarion

Note: requires Everything running in the background.

Setup

Download the Clarion module: [Clarion_Everything.zip](#)

Source code of the module is included.

Extract Everything.lib to your clarion project.

Add the LIB to "Library, object, and resource files" in your Clarion project and put the following in your global map.

```

MODULE('Everything.lib')
    Everything_GetLastError(), LONG, RAW, PASCAL
    Everything_SetSearch(*CSTRING), RAW, PASCAL
    Everything_SetMatchPath(LONG), RAW, PASCAL
    Everything_SetMatchCase(LONG), RAW, PASCAL
    Everything_SetMatchWholeWord(LONG), RAW, PASCAL
    Everything_EnableRegex(LONG), RAW, PASCAL
    Everything_SetMax(LONG), RAW, PASCAL
    Everything_SetOffset(LONG), RAW, PASCAL
    Everything_Query(), RAW, PASCAL
    Everything_SetDetail(LONG), RAW, PASCAL
    Everything_SortResults(LONG), RAW, PASCAL
    Everything_GetNumFileResults(), LONG, RAW, PASCAL
    Everything_GetNumFolderResults(), LONG, RAW, PASCAL
    Everything_GetNumResults(), LONG, RAW, PASCAL
    Everything_GetTotFileResults(), LONG, RAW, PASCAL
    Everything_GetTotFolderResults(), LONG, RAW, PASCAL
    Everything_GetTotResults(), LONG, RAW, PASCAL
    Everything_IsVolumeResult(LONG), LONG, RAW, PASCAL
    Everything_IsFolderResult(LONG), LONG, RAW, PASCAL
    Everything_IsFileResult(LONG), LONG, RAW, PASCAL
    Everything_GetResultFileName(LONG), *CSTRING, RAW, PASCAL
    Everything_GetResultPath(LONG), *CSTRING, RAW, PASCAL
    Everything_GetResult(LONG, LONG), RAW, PASCAL
END

```

Changes

added:

- [Everything_SetDetail](#)
- [Everything_SortResults](#)

Equates

```
!return values from Everything_GetLastError
EVERYTHING_OK                                EQUATE(0) ! no error
EVERYTHING_ERROR_MEMORY                      EQUATE(1) ! out of memory
EVERYTHING_ERROR_IPC                         EQUATE(2) ! IPC not available
EVERYTHING_ERROR_REGISTERCLASSEX             EQUATE(3) ! RegisterClassEx failed
EVERYTHING_ERROR_CREATEWINDOW                EQUATE(4) ! CreateWindow failed
EVERYTHING_ERROR_CREATETHREAD                 EQUATE(5) ! CreateThread failed
EVERYTHING_ERROR_INVALIDINDEX                EQUATE(6) ! Invalid result index (must be >=0)
EVERYTHING_ERROR_INVALIDCALL                 EQUATE(7) ! Call Everything_Query before get

!attributes
EVERYTHING_FILE_ATTRIBUTE_READONLY            EQUATE(0001h)
EVERYTHING_FILE_ATTRIBUTE_HIDDEN              EQUATE(0002h)
EVERYTHING_FILE_ATTRIBUTE_SYSTEM              EQUATE(0004h)
EVERYTHING_FILE_ATTRIBUTE_DIRECTORY           EQUATE(0010h)
EVERYTHING_FILE_ATTRIBUTE_ARCHIVE             EQUATE(0020h)
EVERYTHING_FILE_ATTRIBUTE_DEVICE              EQUATE(0040h)
EVERYTHING_FILE_ATTRIBUTE_NORMAL              EQUATE(0080h)
EVERYTHING_FILE_ATTRIBUTE_TEMPORARY           EQUATE(0100h)
EVERYTHING_FILE_ATTRIBUTE_SPARSE_FILE         EQUATE(0200h)
EVERYTHING_FILE_ATTRIBUTE_REPARSE_POINT       EQUATE(0400h)
EVERYTHING_FILE_ATTRIBUTE_COMPRESSED          EQUATE(0800h)
EVERYTHING_FILE_ATTRIBUTE_OFFLINE             EQUATE(1000h)
EVERYTHING_FILE_ATTRIBUTE_NOT_CONTENT_INDEXED EQUATE(2000h)
EVERYTHING_FILE_ATTRIBUTE_ENCRYPTED            EQUATE(4000h)
```

Limits and Notes

Everything **MUST** be running for this to work.

See Everything [IPC](#).

You can only do one search at a time.

If there is demand for it, I can add multiple thread safe searches.

Descriptions

Everything_GetLastError

```
myError LONG

!check the equates for error message descriptions
myError = Everything_GetLastError()
```

Everything_SetSearch

```
myString STRING

!Set the search string
!default search is '' (empty string)
Everything_SetSearch(myString)
```

Everything_SetMatchPath

```
val LONG

!set val to zero to disable path matching
!set val to non-zero to enable path matching
!default is 0 (off)
Everything_SetMatchPath(val)
```

Everything_SetMatchCase

```
val LONG

!set val to zero to disable case matching
!set val to non-zero to enable case matching
!default is 0 (off)
Everything_SetMatchCase(val)
```

Everything_SetMatchWholeWord

```
val LONG

!set val to zero to disable whole word matching
!set val to non-zero to enable whole word matching
!default is 0 (off)
Everything_SetMatchWholeWord(val)
```

Everything_EnableRegex

```
val LONG

!set val to zero to disable regex
!set val to non-zero to enable regex
!default is 0 (off)
Everything_EnableRegex(val)
```

Everything_SetMax

val LONG

!If you display the results in a list, set this value to how many items are visible
!How many results would you like Everything to return?
!Use this for displaying small lists.
!Default is 0xFFFFFFFF (all results)
Everything_SetMax(val)

Everything_SetOffset

offset LONG

!If you display a list with a scroll bar, set the offset to the scroll bar's offset.
!From what index would you like Everything to return results from?
!zero means return results starting from the first result.
!10 would mean return results starting from the 10th result.
!use this for displaying small lists.
!default is 0
!This is 0 based
Everything_SetOffset(offset)

Everything_Query

!Execute the query
!use the Everything_Set* functions to override default search parameters
!This sends the search options to "Everything" over the IPC.
!It does not return until it has a copy of ALL the results.
!Use Everything_GetResult to enumerate results.
!100 results will take about 1 millisecond.
!1 million results will take about 1 second.
Everything_Query()

Everything_SetDetail

detail LONG

!The result detail
!Set detail to nonzero to get file information from Everything_GetResult
!Set detail to zero to get file name, path name offsets and lengths and partial file
!The default detail is zero.
!This can be called at anytime.
!make sure you sort the results with Everything_SortResults() if you set the detail
Everything_SetDetail(detail)

Everything_SortResults

NoDriveLetter LONG

```
!Set NoDriveLetter to nonzero to sort by path excluding drive letters.  
!Set NoDriveLetter to zero to sort by path.  
!Use this before getting detailed results (set NoDriveLetter to zero).  
!Results are sorted by name by default.  
!Can only be called after a call to Everything_Query  
Everything_SortResults(NoDriveLetter)
```

Everything_GetNumFileResults

```
numFiles LONG  
  
!Get the number of files in the visible results list.  
numFiles = Everything_GetNumFileResults()
```

Everything_GetNumFolderResults

```
numFolders LONG  
  
!Get the number of folders in the visible results list.  
numFolders = Everything_GetNumFolderResults()
```

Everything_GetNumResults

```
numResults LONG  
  
!Get the number of files and folders in the visible results list.  
numResults = Everything_GetNumResults()
```

Everything_GetTotFileResults

```
totFiles LONG  
  
!Get the total number of files in the results list.  
totFiles = Everything_GetTotFileResults()
```

Everything_GetTotFolderResults

```
totFolders LONG  
  
!Get the total number of files in the results list.  
totFolders = Everything_GetTotFolderResults()
```

Everything_GetTotResults

```
totResults LONG

!Get the total number of files and folders in the results list.
!index is 0 based
totResults = Everything_GetTotResults()
```

Everything_IsVolumeResult

```
index LONG
val LONG

!returns zero if the result is not a volume.
!returns non-zero if the result is a volume.
!index is 0 based
!deprecated - Use Everything_GetResult
val = Everything_IsVolumeResult(index)
```

Everything_IsFolderResult

```
index LONG
val LONG

!returns zero if the result is not a folder.
!returns non-zero if the result is a folder.
!index is 0 based
!deprecated - Use Everything_GetResult
val = Everything_IsFolderResult(index)
```

Everything_IsFileResult

```
index LONG
val LONG

!returns zero if the result is not a file
!returns non-zero if the result is a file.
!index is 0 based
!deprecated - Use Everything_GetResult
val = Everything_IsFileResult(index)
```

Everything_GetResultFileName

```
index LONG
filename CSTRING

!index is 0 based
!this gets the result file name
!deprecated - Use Everything_GetResult
```

```
filename = Everything_GetResultFileName(index)
```

Everything_GetResultPath

```
index LONG
path CSTRING

!index is 0 based
!this gets the result path
!deprecated - Use Everything_GetResult
path = Everything_GetResultPath(index)
```

Everything_GetResult

```
index LONG
result GROUP
FullPathFileName      STRING(260) !The full path and file name of the result
PathPartLength        LONG ! Length of the path part only
FileNamePartOffset    LONG ! Offset into FullPathFileName for the file name part c
SizeHigh              LONG ! High part of size (ignore this if you do not need the
SizeLow               ULONG ! Low part of size
Attributes             LONG ! Attribute flags can be zero or more EVERYTHING_FILE_A
CreationYear          LONG ! The year (1601 - 30827).
CreationMonth          LONG ! The month (January = 1, December = 12)
CreationDayOfWeek      LONG ! The day of the week. (sunday = 0, monday = 1, Saturda
CreationDay            LONG ! The day of the month (1-31).
CreationHour           LONG ! The hour (0-23).
CreationMinute         LONG ! The minute (0-59).
CreationSecond         LONG ! The second (0-59).
CreationMilliseconds   LONG ! The millisecond (0-999).
LastAccessYear         LONG ! The year (1601 - 30827).
LastAccessMonth        LONG ! The month (January = 1, December = 12)
LastAccessDayOfWeek    LONG ! The day of the week. (sunday = 0, monday = 1, Saturda
LastAccessDay          LONG ! The day of the month (1-31).
LastAccessHour         LONG ! The hour (0-23).
LastAccessMinute       LONG ! The minute (0-59).
LastAccessSecond       LONG ! The second (0-59).
LastAccessMilliseconds LONG ! The millisecond (0-999).
LastWriteYear          LONG ! The year (1601 - 30827).
LastWriteMonth         LONG ! The month (January = 1, December = 12)
LastWriteDayOfWeek      LONG ! The day of the week. (sunday = 0, monday = 1, Saturda
LastWriteDay           LONG ! The day of the month (1-31).
LastWriteHour          LONG ! The hour (0-23).
LastWriteMinute        LONG ! The minute (0-59).
LastWriteSecond        LONG ! The second (0-59).
LastWriteMilliseconds  LONG ! The millisecond (0-999).
    END

!index is 0 based
!fills in the result group
!See Everything_SetDetail for what information is filled in.
Everything_GetResult(index,ADDRESS(result))
```

Fast Program Example


```

x LONG,AUTO
m LONG
result GROUP
FullPathFileName      STRING(260) !The full path and file name of the result
PathPartLength        LONG ! Length of the path part only
FileNamePartOffset    LONG ! Offset into FullPathFileName for the file name part c
SizeHigh              LONG ! Not used in this example.
SizeLow               ULONG ! Not used in this example.
Attributes             LONG ! Attribute flags can be zero or EVERYTHING_FILE_ATTRIE
CreationYear          LONG ! Not used in this example.
CreationMonth         LONG ! Not used in this example.
CreationDayOfWeek     LONG ! Not used in this example.
CreationDay           LONG ! Not used in this example.
CreationHour          LONG ! Not used in this example.
CreationMinute        LONG ! Not used in this example.
CreationSecond        LONG ! Not used in this example.
CreationMilliseconds  LONG ! Not used in this example.
LastAccessYear        LONG ! Not used in this example.
LastAccessMonth       LONG ! Not used in this example.
LastAccessDayOfWeek   LONG ! Not used in this example.
LastAccessDay         LONG ! Not used in this example.
LastAccessHour        LONG ! Not used in this example.
LastAccessMinute      LONG ! Not used in this example.
LastAccessSecond      LONG ! Not used in this example.
LastAccessMilliseconds LONG ! Not used in this example.
LastWriteYear         LONG ! Not used in this example.
LastWriteMonth        LONG ! Not used in this example.
LastWriteDayOfWeek    LONG ! Not used in this example.
LastWriteDay          LONG ! Not used in this example.
LastWriteHour         LONG ! Not used in this example.
LastWriteMinute       LONG ! Not used in this example.
LastWriteSecond       LONG ! Not used in this example.
LastWriteMilliseconds LONG ! Not used in this example.
END

!setup a query using the default search parameters
!change the search string to 'whatthe'
Everything_SetSearch('whatthe')

!Find all the results for 'whatthe'
Everything_Query()

m = Everything_GetNumResults()
LOOP x = 0 TO m - 1

    Everything_GetResult(x,ADDRESS(result))

    ...
    !do something with name and path
    ...
END

```

Detailed Program Example

```

x LONG,AUTO
m LONG
result GROUP
FullPathFileName      STRING(260) !The full path and file name of the result
PathPartLength        LONG ! Length of the path part only
FileNamePartOffset    LONG ! Offset into FullPathFileName for the file name part c
SizeHigh              LONG ! High part of size (ignore this if you do not need the
SizeLow               ULONG ! Low part of size
Attributes             LONG ! Attribute flags can be zero or more EVERYTHING_FILE_A

```

```

CreationYear          LONG ! The year (1601 - 30827).
CreationMonth         LONG ! The month (January = 1, December = 12)
CreationDayOfWeek     LONG ! The day of the week. (sunday = 0, monday = 1, Saturda
CreationDay          LONG ! The day of the month (1-31).
CreationHour          LONG ! The hour (0-23).
CreationMinute        LONG ! The minute (0-59).
CreationSecond        LONG ! The second (0-59).
CreationMilliseconds  LONG ! The millisecond (0-999).
LastAccessYear        LONG ! The year (1601 - 30827).
LastAccessMonth       LONG ! The month (January = 1, December = 12)
LastAccessDayOfWeek   LONG ! The day of the week. (sunday = 0, monday = 1, Saturda
LastAccessDay         LONG ! The day of the month (1-31).
LastAccessHour        LONG ! The hour (0-23).
LastAccessMinute      LONG ! The minute (0-59).
LastAccessSecond      LONG ! The second (0-59).
LastAccessMilliseconds LONG ! The millisecond (0-999).
LastWriteYear         LONG ! The year (1601 - 30827).
LastWriteMonth        LONG ! The month (January = 1, December = 12)
LastWriteDayOfWeek    LONG ! The day of the week. (sunday = 0, monday = 1, Saturda
LastWriteDay          LONG ! The day of the month (1-31).
LastWriteHour         LONG ! The hour (0-23).
LastWriteMinute       LONG ! The minute (0-59).
LastWriteSecond       LONG ! The second (0-59).
LastWriteMilliseconds LONG ! The millisecond (0-999).
        END

!setup a query using the default search parameters
!change the search string to 'whatthe'
Everything_SetSearch('whatthe')

!Find all the results for 'whatthe'
Everything_Query()

!turn detail on
Everything_SetDetail(1)

!Sort the results
!this is required for nonzero detail
Everything_SortResults(0)

m = Everything_GetNumResults()
LOOP x = 0 TO m - 1
    Everything_GetResult(x, ADDRESS(result))
    ...
    !do something with result
    ...
END

```

Everything_CleanUp

The **Everything_CleanUp** function frees any allocated memory by the library.

Syntax

```
void Everything_CleanUp(void);
```

Parameters

This function has no parameters.

Return Value

This function has no return value.

Remarks

You should call **Everything_CleanUp** to free any memory allocated by the Everything SDK.

Everything_CleanUp should be the last call to the Everything SDK.

Call [Everything_Reset](#) to free any allocated memory for the current search and results.

[Everything_Reset](#) will also reset the search and result state to their defaults.

Calling [Everything_SetSearch](#) frees the old search and allocates the new search string.

Calling [Everything_Query](#) frees the old result list and allocates the new result list.

Example

```
Everything_CleanUp();
```

See Also

- [Everything_SetSearch](#)
- [Everything_Reset](#)
- [Everything_Query](#)

Everything_DeleteRunHistory

The **Everything_DeleteRunHistory** function deletes all run history.

Syntax

```
BOOL Everything_DeleteRunHistory(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if run history is cleared.

The function returns 0 if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
------------	---------

Remarks

Calling this function will clear all run history from memory and disk.

Example

```
// clear run history
Everything_DeleteRunHistory();
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_GetRunCountFromFileName](#)
- [Everything_SetRunCountFromFileName](#)
- [Everything_IncRunCountFromFileName](#)

Everything_Exit

The **Everything_Exit** function requests Everything to exit.

Syntax

```
BOOL Everything_Exit(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if the exit request was successful.

The function returns 0 if the request failed. To get extended error information, call [Everything_GetLastError](#)

Error code

Meaning

Remarks

Request Everything to save settings and data to disk and exit.

Example

```
// request Everything to exit.
Everything_Exit();
```

Function Information

Requires Everything 1.4.1 or later.

See Also

Everything_GetBuildNumber

The **Everything_GetBuildNumber** function retrieves the build number of Everything.

Syntax

```
DWORD Everything_GetBuildNumber(void);
```

Parameters

No parameters.

Return Value

The function returns the build number.

The function returns 0 if build information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code

Meaning

Remarks

Everything uses the following version format:

major.minor.revision.build

The build part is incremental and unique for all Everything versions.

Example

```
DWORD buildNumber;

// get the build number
buildNumber = Everything_GetBuildNumber();

if (buildNumber >= 686)
{
    // do something with build 686 or later.
}
```

Function Information

Requires Everything 1.0.0.0 or later.

See Also

- [Everything_GetMajorVersion](#)
- [Everything_GetMinorVersion](#)
- [Everything_GetRevision](#)
- [Everything_GetTargetMachine](#)

Everything_GetLastError

The **Everything_GetLastError** function retrieves the last-error code value.

Syntax

```
DWORD Everything_GetLastError(void);
```

Parameters

This function has no parameters.

Return Value

Error code	Value	Meaning
EVERYTHING_OK	0	The operation completed successfully.
EVERYTHING_ERROR_MEMORY	1	Failed to allocate memory for the search query.
EVERYTHING_ERROR_IPC	2	IPC is not available.
EVERYTHING_ERROR_REGISTERCLASSEX	3	Failed to register the search query window class.
EVERYTHING_ERROR_CREATEWINDOW	4	Failed to create the search query window.
EVERYTHING_ERROR_CREATETHREAD	5	Failed to create the search query thread.
EVERYTHING_ERROR_INVALIDINDEX	6	Invalid index. The index must be greater or equal to 0 and less than the number of visible results.
EVERYTHING_ERROR_INVALIDCALL	7	Invalid call.

Example

```
// execute the query
if (!Everything_Query(true))
{
    DWORD dwLastError = Everything_GetLastError();
    if (dwLastError == EVERYTHING_ERROR_IPC)
    {
        // IPC not available.
    }
}
```

Everything_GetMajorVersion

The **Everything_GetMajorVersion** function retrieves the major version number of Everything.

Syntax

```
DWORD Everything_GetMajorVersion(void);
```

Parameters

No parameters.

Return Value

The function returns the major version number.

The function returns 0 if major version information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Everything uses the following version format:

major.minor.revision.build

The build part is incremental and unique for all Everything versions.

Example

```
DWORD majorVersion;
DWORD minorVersion;
DWORD revision;

// get version information
majorVersion = Everything_GetMajorVersion();
minorVersion = Everything_GetMinorVersion();
revision = Everything_GetRevision();

if ((majorVersion > 1) || ((majorVersion == 1) && (minorVersion > 3)) || ((majorVersion == 1) && (minorVersion == 3) && (revision > 0)))
{
    // do something with version 1.3.4 or later
}
```

Function Information

Requires Everything 1.0.0.0 or later.

See Also

- [Everything_GetMinorVersion](#)
- [Everything_GetRevision](#)
- [Everything_GetBuildNumber](#)
- [Everything_GetTargetMachine](#)

Everything_GetMatchCase

The **Everything_GetMatchCase** function returns the match case state.

Syntax


```
BOOL Everything_GetMatchCase(void);
```

Return Value

The return value is the match case state.

The function returns TRUE if match case is enabled.

The function returns FALSE if match case is disabled.

Remarks

Get the internal state of the match case switch.

The default state is FALSE, or disabled.

Example

```
BOOL bEnabled = Everything_GetMatchCase();
```

See Also

- [Everything_SetMatchCase](#)
- [Everything_Query](#)

Everything_GetMatchPath

The **Everything_GetMatchPath** function returns the state of the match full path switch.

Syntax

```
BOOL Everything_GetMatchPath(void);
```

Return Value

The return value is the state of the match full path switch.

The function returns TRUE if match full path is enabled.

The function returns FALSE if match full path is disabled.

Remarks

Get the internal state of the match full path switch.

The default state is FALSE, or disabled.

Example

```
BOOL bEnabled = Everything_GetMatchPath();
```

See Also

- [Everything_SetMatchPath](#)
- [Everything_Query](#)

Everything_GetMatchWholeWord

The **Everything_GetMatchWholeWord** function returns the match whole word state.

Syntax

```
BOOL Everything_GetMatchWholeWord(void);
```

Return Value

The return value is the match whole word state.

The function returns TRUE if match whole word is enabled.

The function returns FALSE if match whole word is disabled.

Remarks

The default state is FALSE, or disabled.

Example

```
BOOL bEnabled = Everything_GetMatchWholeWord();
```

See Also

- [Everything_SetMatchWholeWord](#)
- [Everything_Query](#)

Everything_GetMax

The **Everything_GetMax** function returns the maximum number of results state.

Syntax

```
BOOL Everything_GetMax(void);
```

Return Value

The return value is the maximum number of results state.

The function returns 0xFFFFFFFF if all results should be returned.

Remarks

The default state is 0xFFFFFFFF, or all results.

Example

```
DWORD dwMaxResults = Everything_GetMax();
```

See Also

- [Everything_SetMax](#)
- [Everything_Query](#)

Everything_GetMinorVersion

The **Everything_GetMinorVersion** function retrieves the minor version number of Everything.

Syntax

```
DWORD Everything_GetMinorVersion(void);
```

Parameters

No parameters.

Return Value

The function returns the minor version number.

The function returns 0 if minor version information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Everything uses the following version format:

major.minor.revision.build

The build part is incremental and unique for all Everything versions.

Example

```
DWORD majorVersion;
DWORD minorVersion;
DWORD revision;

// get version information
majorVersion = Everything_GetMajorVersion();
minorVersion = Everything_GetMinorVersion();
revision = Everything_GetRevision();

if ((majorVersion > 1) || ((majorVersion == 1) && (minorVersion > 3)) || ((majorVersion == 1) && (minorVersion == 3) && (revision > 0)))
{
    // do something with version 1.3.4 or later
}
```

Function Information

Requires Everything 1.0.0.0 or later.

See Also

- [Everything_GetMajorVersion](#)
- [Everything_GetRevision](#)
- [Everything_GetBuildNumber](#)
- [Everything_GetTargetMachine](#)

Everything_GetNumFileResults

The **Everything_GetNumFileResults** function returns the number of visible file results.

Syntax

```
DWORD Everything_GetNumFileResults(void);
```

Parameters

This function has no parameters.

Return Value

Returns the number of visible file results.

If the function fails the return value is 0. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetNumFileResults .

Remarks

You must call [Everything_Query](#) before calling [Everything_GetNumFileResults](#).

Use [Everything_GetTotFileResults](#) to retrieve the total number of file results.

If the result offset state is 0, and the max result is 0xFFFFFFFF, [Everything_GetNumFileResults](#) will return the total number of file results and all file results will be visible.

[Everything_GetNumFileResults](#) is not supported when using [Everything_SetRequestFlags](#)

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// get the number of visible file results.
DWORD dwNumFileResults = Everything_GetNumFileResults();
```

See Also

- [Everything_Query](#)
- [Everything_GetNumResults](#)
- [Everything_GetNumFolderResults](#)
- [Everything_GetTotResults](#)
- [Everything_GetTotFileResults](#)
- [Everything_GetTotFolderResults](#)

Everything_GetNumFolderResults

The **Everything_GetNumFolderResults** function returns the number of visible folder results.

Syntax

```
DWORD Everything_GetNumFolderResults(void);
```

Parameters

This functions has no parameters.

Return Value

Returns the number of visible folder results.

If the function fails the return value is 0. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetNumFolderResults.

Remarks

You must call [Everything_Query](#) before calling Everything_GetNumFolderResults.

Use [Everything_GetTotFolderResults](#) to retrieve the total number of folder results.

If the result offset state is 0, and the max result is 0xFFFFFFFF, Everything_GetNumFolderResults will return the total number of folder results and all folder results will be visible.

Everything_GetNumFolderResults is not supported when using [Everything_SetRequestFlags](#)

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// get the number of visible folder results
DWORD dwNumFolderResults = Everything_GetNumFolderResults();
```

See Also

- [Everything_Query](#)
- [Everything_GetNumResults](#)
- [Everything_GetNumFileResults](#)
- [Everything_GetTotResults](#)
- [Everything_GetTotFileResults](#)
- [Everything_GetTotFolderResults](#)

Everything_GetNumResults

The **Everything_GetNumResults** function returns the number of visible file and folder results.

Syntax

```
DWORD Everything_GetNumResults(void);
```

Parameters

This functions has no parameters.

Return Value

Returns the number of visible file and folder results.

If the function fails the return value is 0. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetNumResults.

Remarks

You must call [Everything_Query](#) before calling Everything_GetNumResults.

Use [Everything_GetTotResults](#) to retrieve the total number of file and folder results.

If the result offset state is 0, and the max result is 0xFFFFFFFF, Everything_GetNumResults will return the total number of file and folder results and all file and folder results will be visible.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// get the number of visible file and folder results.
DWORD dwNumResults = Everything_GetNumResults();
```

See Also

- [Everything_Query](#)
- [Everything_GetNumFileResults](#)
- [Everything_GetNumFolderResults](#)
- [Everything_GetTotResults](#)
- [Everything_GetTotFileResults](#)

- [Everything_GetTotFolderResults](#)

Everything_GetOffset

The **Everything_GetOffset** function returns the first item offset of the available results.

Syntax

```
DWORD Everything_GetOffset(void);
```

Return Value

The return value is the first item offset.

Remarks

The default offset is 0.

Example

```
DWORD dwOffset = Everything_GetOffset();
```

See Also

- [Everything_SetOffset](#)
- [Everything_Query](#)

Everything_GetRegex

The **Everything_GetRegex** function returns the regex state.

Syntax

```
BOOL Everything_GetRegex(void);
```

Return Value

The return value is the regex state.

The function returns TRUE if regex is enabled.

The function returns FALSE if regex is disabled.

Remarks

The default state is FALSE.

Example

```
BOOL bRegex = Everything_GetRegex();
```

See Also

- [Everything_SetRegex](#)
- [Everything_Query](#)

Everything_GetReplyID

The **Everything_GetReplyID** function returns the current reply identifier for the IPC query reply.

Syntax

```
DWORD Everything_GetReplyID(void);
```

Return Value

The return value is the current reply identifier.

Remarks

The default reply identifier is 0.

Example

```
DWORD id = Everything_GetReplyID();
```

See Also

- [Everything_SetReplyWindow](#)
- [Everything_GetReplyWindow](#)
- [Everything_SetReplyID](#)
- [Everything_Query](#)
- [Everything_IsQueryReply](#)

Everything_GetReplyWindow

The **Everything_GetReplyWindow** function returns the current reply window for the IPC query reply.

Syntax

```
HWND Everything_GetReplyWindow(void);
```

Return Value

The return value is the current reply window.

Remarks

The default reply window is 0, or no reply window.

Example

```
HWND hWnd = Everything_GetReplyWindow();
```

See Also

- [Everything_SetReplyWindow](#)
- [Everything_SetReplyID](#)
- [Everything_GetReplyID](#)
- [Everything_Query](#)
- [Everything_IsQueryReply](#)

Everything_GetRequestFlags

The **Everything_GetRequestFlags** function returns the desired result data flags.

Syntax

```
DWORD Everything_GetRequestFlags(void);
```

Return Value

Returns zero or more of the following request flags:

EVERYTHING_REQUEST_FILE_NAME	(0x00000001)
EVERYTHING_REQUEST_PATH	(0x00000002)
EVERYTHING_REQUEST_FULL_PATH_AND_FILE_NAME	(0x00000004)
EVERYTHING_REQUEST_EXTENSION	(0x00000008)
EVERYTHING_REQUEST_SIZE	(0x00000010)
EVERYTHING_REQUEST_DATE_CREATED	(0x00000020)
EVERYTHING_REQUEST_DATE_MODIFIED	(0x00000040)
EVERYTHING_REQUEST_DATE_ACCESSED	(0x00000080)
EVERYTHING_REQUEST_ATTRIBUTES	(0x00000100)
EVERYTHING_REQUEST_FILE_LIST_FILE_NAME	(0x00000200)
EVERYTHING_REQUEST_RUN_COUNT	(0x00000400)
EVERYTHING_REQUEST_DATE_RUN	(0x00000800)
EVERYTHING_REQUEST_DATE_RECENTLY_CHANGED	(0x00001000)
EVERYTHING_REQUEST_HIGHLIGHTED_FILE_NAME	(0x00002000)
EVERYTHING_REQUEST_HIGHLIGHTED_PATH	(0x00004000)
EVERYTHING_REQUEST_HIGHLIGHTED_FULL_PATH_AND_FILE_NAME	(0x00008000)

Remarks

The default request flags are `EVERYTHING_REQUEST_FILE_NAME | EVERYTHING_REQUEST_PATH` (0x00000003)

Example

```
DWORD dwRequestFlags = Everything_GetRequestFlags();
```

See Also

- [Everything_SetRequestFlags](#)
- [Everything_Query](#)

Everything_GetResultAttributes

The **Everything_GetResultAttributes** function retrieves the attributes of a visible result.

Syntax

```
DWORD Everything_GetResultAttributes(
    DWORD dwIndex,
);
```

Parameters

dwIndex

Zero based index of the visible result.

Return Value

The function returns zero or more of FILE_ATTRIBUTE_* flags.

The function returns INVALID_FILE_ATTRIBUTES if attribute information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultAttributes.
EVERYTHING_ERROR_INVALIDREQUEST	Attribute information was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_ATTRIBUTES before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

dwIndex must be a valid visible result index. To determine if a result index is visible use the [Everything_GetNumResults](#) function.

Example

```
DWORD attributes;

// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the attributes of the first visible result.
attributes = Everything_GetResultAttributes(0);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultDateAccessed

The **Everything_GetResultDateAccessed** function retrieves the accessed date of a visible result.

Syntax

```
BOOL Everything_GetResultDateAccessed(  
    DWORD dwIndex,  
    FILETIME *lpDateAccessed  
);
```

Parameters

dwIndex

Zero based index of the visible result.

lpDateAccessed

Pointer to a FILETIME to hold the accessed date of the result.

Return Value

The function returns non-zero if successful.

The function returns 0 if the accessed date information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultDateAccessed.
EVERYTHING_ERROR_INVALIDREQUEST	Accessed date was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_DATE_ACCESSED before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

dwIndex must be a valid visible result index. To determine if a result index is visible use the [Everything_GetNumResults](#) function.

Example

```
FILETIME dateAccessed;  
  
// set the search text to abc AND 123  
Everything_SetSearch("abc 123");  
  
// execute the query  
Everything_Query(TRUE);
```

```
// Get the accessed date of the first visible result.  
Everything_GetResultDateAccessed(0,&dateAccessed);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultDateCreated

The **Everything_GetResultDateCreated** function retrieves the created date of a visible result.

Syntax

```
BOOL Everything_GetResultDateCreated(  
    DWORD dwIndex,  
    FILETIME *lpDateCreated  
);
```

Parameters

dwIndex

Zero based index of the visible result.

lpDateCreated

Pointer to a FILETIME to hold the created date of the result.

Return Value

The function returns non-zero if successful.

The function returns 0 if the date created information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultDateCreated.

EVERYTHING_ERROR_INVALIDREQUEST	Date created was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_DATE_CREATED before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

dwIndex must be a valid visible result index. To determine if a result index is visible use the [Everything_GetNumResults](#) function.

Example

```
FILETIME dateCreated;

// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the created date of the first visible result.
Everything_GetResultDateCreated(0, &dateCreated);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultDateModified

The **Everything_GetResultDateModified** function retrieves the modified date of a visible result.

Syntax

```
BOOL Everything_GetResultDateModified(
    DWORD dwIndex,
    FILETIME *lpDateModified
);
```

Parameters

dwIndex

Zero based index of the visible result.

lpDateModified

Pointer to a FILETIME to hold the modified date of the result.

Return Value

The function returns non-zero if successful.

The function returns 0 if the modified date information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultDateModified.
EVERYTHING_ERROR_INVALIDREQUEST	Modified date was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_DATE_MODIFIED before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

dwIndex must be a valid visible result index. To determine if a result index is visible use the [Everything_GetNumResults](#) function.

Example

```
FILETIME dateModified;

// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the modified date of the first visible result.
Everything_GetResultDateModified(0,&dateModified);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultDateRecentlyChanged

The **Everything_GetResultDateRecentlyChanged** function retrieves the recently changed date of a visible result.

Syntax

```
BOOL Everything_GetResultDateRecentlyChanged(  
    DWORD dwIndex,  
    FILETIME *lpDateRecentlyChanged  
);
```

Parameters

dwIndex

Zero based index of the visible result.

lpDateRecentlyChanged

Pointer to a FILETIME to hold the recently changed date of the result.

Return Value

The function returns non-zero if successful.

The function returns 0 if the recently changed date information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultDateRecentlyChanged.
EVERYTHING_ERROR_INVALIDREQUEST	Recently changed date was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_DATE_RECENTLY_CHANGED before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

dwIndex must be a valid visible result index. To determine if a result index is visible use the [Everything_GetNumResults](#) function.

The date recently changed is the date and time of when the result was changed in the Everything index, this could

be from a file creation, rename, attribute or content change.

Example

```
FILETIME dateRecentlyChanged;

// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the recently changed date of the first visible result.
Everything_GetResultDateRecentlyChanged(0, &dateRecentlyChanged);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultDateRun

The **Everything_GetResultDateRun** function retrieves the run date of a visible result.

Syntax

```
BOOL Everything_GetResultDateRun(
    DWORD dwIndex,
    FILETIME *lpDateRun
);
```

Parameters

dwIndex

Zero based index of the visible result.

lpDateRun

Pointer to a FILETIME to hold the Run date of the result.

Return Value

The function returns non-zero if successful.

The function returns 0 if the run date information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultDateRun.
EVERYTHING_ERROR_INVALIDREQUEST	Run date was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_DATE_RUN before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

dwIndex must be a valid visible result index. To determine if a result index is visible use the [Everything_GetNumResults](#) function.

Example

```
FILETIME dateRun;

// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the run date of the first visible result.
Everything_GetResultDateRun(0,&dateRun);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultExtension

The **Everything_GetResultExtension** function retrieves the extension part of a visible result.

Syntax

```
LPCTSTR Everything_GetResultExtension(  
    DWORD dwIndex  
);
```

Parameters

dwIndex

Zero based index of the visible result.

Return Value

The function returns a pointer to a null terminated string of **TCHARs**.

If the function fails the return value is NULL. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultExtension .
EVERYTHING_ERROR_INVALIDREQUEST	Extension was not requested or is unavailable, Call Everything_SetRequestFlags with <code>EVERYTHING_REQUEST_EXTENSION</code> before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

The ANSI / Unicode version of this function must match the ANSI / Unicode version of the call to [Everything_Query](#).

The function returns a pointer to an internal structure that is only valid until the next call to [Everything_Query](#), [Everything_Reset](#) or [Everything_CleanUp](#).

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumResults](#) function.

Example

```
// set the search text to abc AND 123  
Everything_SetSearch("abc 123");  
  
// execute the query  
Everything_Query(TRUE);  
  
// Get the extension part of the first visible result.  
LPCTSTR lpExtension = Everything_GetResultExtension(0);
```

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultFileListFileName

The **Everything_GetResultFileListFileName** function retrieves the file list full path and filename of the visible result.

Syntax

```
LPCTSTR Everything_GetResultFileListFileName(  
    DWORD dwIndex  
);
```

Parameters

dwIndex

Zero based index of the visible result.

Return Value

The function returns a pointer to a null terminated string of **TCHARs**.

If the function fails the return value is NULL. To get extended error information, call [Everything_GetLastError](#).

If the result specified by dwIndex is not in a file list, then the filename returned is an empty string.

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultFileListFileName.
EVERYTHING_ERROR_INVALIDREQUEST	The file list filename was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_FILE_LIST_FILE_NAME before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

The ANSI / Unicode version of this function must match the ANSI / Unicode version of the call to [Everything_Query](#).

The function returns a pointer to an internal structure that is only valid until the next call to [Everything_Query](#) or [Everything_Reset](#).

dwIndex must be a valid visible result index. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the file list filename of the first visible result.
LPCTSTR lpFileListFileName = Everything_GetResultFileListFileName(0);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultFileName

The **Everything_GetResultFileName** function retrieves the file name part only of the visible result.

Syntax

```
LPCTSTR Everything_GetResultFileName(
    DWORD index
);
```

Parameters

index

Zero based index of the visible result.

Return Value

The function returns a pointer to a null terminated string of **TCHARs**.

If the function fails the return value is NULL. To get extended error information, call [Everything_GetLastError](#).

Error code

Meaning

EVERYTHING_ERROR_INVALIDCALL

Call [Everything_Query](#) before calling [Everything_GetResultFileName](#).

EVERYTHING_ERROR_INVALIDINDEX

index must be greater than or equal to 0 and less than the visible number of results.

Remarks

The ANSI / Unicode version of this function must match the ANSI / Unicode version of the call to [Everything_Query](#).

The function is faster than [Everything_GetResultFullPathName](#) as no memory copying occurs.

The function returns a pointer to an internal structure that is only valid until the next call to [Everything_Query](#) or [Everything_Reset](#).

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the file name part of the first visible result.
LPCTSTR cFileName = Everything_GetResultFileName(0);
```

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_GetResultPath](#)
- [Everything_GetResultFullPathName](#)

Everything_GetResultFullPathName

The **Everything_GetResultFullPathName** function retrieves the full path and file name of the visible result.

Syntax

```
DWORD Everything_GetResultFullPathName(
    DWORD index,
    LPTSTR lpString,
    DWORD nMaxCount
);
```

Parameters

index

Zero based index of the visible result.

lpString [out]

Pointer to the buffer that will receive the text. If the string is as long or longer than the buffer, the string is truncated and terminated with a NULL character.

nMaxCount

Specifies the maximum number of characters to copy to the buffer, including the NULL character. If the text exceeds this limit, it is truncated.

Return Value

If *lpString* is NULL, the return value is the number of **TCHARs** excluding the null terminator needed to store the full path and file name of the visible result.

If *lpString* is not NULL, the return value is the number of **TCHARs** excluding the null terminator copied into *lpString*.

If the function fails the return value is 0. To get extended error information, call [Everything_GetLastError](#).

Error code	Description
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultFullPathName .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

You can mix ANSI / Unicode versions of [Everything_GetResultFullPathName](#) and [Everything_Query](#).

Example

```
TCHAR buf[MAX_PATH];

// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the full path and file name of the first visible result.
Everything_GetResultFullPathName(0,buf,sizeof(buf) / sizeof(TCHAR));
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_GetResultFileName](#)
- [Everything_GetResultPath](#)

Everything_GetResultHighlightedFileName

The **Everything_GetResultHighlightedFileName** function retrieves the highlighted file name part of the visible result.

Syntax

```
LPCTSTR Everything_GetResultHighlightedFileName (  
    int index  
);
```

Parameters

index

Zero based index of the visible result.

Return Value

The function returns a pointer to a null terminated string of **TCHARs**.

If the function fails the return value is NULL. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultHighlightedFileName.
EVERYTHING_ERROR_INVALIDREQUEST	Highlighted file name information was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_DATE_HIGHLIGHTED_FILE_NAME before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

The ANSI / Unicode version of this function must match the ANSI / Unicode version of the call to [Everything_Query](#).

The function returns a pointer to an internal structure that is only valid until the next call to [Everything_Query](#) or [Everything_Reset](#).

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Text inside a * quote is highlighted, two consecutive *'s is a single literal *.

For example, in the highlighted text: abc *123* the 123 part is highlighted.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the highlighted file name of the first visible result.
LPCTSTR lpHighlightedFileName = Everything_GetResultHighlightedFileName(0);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultHighlightedFullPathAndFileName

The **Everything_GetResultHighlightedFullPathAndFileName** function retrieves the highlighted full path and file name of the visible result.

Syntax

```
LPCTSTR Everything_GetResultHighlightedFullPathAndFileName (
    int index
);
```

Parameters

index

Zero based index of the visible result.

Return Value

The function returns a pointer to a null terminated string of **TCHARs**.

If the function fails the return value is NULL. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultHighlightedFullPathAndFileName .
EVERYTHING_ERROR_INVALIDREQUEST	Highlighted full path and file name information was not requested or is Call Everything_SetRequestFlags with EVERYTHING_REQUEST_DATE_HIGHLIGHTED_FULL_PATH_AND before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible numt

Remarks

The ANSI / Unicode version of this function must match the ANSI / Unicode version of the call to [Everything_Query](#).

The function returns a pointer to an internal structure that is only valid until the next call to [Everything_Query](#) or [Everything_Reset](#).

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Text inside a * quote is highlighted, two consecutive *'s is a single literal *.

For example, in the highlighted text: abc *123* the 123 part is highlighted.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the highlighted full path and file name information of the first visible result
LPCTSTR lpHighlightedFullPathAndFileName = Everything_GetResultHighlightedFullPathAndFileName(0);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultHighlightedPath

The **Everything_GetResultHighlightedPath** function retrieves the highlighted path part of the visible result.

Syntax

```
LPCTSTR Everything_GetResultHighlightedPath(  
    int index  
);
```

Parameters

index

Zero based index of the visible result.

Return Value

The function returns a pointer to a null terminated string of **TCHARs**.

If the function fails the return value is NULL. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultHighlightedPath .
EVERYTHING_ERROR_INVALIDREQUEST	Highlighted path information was not requested or is unavailable, Call Everything_SetRequestFlags with <code>EVERYTHING_REQUEST_DATE_HIGHLIGHTED_PATH</code> before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

The ANSI / Unicode version of this function must match the ANSI / Unicode version of the call to [Everything_Query](#). The function returns a pointer to an internal structure that is only valid until the next call to [Everything_Query](#) or [Everything_Reset](#).

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Text inside a * quote is highlighted, two consecutive *'s is a single literal *.

For example, in the highlighted text: abc *123* the 123 part is highlighted.

Example

```
// set the search text to abc AND 123  
Everything_SetSearch("abc 123");  
  
// execute the query  
Everything_Query(TRUE);  
  
// Get the highlighted path of the first visible result.  
LPCTSTR lpHighlightedPath = Everything_GetResultHighlightedPath(0);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultListRequestFlags

The **Everything_GetResultListRequestFlags** function returns the flags of available result data.

Syntax

```
DWORD Everything_GetResultListRequestFlags(void);
```

Return Value

Returns zero or more of the following request flags:

EVERYTHING_REQUEST_FILE_NAME	(0x00000001)
EVERYTHING_REQUEST_PATH	(0x00000002)
EVERYTHING_REQUEST_FULL_PATH_AND_FILE_NAME	(0x00000004)
EVERYTHING_REQUEST_EXTENSION	(0x00000008)
EVERYTHING_REQUEST_SIZE	(0x00000010)
EVERYTHING_REQUEST_DATE_CREATED	(0x00000020)
EVERYTHING_REQUEST_DATE_MODIFIED	(0x00000040)
EVERYTHING_REQUEST_DATE_ACCESSED	(0x00000080)
EVERYTHING_REQUEST_ATTRIBUTES	(0x00000100)
EVERYTHING_REQUEST_FILE_LIST_FILE_NAME	(0x00000200)
EVERYTHING_REQUEST_RUN_COUNT	(0x00000400)
EVERYTHING_REQUEST_DATE_RUN	(0x00000800)
EVERYTHING_REQUEST_DATE_RECENTLY_CHANGED	(0x00001000)
EVERYTHING_REQUEST_HIGHLIGHTED_FILE_NAME	(0x00002000)
EVERYTHING_REQUEST_HIGHLIGHTED_PATH	(0x00004000)
EVERYTHING_REQUEST_HIGHLIGHTED_FULL_PATH_AND_FILE_NAME	(0x00008000)

Remarks

The requested result data may differ to the desired result data specified in [Everything_SetRequestFlags](#).

Example

```
DWORD dwRequestFlags = Everything_GetResultListRequestFlags();
```

See Also

- [Everything_SetRequestFlags](#)
- [Everything_Query](#)

Everything_GetResultListSort

The **Everything_GetResultListSort** function returns the actual sort order for the results.

Syntax

```
DWORD Everything_GetResultListSort(void);
```

Return Value

Returns one of the following sort types:

EVERYTHING_SORT_NAME_ASCENDING	(1)
EVERYTHING_SORT_NAME_DESCENDING	(2)
EVERYTHING_SORT_PATH_ASCENDING	(3)
EVERYTHING_SORT_PATH_DESCENDING	(4)
EVERYTHING_SORT_SIZE_ASCENDING	(5)
EVERYTHING_SORT_SIZE_DESCENDING	(6)
EVERYTHING_SORT_EXTENSION_ASCENDING	(7)
EVERYTHING_SORT_EXTENSION_DESCENDING	(8)
EVERYTHING_SORT_TYPE_NAME_ASCENDING	(9)
EVERYTHING_SORT_TYPE_NAME_DESCENDING	(10)
EVERYTHING_SORT_DATE_CREATED_ASCENDING	(11)
EVERYTHING_SORT_DATE_CREATED_DESCENDING	(12)
EVERYTHING_SORT_DATE_MODIFIED_ASCENDING	(13)
EVERYTHING_SORT_DATE_MODIFIED_DESCENDING	(14)
EVERYTHING_SORT_ATTRIBUTES_ASCENDING	(15)
EVERYTHING_SORT_ATTRIBUTES_DESCENDING	(16)
EVERYTHING_SORT_FILE_LIST_FILENAME_ASCENDING	(17)
EVERYTHING_SORT_FILE_LIST_FILENAME_DESCENDING	(18)
EVERYTHING_SORT_RUN_COUNT_ASCENDING	(19)
EVERYTHING_SORT_RUN_COUNT_DESCENDING	(20)
EVERYTHING_SORT_DATE_RECENTLY_CHANGED_ASCENDING	(21)
EVERYTHING_SORT_DATE_RECENTLY_CHANGED_DESCENDING	(22)
EVERYTHING_SORT_DATE_ACCESSED_ASCENDING	(23)
EVERYTHING_SORT_DATE_ACCESSED_DESCENDING	(24)
EVERYTHING_SORT_DATE_RUN_ASCENDING	(25)
EVERYTHING_SORT_DATE_RUN_DESCENDING	(26)

Remarks

Everything_GetResultListSort must be called after calling [Everything_Query](#).

If no desired sort order was specified the result list is sorted by EVERYTHING_SORT_NAME_ASCENDING.

The result list sort may differ to the desired sort specified in [Everything_SetSort](#).

Example

```
DWORD dwSort = Everything_GetResultListSort();
```

See Also

- [Everything_SetSort](#)
- [Everything_Query](#)

Everything_GetResultPath

The **Everything_GetResultPath** function retrieves the path part of the visible result.

Syntax

```
LPCTSTR Everything_GetResultPath(  
    int index  
);
```

Parameters

index

Zero based index of the visible result.

Return Value

The function returns a pointer to a null terminated string of **TCHARs**.

If the function fails the return value is NULL. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultPath.
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

The ANSI / Unicode version of this function must match the ANSI / Unicode version of the call to [Everything_Query](#).

The function is faster than Everything_GetResultFullPathName as no memory copying occurs.

The function returns a pointer to an internal structure that is only valid until the next call to [Everything_Query](#) or [Everything_Reset](#).

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the path part of the first visible result.
LPCTSTR lpPath = Everything_GetResultPath(0);
```

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_GetResultFileName](#)
- [Everything_GetResultFullPathName](#)

Everything_GetResultRunCount

The **Everything_GetResultRunCount** function retrieves the number of times a visible result has been run from Everything.

Syntax

```
DWORD Everything_GetResultRunCount (
    DWORD dwIndex,
);
```

Parameters

dwIndex

Zero based index of the visible result.

Return Value

The function returns the number of times the result has been run from Everything.

The function returns 0 if the run count information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_SUCCESS	The run count is 0.
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultRunCount.

EVERYTHING_ERROR_INVALIDREQUEST	Run count information was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_RUN_COUNT before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

dwIndex must be a valid visible result index. To determine if a result index is visible use the [Everything_GetNumResults](#) function.

Example

```
DWORD runCount;

// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the run count of the first visible result.
runCount = Everything_GetResultRunCount(0);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetResultSize

The **Everything_GetResultSize** function retrieves the size of a visible result.

Syntax

```
BOOL Everything_GetResultSize(
    DWORD dwIndex,
    LARGE_INTEGER *lpSize
);
```

Parameters

dwIndex

Zero based index of the visible result.

lpSize

Pointer to a LARGE_INTEGER to hold the size of the result.

Return Value

The function returns non-zero if successful.

The function returns 0 if size information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetResultSize.
EVERYTHING_ERROR_INVALIDREQUEST	Size was not requested or is unavailable, Call Everything_SetRequestFlags with EVERYTHING_REQUEST_SIZE before calling Everything_Query .
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

dwIndex must be a valid visible result index. To determine if a result index is visible use the [Everything_GetNumResults](#) function.

Example

```
LARGE_INTEGER size;

// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// Get the size of the first visible result.
Everything_GetResultSize(0, &size);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_Reset](#)
- [Everything_SetRequestFlags](#)

Everything_GetRevision

The **Everything_GetRevision** function retrieves the revision number of Everything.

Syntax

```
DWORD Everything_GetRevision(void);
```

Parameters

No parameters.

Return Value

The function returns the revision number.

The function returns 0 if revision information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Everything uses the following version format:

major.minor.revision.build

The build part is incremental and unique for all Everything versions.

Example

```
DWORD majorVersion;  
DWORD minorVersion;  
DWORD revision;  
  
// get version information  
majorVersion = Everything_GetMajorVersion();  
minorVersion = Everything_GetMinorVersion();  
revision = Everything_GetRevision();
```

```
if ((majorVersion > 1) || ((majorVersion == 1) && (minorVersion > 3)) || ((majorVers
{
    // do something with version 1.3.4 or later
}
```

Function Information

Requires Everything 1.0.0.0 or later.

See Also

- [Everything_GetMajorVersion](#)
- [Everything_GetMinorVersion](#)
- [Everything_GetBuildNumber](#)
- [Everything_GetTargetMachine](#)

Everything_GetRunCountFromFileName

The **Everything_GetRunCountFromFileName** function gets the run count from a specified file in the Everything index by file name.

Syntax

```
DWORD Everything_GetRunCountFromFileName(
    LPCTSTR lpFileName
);
```

Parameters

lpFileName [in]

Pointer to a null-terminated string that specifies a fully qualified file name in the Everything index.

Return Value

The function returns the number of times the file has been run from Everything.

The function returns 0 if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_SUCCESS	The run count is 0.
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Example

```
DWORD runCount;  
  
// get the run count of a file.  
runCount = Everything_GetRunCountFromFileName(TEXT("C:\\folder\\file.doc"));
```

See Also

- [Everything_SetRunCountFromFileName](#)
- [Everything_IncRunCountFromFileName](#)

Everything_GetSearch

The **Everything_GetSearch** function retrieves the search text to use for the next call to [Everything_Query](#).

Syntax

```
LPCTSTR Everything_GetSearch(void);
```

Return Value

The return value is a pointer to the null terminated search string.

The unicode and ansi version must match the call to [Everything_SetSearch](#).

The function will fail if you call [Everything_GetSearchA](#) after a call to [Everything_SetSearchW](#)

The function will fail if you call [Everything_GetSearchW](#) after a call to [Everything_SetSearchA](#)

If the function fails, the return value is NULL.

To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Mismatched unicode / ansi call.

Remarks

Get the internal state of the search text.

The default string is an empty string.

Example

```
LPCTSTR lpSearchString = Everything_GetSearch();
```

See Also

- [Everything_SetSearch](#)
- [Everything_Query](#)

Everything_GetSort

The **Everything_GetSort** function returns the desired sort order for the results.

Syntax

```
DWORD Everything_GetSort(void);
```

Return Value

Returns one of the following sort types:

EVERYTHING_SORT_NAME_ASCENDING	(1)
EVERYTHING_SORT_NAME_DESCENDING	(2)
EVERYTHING_SORT_PATH_ASCENDING	(3)
EVERYTHING_SORT_PATH_DESCENDING	(4)
EVERYTHING_SORT_SIZE_ASCENDING	(5)
EVERYTHING_SORT_SIZE_DESCENDING	(6)
EVERYTHING_SORT_EXTENSION_ASCENDING	(7)
EVERYTHING_SORT_EXTENSION_DESCENDING	(8)
EVERYTHING_SORT_TYPE_NAME_ASCENDING	(9)
EVERYTHING_SORT_TYPE_NAME_DESCENDING	(10)
EVERYTHING_SORT_DATE_CREATED_ASCENDING	(11)
EVERYTHING_SORT_DATE_CREATED_DESCENDING	(12)
EVERYTHING_SORT_DATE_MODIFIED_ASCENDING	(13)
EVERYTHING_SORT_DATE_MODIFIED_DESCENDING	(14)
EVERYTHING_SORT_ATTRIBUTES_ASCENDING	(15)
EVERYTHING_SORT_ATTRIBUTES_DESCENDING	(16)
EVERYTHING_SORT_FILE_LIST_FILENAME_ASCENDING	(17)
EVERYTHING_SORT_FILE_LIST_FILENAME_DESCENDING	(18)
EVERYTHING_SORT_RUN_COUNT_ASCENDING	(19)
EVERYTHING_SORT_RUN_COUNT_DESCENDING	(20)
EVERYTHING_SORT_DATE_RECENTLY_CHANGED_ASCENDING	(21)
EVERYTHING_SORT_DATE_RECENTLY_CHANGED_DESCENDING	(22)
EVERYTHING_SORT_DATE_ACCESSED_ASCENDING	(23)
EVERYTHING_SORT_DATE_ACCESSED_DESCENDING	(24)
EVERYTHING_SORT_DATE_RUN_ASCENDING	(25)
EVERYTHING_SORT_DATE_RUN_DESCENDING	(26)

Remarks

The default sort is EVERYTHING_SORT_NAME_ASCENDING (1)

Example

```
DWORD dwSort = Everything_GetSort();
```

See Also

- [Everything_SetSort](#)
- [Everything_Query](#)

Everything_GetTargetMachine

The **Everything_GetTargetMachine** function retrieves the target machine of Everything.

Syntax

```
DWORD Everything_GetTargetMachine(void);
```

Parameters

No parameters.

Return Value

The function returns one of the following:

Macro	Meaning
EVERYTHING_TARGET_MACHINE_X86 (1)	Target machine is x86 (32 bit).
EVERYTHING_TARGET_MACHINE_X64 (2)	Target machine is x64 (64 bit).
EVERYTHING_TARGET_MACHINE_ARM (3)	Target machine is ARM.

The function returns 0 if target machine information is unavailable. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Everything uses the following version format:
major.minor.revision.build

The build part is incremental and unique for all Everything versions.

Example

```
DWORD targetMachine;

// Get the attributes of the first visible result.
targetMachine = Everything_GetTargetMachine();

if (targetMachine == EVERYTHING_TARGET_MACHINE_X64)
{
    // do something with x64 build.
}
else
if (targetMachine == EVERYTHING_TARGET_MACHINE_X86)
{
    // do something with x86 build.
}
```

Function Information

Requires Everything 1.4.0 or later.

See Also

- [Everything_GetMajorVersion](#)
- [Everything_GetMinorVersion](#)
- [Everything_GetRevision](#)
- [Everything_GetBuildNumber](#)

Everything_GetTotFileResults

The **Everything_GetTotFileResults** function returns the total number of file results.

Syntax

```
DWORD Everything_GetTotFileResults(void);
```

Parameters

This functions has no parameters.

Return Value

Returns the total number of file results.

If the function fails the return value is 0. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetTotFileResults.

Remarks

You must call [Everything_Query](#) before calling Everything_GetTotFileResults.

Use [Everything_GetNumFileResults](#) to retrieve the number of visible file results.

Use the result offset and max result values to limit the number of visible results.

Everything_GetTotFileResults is not supported when using [Everything_SetRequestFlags](#)

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// get the total number of file results.
DWORD dwTotFileResults = Everything_GetTotFileResults();
```

See Also

- [Everything_Query](#)
- [Everything_GetNumResults](#)
- [Everything_GetNumFileResults](#)
- [Everything_GetNumFolderResults](#)
- [Everything_GetTotResults](#)
- [Everything_GetTotFolderResults](#)

Everything_GetTotFolderResults

The **Everything_GetTotFolderResults** function returns the total number of folder results.

Syntax

```
DWORD Everything_GetTotFolderResults(void);
```

Parameters

This functions has no parameters.

Return Value

Returns the total number of folder results.

If the function fails the return value is 0. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetTotFolderResults .

Remarks

You must call [Everything_Query](#) before calling [Everything_GetTotFolderResults](#).

Use [Everything_GetNumFolderResults](#) to retrieve the number of visible folder results.

Use the result offset and max result values to limit the number of visible results.

[Everything_GetTotFolderResults](#) is not supported when using [Everything_SetRequestFlags](#)

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// get the total number of folder results.
DWORD dwTotFolderResults = Everything_GetTotFolderResults();
```

See Also

- [Everything_Query](#)
- [Everything_GetNumResults](#)
- [Everything_GetNumFileResults](#)
- [Everything_GetNumFolderResults](#)
- [Everything_GetTotResults](#)
- [Everything_GetTotFileResults](#)

Everything_GetTotResults

The **Everything_GetTotResults** function returns the total number of file and folder results.

Syntax

```
DWORD Everything_GetTotResults(void);
```

Parameters

This functions has no parameters.

Return Value

Returns the total number of file and folder results.

If the function fails the return value is 0. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_GetTotResults.

Remarks

You must call [Everything_Query](#) before calling Everything_GetTotResults.

Use [Everything_GetNumResults](#) to retrieve the number of visible file and folder results.

Use the result offset and max result values to limit the number of visible results.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// get the total number of file and folder results.
DWORD dwTotResults = Everything_GetTotResults();
```

See Also

- [Everything_Query](#)
- [Everything_GetNumResults](#)
- [Everything_GetNumFileResults](#)
- [Everything_GetNumFolderResults](#)
- [Everything_GetTotFileResults](#)
- [Everything_GetTotFolderResults](#)

Everything_IncRunCountFromFileName

The **Everything_IncRunCountFromFileName** function increments the run count by one for a specified file in the Everything by file name.

Syntax

```
DWORD Everything_IncRunCountFromFileName(
```

```
LPCTSTR lpFileName  
) ;
```

Parameters

lpFileName [in]

Pointer to a null-terminated string that specifies a fully qualified file name in the Everything index.

Return Value

The function returns the new run count for the specified file.

The function returns 0 if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

The file does not have to exist. When the file is created it will have the correct run history.

Run history information is preserved between file deletion and creation.

Calling this function will update the date run to the current time for the specified file.

Incrementing a file with a run count of 4294967295 (0xffffffff) will do nothing.

Example

```
// run a file  
system("C:\\folder\\file.doc");  
  
// increment the run count in Everything.  
Everything_IncRunCountFromFileName(TEXT("C:\\folder\\file.doc"));
```

See Also

- [Everything_GetRunCountFromFileName](#)
- [Everything_SetRunCountFromFileName](#)

Everything_IsAdmin

The **Everything_IsAdmin** function checks if Everything is running as administrator or as a standard user.

Syntax

```
BOOL Everything_IsAdmin(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if the Everything is running as an administrator.

The function returns 0 Everything is running as a standard user or if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_SUCCESS	Everything is running as a standard user.
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Example

```
BOOL isAdmin;  
  
isAdmin = Everything_IsAdmin();
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_IsAppData](#)

Everything_IsAppData

The **Everything_IsAppData** function checks if Everything is saving settings and data to %APPDATA%\Everything or to the same location as the Everything.exe.

Syntax

```
BOOL Everything_IsAppData(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if settings and data are saved in %APPDATA%\Everything.

The function returns 0 if settings and data are saved to the same location as the Everything.exe or if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_SUCCESS	Settings and data are stored in the same location as the Everything.exe.
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Example

```
BOOL isAppData;  
  
isAppData = Everything_IsAppData();
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_IsAdmin](#)

Everything_IsDBLoaded

The **Everything_IsDBLoaded** function checks if the database has been fully loaded.

Syntax

```
BOOL Everything_IsDBLoaded(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if the Everything database is fully loaded.

The function returns 0 if the database has not fully loaded or if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_SUCCESS	The database is still loading.
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

When Everything is loading, any queries will appear to return no results.

Use **Everything_IsDBLoaded** to determine if the database has been loaded before performing a query.

Example

```
for (;;)
{
    if (Everything_IsDBLoaded())
    {
        // perform a query...
        break;
    }
    else
    {
        if (Everything_GetLastError())
        {
            // IPC not running.
            break;
        }
    }

    // wait for database to load..
    Sleep(1000);
}
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)

Everything_IsFastSort

The **Everything_IsFastSort** function checks if the specified file information is indexed and has fast sort enabled.

Syntax

```
BOOL Everything_IsFastSort(DWORD sortType);
```

Parameters

sortType

Can be one of the following:

Code	Value	Description
EVERYTHING_IPC_SORT_NAME_ASCENDING	1	Name ascending
EVERYTHING_IPC_SORT_NAME_DESCENDING	2	Name descending
EVERYTHING_IPC_SORT_PATH_ASCENDING	3	Path ascending
EVERYTHING_IPC_SORT_PATH_DESCENDING	4	Path descending
EVERYTHING_IPC_SORT_SIZE_ASCENDING	5	Size ascending
EVERYTHING_IPC_SORT_SIZE_DESCENDING	6	Size descending
EVERYTHING_IPC_SORT_EXTENSION_ASCENDING	7	Extension ascending
EVERYTHING_IPC_SORT_EXTENSION_DESCENDING	8	Extension descending
EVERYTHING_IPC_SORT_TYPE_NAME_ASCENDING	9	Type name ascending
EVERYTHING_IPC_SORT_TYPE_NAME_DESCENDING	10	Type name descending

EVERYTHING_IPC_SORT_DATE_CREATED_ASCENDING	11	Date created ascending
EVERYTHING_IPC_SORT_DATE_CREATED_DESCENDING	12	Date created descending
EVERYTHING_IPC_SORT_DATE_MODIFIED_ASCENDING	13	Date modified ascending
EVERYTHING_IPC_SORT_DATE_MODIFIED_DESCENDING	14	Date modified descending
EVERYTHING_IPC_SORT_ATTRIBUTES_ASCENDING	15	Attributes ascending
EVERYTHING_IPC_SORT_ATTRIBUTES_DESCENDING	16	Attributes descending
EVERYTHING_IPC_SORT_FILE_LIST_FILENAME_ASCENDING	17	File list filename ascending
EVERYTHING_IPC_SORT_FILE_LIST_FILENAME_DESCENDING	18	File list filename descending
EVERYTHING_IPC_SORT_RUN_COUNT_ASCENDING	19	Run count ascending
EVERYTHING_IPC_SORT_RUN_COUNT_DESCENDING	20	Run count descending
EVERYTHING_IPC_SORT_DATE_RECENTLY_CHANGED_ASCENDING	21	Recently changed ascending
EVERYTHING_IPC_SORT_DATE_RECENTLY_CHANGED_DESCENDING	22	Recently changed descending
EVERYTHING_IPC_SORT_DATE_ACCESSED_ASCENDING	23	Date accessed ascending
EVERYTHING_IPC_SORT_DATE_ACCESSED_DESCENDING	24	Date accessed descending
EVERYTHING_IPC_SORT_DATE_RUN_ASCENDING	25	Date run ascending
EVERYTHING_IPC_SORT_DATE_RUN_DESCENDING	26	Date run descending

Return Value

The function returns non-zero if the specified information is indexed and has fast sort enabled.

The function returns 0 if the specified information does not have fast sort enabled or if an error occurred. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_SUCCESS	The specified information does not have fast sort enabled.
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

The following sort types are always fast sort types:

Code	Value	Description
EVERYTHING_IPC_SORT_NAME_ASCENDING	1	Name ascending
EVERYTHING_IPC_SORT_NAME_DESCENDING	2	Name descending
EVERYTHING_IPC_SORT_RUN_COUNT_ASCENDING	19	Run count ascending
EVERYTHING_IPC_SORT_RUN_COUNT_DESCENDING	20	Run count descending
EVERYTHING_IPC_SORT_DATE_RECENTLY_CHANGED_ASCENDING	21	Recently changed ascending
EVERYTHING_IPC_SORT_DATE_RECENTLY_CHANGED_DESCENDING	22	Recently changed descending
EVERYTHING_IPC_SORT_DATE_RUN_ASCENDING	25	Date run ascending
EVERYTHING_IPC_SORT_DATE_RUN_DESCENDING	26	Date run descending

Example

```
BOOL isFastSizeSort;  
  
// Check if sorting by size descending will be instant.  
isFastSizeSort = Everything_IsFastSort(EVERYTHING_IPC_SORT_SIZE_DESCENDING);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_IsFileInfoIndexed](#)
- [Enable or disable file information indexing](#)
- [Enable or disable fast sorting](#)

Everything_IsFileInfoIndexed

The **Everything_IsFileInfoIndexed** function checks if the specified file information is indexed.

Syntax

```
BOOL Everything_IsFileInfoIndexed(DWORD fileInfoType);
```

Parameters

fileInfoType

Can be one of the following:

Code	Value	Description
EVERYTHING_IPC_FILE_INFO_FILE_SIZE	1	File size
EVERYTHING_IPC_FILE_INFO_FOLDER_SIZE	2	Folder size
EVERYTHING_IPC_FILE_INFO_DATE_CREATED	3	Date created
EVERYTHING_IPC_FILE_INFO_DATE_MODIFIED	4	Date modified
EVERYTHING_IPC_FILE_INFO_DATE_ACCESSED	5	Date accessed
EVERYTHING_IPC_FILE_INFO_ATTRIBUTES	6	Attributes

Return Value

The function returns non-zero if the specified information is indexed.

The function returns 0 if the specified information is not indexed or if an error occurred. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_SUCCESS	The specified information is not indexed.
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Example

```
BOOL isSizeIndexed;  
  
isSizeIndexed = Everything_IsFileInfoIndexed(EVERYTHING_IPC_FILE_INFO_FILE_SIZE);
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_IsFastSort](#)
- [Enable or disable file information indexing](#)
- [Enable or disable fast sorting](#)

Everything_IsFileResult

The **Everything_IsFileResult** function determines if the visible result is file.

Syntax

```
BOOL Everything_IsFileResult(  
    DWORD index  
);
```

Parameters

index

Zero based index of the visible result.

Return Value

The function returns TRUE, if the visible result is a file (For example: C:\ABC.123).

The function returns FALSE, if the visible result is a folder or volume (For example: C: or c:\WINDOWS).

If the function fails the return value is FALSE. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_IsFileResult.

EVERYTHING_ERROR_INVALIDINDEX

index must be greater than or equal to 0 and less than the visible number of results.

Remarks

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// determine if the first visible result is a file.
BOOL bIsFileResult = Everything_IsFileResult(0);
```

See Also

- [Everything_Query](#)
- [Everything_IsVolumeResult](#)
- [Everything_IsFolderResult](#)

Everything_IsFolderResult

The **Everything_IsFolderResult** function determines if the visible result is a folder.

Syntax

```
BOOL Everything_IsFolderResult(  
    DWORD index  
);
```

Parameters

index

Zero based index of the visible result.

Return Value

The function returns TRUE, if the visible result is a folder or volume (For example: C: or c:\WINDOWS).

The function returns FALSE, if the visible result is a file (For example: C:\ABC.123).

If the function fails the return value is FALSE. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_IsFolderResult.
EVERYTHING_ERROR_INVALIDINDEX	Index must be greater than or equal to 0 and less than the visible number of results.

Remarks

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// determine if the first visible result is a folder.
BOOL bIsFolderResult = Everything_IsFolderResult(0);
```

See Also

- [Everything_Query](#)
- [Everything_IsVolumeResult](#)
- [Everything_IsFileResult](#)

Everything_IsQueryReply

The **Everything_IsQueryReply** function checks if the specified window message is a query reply.

Syntax

```
BOOL EVERYTHINGAPI Everything_IsQueryReply(
    UINT message,
    WPARAM wParam,
    LPARAM lParam,
    DWORD nId
);
```

Parameters

message

Specifies the message identifier.

wParam

Specifies additional information about the message.

lParam

Specifies additional information about the message.

nId

The unique identifier specified with [Everything_SetReplyID](#), or 0 for the default ID.

This is the value used to compare with the dwData member of the COPYDATASTRUCT if the message is WM_COPYDATA.

Return Value

Returns TRUE if the message is a query reply.

If the function fails the return value is FALSE. To get extended error information, call: [Everything_GetLastError](#).

Remarks

This function checks if the message is a WM_COPYDATA message. If the message is a WM_COPYDATA message the function checks if the ReplyID matches the dwData member of the COPYDATASTRUCT. If they match the function makes a copy of the query results.

You must call [Everything_IsQueryReply](#) in the windows message handler to check for an IPC query reply if you call [Everything_Query](#) with bWait set to FALSE.

If the function returns TRUE you should return TRUE.

If the function returns TRUE you can call the following functions to read the results:

- [Everything_SortResultsByPath](#)
- [Everything_Reset](#)
- [Everything_GetNumFileResults](#)
- [Everything_GetNumFolderResults](#)
- [Everything_GetNumResults](#)
- [Everything_GetTotFileResults](#)
- [Everything_GetTotFolderResults](#)
- [Everything_GetTotResults](#)
- [Everything_IsVolumeResult](#)
- [Everything_IsFolderResult](#)
- [Everything_IsFileResult](#)
- [Everything_GetResultFileName](#)
- [Everything_GetResultPath](#)
- [Everything_GetResultFullPathName](#)

Example

```
LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    if (Everything_IsQueryReply(uMsg, wParam, lParam, 0))
    {
        // ...
    }
}
```

```

        // do something with the results..
        // ...

        return TRUE;
    }

    // return the default window proc..
    return DefWindowProc(hwnd,uMsg,wParam,lParam);
}

```

Implementation

```

BOOL EVERYTHINGAPI Everything_IsQueryReply(UINT message,WPARAM wParam,LPARAM lParam,
{
    if (message == WM_COPYDATA)
    {
        COPYDATASTRUCT *cds = (COPYDATASTRUCT *)lParam;

        if (cds)
        {
            if (cds->dwData == _Everything_ReplyID)
            {
                if (_Everything_IsUnicodeQuery)
                {
                    if (!_Everything_List) HeapFree(GetProcessHeap(),0,cds->lpData);
                    _Everything_List = (EVERYTHING_IPC_LISTW *)HeapAlloc(GetProcessHeap(),0,cds->dwData);
                    if (_Everything_List)
                    {
                        CopyMemory(_Everything_List,cds->lpData,cds->dwData);
                    }
                    else
                    {
                        _Everything_LastError = EVERYTHING_E_OUT_OF_MEMORY;
                    }

                    return TRUE;
                }
            }
            else
            {
                if (!_Everything_List) HeapFree(GetProcessHeap(),0,cds->lpData);
                _Everything_List = (EVERYTHING_IPC_LISTW *)HeapAlloc(GetProcessHeap(),0,cds->dwData);
                if (_Everything_List)
                {
                    CopyMemory(_Everything_List,cds->lpData,cds->dwData);
                }
                else
                {
                    _Everything_LastError = EVERYTHING_E_OUT_OF_MEMORY;
                }

                return TRUE;
            }
        }
    }

    return FALSE;
}

```


See Also

- [Everything_Query](#)
- [Everything_SetReplyWindow](#)
- [Everything_SetReplyID](#)
- [Everything_GetReplyWindow](#)
- [Everything_GetReplyID](#)

Everything_IsVolumeResult

The **Everything_IsVolumeResult** function determines if the visible result is the root folder of a volume.

Syntax

```
BOOL Everything_IsVolumeResult(  
    DWORD index  
);
```

Parameters

index

Zero based index of the visible result.

Return Value

The function returns TRUE, if the visible result is a volume (For example: C:).

The function returns FALSE, if the visible result is a folder or file (For example: C:\ABC.123).

If the function fails the return value is FALSE. To get extended error information, call [Everything_GetLastError](#).

Error code	Meaning
EVERYTHING_ERROR_INVALIDCALL	Call Everything_Query before calling Everything_IsVolumeResult.
EVERYTHING_ERROR_INVALIDINDEX	index must be greater than or equal to 0 and less than the visible number of results.

Remarks

You can only call this function for a visible result. To determine if a result is visible use the [Everything_GetNumFileResults](#) function.

Example

```
// set the search text to abc AND 123
```

```
Everything_SetSearch("abc 123");

// execute the query
Everything_Query(TRUE);

// determine if the first visible result is a volume.
BOOL bIsVolumeResult = Everything_IsVolumeResult(0);
```

See Also

- [Everything_Query](#)
- [Everything_IsFolderResult](#)
- [Everything_IsFileResult](#)

Everything_Query

The **Everything_Query** function executes an Everything IPC query with the current search state.

Syntax

```
void Everything_Query(  
    BOOL bWait  
);
```

Parameters

bWait

Should the function wait for the results or return immediately.
Set this to FALSE to post the IPC Query and return immediately.
Set this to TRUE to send the IPC Query and wait for the results.

Return Value

If the function succeeds, the return value is TRUE.

If the function fails, the return value is FALSE. To get extended error information, call [Everything_GetLastError](#)

Error code	Description
EVERYTHING_ERROR_CREATETHREAD	Failed to create the search query thread.
EVERYTHING_ERROR_REGISTERCLASSEX	Failed to register the search query window class.
EVERYTHING_ERROR_CREATEWINDOW	Failed to create the search query window.
EVERYTHING_ERROR_IPC	IPC is not available. Make sure Everything is running.
EVERYTHING_ERROR_MEMORY	Failed to allocate memory for the search query.

Remarks

If bWait is FALSE you must call [Everything_SetReplyWindow](#) before calling Everything_Query. Use the [Everything_IsQueryReply](#) function to check for query replies.

Optionally call the following functions to set the search state before calling Everything_Query:

- [Everything_SetSearch](#)
- [Everything_SetMatchPath](#)
- [Everything_SetMatchCase](#)
- [Everything_SetMatchWholeWord](#)
- [Everything_SetRegex](#)
- [Everything_SetMax](#)
- [Everything_SetOffset](#)
- [Everything_SetReplyID](#)
- [Everything_SetRequestFlags](#)

You can mix ANSI / Unicode version of Everything_SetSearch and Everything_Query.

The ANSI / Unicode version of Everything_Query MUST match the ANSI / Unicode version of Everything_GetResultName and Everything_GetResultPath.

The search state is not modified from a call to Everything_Query.

The default state is as follows:

See [Everything_Reset](#) for the default search state.

Example

```
// set the search text to abc AND 123
Everything_SetSearch("abc 123");

// enable case sensitive searching.
Everything_SetMatchCase(TRUE);

// execute the query
Everything_Query(TRUE);
```

See Also

- [Everything_SetSearch](#)
- [Everything_SetMatchPath](#)
- [Everything_SetMatchCase](#)
- [Everything_SetMatchWholeWord](#)
- [Everything_SetRegex](#)
- [Everything_SetMax](#)
- [Everything_SetOffset](#)
- [Everything_SortResultsByPath](#)

- [Everything_GetLastError](#)
- [Everything_GetNumFileResults](#)
- [Everything_GetNumFolderResults](#)
- [Everything_GetNumResults](#)
- [Everything_GetTotFileResults](#)
- [Everything_GetTotFolderResults](#)
- [Everything_GetTotResults](#)
- [Everything_IsVolumeResult](#)
- [Everything_IsFolderResult](#)
- [Everything_IsFileResult](#)
- [Everything_GetResultFileName](#)
- [Everything_GetResultPath](#)
- [Everything_GetResultFullPathName](#)
- [Everything_SetReplyWindow](#)
- [Everything_SetReplyID](#)
- [Everything_Reset](#)

Everything_RebuildDB

The **Everything_RebuildDB** function requests Everything to forcefully rebuild the Everything index.

Syntax

```
BOOL Everything_RebuildDB(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if the request to forcefully rebuild the Everything index was successful.

The function returns 0 if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Requesting a rebuild will mark all indexes as dirty and start the rebuild process.

Use **Everything_IsDBLoaded** to determine if the database has been rebuilt before performing a query.

Example

```
// rebuild the database.  
Everything_RebuildDB();
```

Function Information

Requires Everything 1.4.1 or later.

See Also

- [Everything_IsDBLoaded](#)

Everything_Reset

The **Everything_Reset** function resets the result list and search state to the default state, freeing any allocated memory by the library.

Syntax

```
void Everything_Reset(void);
```

Parameters

This function has no parameters.

Return Value

This function has no return value.

Remarks

Calling [Everything_SetSearch](#) frees the old search and allocates the new search string.

Calling [Everything_Query](#) frees the old result list and allocates the new result list.

Calling [Everything_Reset](#) frees the current search and current result list.

The default state:

```
Everything_SetSearch("\\");  
Everything_SetMatchPath(FALSE);  
Everything_SetMatchCase(FALSE);  
Everything_SetMatchWholeWord(FALSE);  
Everything_SetRegex(FALSE);  
Everything_SetMax(0xFFFFFFFF);  
Everything_SetOffset(0);  
Everything_SetReplyWindow(0);
```

```
Everything_SetReplyID(0);
```

Example

```
Everything_Reset();
```

See Also

- [Everything_SetSearch](#)
- [Everything_Query](#)
- [Everything_CleanUp](#)

Everything_SaveDB

The **Everything_SaveDB** function requests Everything to save the index to disk.

Syntax

```
BOOL Everything_SaveDB(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if the request to save the Everything index to disk was successful.

The function returns 0 if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

The index is only saved to disk when you exit Everything.

Call **Everything_SaveDB** to write the index to the file: Everything.db

Example

```
// flush index to disk
Everything_SaveDB();
```

Function Information

Requires Everything 1.4.1 or later.

See Also

Everything_SaveRunHistory

The **Everything_SaveRunHistory** function requests Everything to save the run history to disk.

Syntax

```
BOOL Everything_SaveRunHistory(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if the request to save the run history to disk was successful.

The function returns 0 if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

The run history is only saved to disk when you close an Everything search window or exit Everything.

Call **Everything_RunHistory** to write the run history to the file: Run History.csv

Example

```
// flush run history to disk
Everything_SaveRunHistory();
```

Function Information

Requires Everything 1.4.1 or later.

See Also

Everything_SetMatchCase

The **Everything_SetMatchCase** function enables or disables full path matching for the next call to [Everything_Query](#).

Syntax

```
void Everything_SetMatchCase(
    BOOL bEnable
);
```

Parameters

bEnable

Specifies whether the search is case sensitive or insensitive.

If this parameter is TRUE, the search is case sensitive.

If the parameter is FALSE, the search is case insensitive.

Remarks

Match case is disabled by default.

Example

```
Everything_SetMatchCase(TRUE);
```

See Also

- [Everything_GetMatchCase](#)
- [Everything_Query](#)

Everything_SetMatchPath

The **Everything_SetMatchPath** function enables or disables full path matching for the next call to [Everything_Query](#).

Syntax

```
void Everything_SetMatchPath(  
    BOOL bEnable  
);
```

Parameters

bEnable

[in] Specifies whether to enable or disable full path matching.

If this parameter is TRUE, full path matching is enabled.

If the parameter is FALSE, full path matching is disabled.

Remarks

If match full path is being enabled, the next call to [Everything_Query](#) will search the full path and file name of each file and folder.

If match full path is being disabled, the next call to [Everything_Query](#) will search the file name only of each file and folder.

Match path is disabled by default.

Enabling match path will add a significant performance hit.

Example

```
Everything_SetMatchPath(TRUE);
```

See Also

- [Everything_GetMatchPath](#)
- [Everything_Query](#)

Everything_SetMatchWholeWord

The **Everything_SetMatchWholeWord** function enables or disables matching whole words for the next call to [Everything_Query](#).

Syntax

```
void Everything_SetMatchWholeWord(  
    BOOL bEnable  
);
```

Parameters

bEnable

Specifies whether the search matches whole words, or can match anywhere.

If this parameter is TRUE, the search matches whole words only.

If the parameter is FALSE, the search can occur anywhere.

Remarks

Match whole word is disabled by default.

Example

```
Everything_SetMatchWholeWord(TRUE);
```

See Also

- [Everything_GetMatchWholeWord](#)
- [Everything_Query](#)

Everything_SetMax

The **Everything_SetMax** function set the maximum number of results to return from [Everything_Query](#).

Syntax

```
void Everything_SetMax(  
    DWORD dwMaxResults  
);
```

Parameters

dwMaxResults

Specifies the maximum number of results to return.

Setting this to 0xffffffff will return all results.

Remarks

The default maximum number of results is 0xffffffff (all results).

If you are displaying the results in a window, set the maximum number of results to the number of visible items in the window.

Example

```
Everything_SetMax(window_height / item_height);
```

See Also

- [Everything_GetMax](#)
- [Everything_Query](#)

Everything_SetOffset

The **Everything_SetOffset** function set the first result offset to return from a call to [Everything_Query](#).

Syntax

```
void Everything_SetOffset(  
    DWORD dwOffset  
);
```

Parameters

dwOffset

Specifies the first result from the available results.
Set this to 0 to return the first available result.

Remarks

The default offset is 0 (the first available result).

If you are displaying the results in a window with a custom scroll bar, set the offset to the vertical scroll bar position. Using a search window can reduce the amount of data sent over the IPC and significantly increase search performance.

Example

```
Everything_SetOffset(scrollbar_vpos);
```

See Also

- [Everything_GetOffset](#)
- [Everything_Query](#)

Everything_SetRegex

The `Everything_SetRegex` function enables or disables Regular Expression searching.

Syntax

```
void Everything_SetRegex(  
    BOOL bEnabled  
);
```

Parameters

bEnabled

Set to non-zero to enable regex, set to zero to disable regex.

Return Value

This function has no return value.

Remarks

Regex is disabled by default.

Example

```
Everything_SetRegex(TRUE);
```

See Also

- [Everything_GetRegex](#)
- [Everything_Query](#)

Everything_SetReplyID

The **Everything_SetReplyID** function sets the unique number to identify the next query.

Syntax

```
void Everything_SetReplyID(  
    DWORD nId  
);
```

Parameters

nID

The unique number to identify the next query.

Return Value

This function has no return value.

Remarks

The default identifier is 0.

Set a unique identifier for the IPC Query.

If you want to post multiple search queries with the same window handle, you must call the `Everything_SetReplyID` function to assign each query a unique identifier.

The `nID` value is the `dwData` member in the `COPYDATASTRUCT` used in the `WM_COPYDATA` reply message.

This function is not required if you call `Everything_Query` with `bWait` set to true.

Example

```
// reply to this window.  
Everything_SetReplyWindow(hwnd);  
  
// set a unique identifier for this query.  
Everything_SetReplyID(1);  
  
// execute the query  
Everything_Query(FALSE);
```

See Also

- [Everything_Query](#)
- [Everything_SetReplyWindow](#)
- [Everything_GetReplyWindow](#)
- [Everything_GetReplyID](#)
- [Everything_IsQueryReply](#)

Everything_SetReplyWindow

The **Everything_SetReplyWindow** function sets the window that will receive the the IPC Query results.

Syntax

```
void Everything_SetReplyWindow(  
    HWND hWnd  
);
```

Parameters

hWnd

The handle to the window that will receive the IPC Query reply.

Return Value

This function has no return value.

Remarks

This function must be called before calling [Everything_Query](#) with `bWait` set to `FALSE`.

Check for results with the specified window using [Everything_IsQueryReply](#).

Call [Everything_SetReplyID](#) with a unique identifier to specify multiple searches.

Example

```
// reply to this window.  
Everything_SetReplyWindow(hWnd);  
  
// execute the query  
Everything_Query(TRUE);
```

See Also

- [Everything_Query](#)
- [Everything_SetReplyID](#)
- [Everything_GetReplyWindow](#)
- [Everything_GetReplyID](#)
- [Everything_IsQueryReply](#)

Everything_SetRequestFlags

The **Everything_SetRequestFlags** function sets the desired result data.

Syntax

```
void Everything_SetRequestFlags(  
    DWORD dwRequestFlags  
);
```

Parameters

dwRequestFlags

The request flags, can be zero or more of the following flags:

EVERYTHING_REQUEST_FILE_NAME	(0x00000001)
EVERYTHING_REQUEST_PATH	(0x00000002)
EVERYTHING_REQUEST_FULL_PATH_AND_FILE_NAME	(0x00000004)
EVERYTHING_REQUEST_EXTENSION	(0x00000008)
EVERYTHING_REQUEST_SIZE	(0x00000010)
EVERYTHING_REQUEST_DATE_CREATED	(0x00000020)
EVERYTHING_REQUEST_DATE_MODIFIED	(0x00000040)
EVERYTHING_REQUEST_DATE_ACCESSED	(0x00000080)
EVERYTHING_REQUEST_ATTRIBUTES	(0x00000100)
EVERYTHING_REQUEST_FILE_LIST_FILE_NAME	(0x00000200)
EVERYTHING_REQUEST_RUN_COUNT	(0x00000400)
EVERYTHING_REQUEST_DATE_RUN	(0x00000800)
EVERYTHING_REQUEST_DATE_RECENTLY_CHANGED	(0x00001000)
EVERYTHING_REQUEST_HIGHLIGHTED_FILE_NAME	(0x00002000)
EVERYTHING_REQUEST_HIGHLIGHTED_PATH	(0x00004000)
EVERYTHING_REQUEST_HIGHLIGHTED_FULL_PATH_AND_FILE_NAME	(0x00008000)

Return Value

This function has no return value.

Remarks

Make sure you include `EVERYTHING_REQUEST_FILE_NAME` and `EVERYTHING_REQUEST_PATH` if you want the result file name information returned.

The default request flags are `EVERYTHING_REQUEST_FILE_NAME | EVERYTHING_REQUEST_PATH` (0x00000003).

When the default flags (`EVERYTHING_REQUEST_FILE_NAME | EVERYTHING_REQUEST_PATH`) are used the SDK will use the old version 1 query.

When any other flags are used the new version 2 query will be tried first, and then fall back to version 1 query.

It is possible the requested data is not available, in which case after you have received your results you should call [Everything_GetResultListRequestFlags](#) to determine the available result data.

This function must be called before [Everything_Query](#).

Example

```
LARGE_INTEGER size;

// set the search.
Everything_SetSearch("123 ABC");

// request filename, path, size and date modified result data.
Everything_SetRequestFlags(EVERYTHING_REQUEST_FILE_NAME | EVERYTHING_REQUEST_PATH |

// execute the query
Everything_Query(TRUE);

// Get the size of the first result.
Everything_GetResultSize(0, &size);
```

Requirements

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)
- [Everything_GetResultListRequestFlags](#)

Everything_SetRunCountFromFileName

The **Everything_SetRunCountFromFileName** function sets the run count for a specified file in the Everything index by file name.

Syntax

```
BOOL Everything_SetRunCountFromFileName(
    LPCTSTR lpFileName,
    DWORD dwRunCount
);
```

Parameters

lpFileName [in]

Pointer to a null-terminated string that specifies a fully qualified file name in the Everything index.

dwRunCount [in]

The new run count.

Return Value

The function returns non-zero if successful.

The function returns 0 if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Set the run count to 0 to remove any run history information for the specified file.

The file does not have to exist. When the file is created it will have the correct run history.

Run history information is preserved between file deletion and creation.

Calling this function will update the date run to the current time for the specified file.

Example

```
// set a file to show higher in the results by setting an exaggerated run count
Everything_SetRunCountFromFileName(TEXT("C:\\folder\\file.doc"),1000);
```

See Also

- [Everything_GetRunCountFromFileName](#)
- [Everything_IncRunCountFromFileName](#)

Everything_SetSearch

The **Everything_SetSearch** function sets the search string for the IPC Query.

Syntax

```
void Everything_SetSearch(
    LPCTSTR lpString
);
```

Parameters

lpString [in]

Pointer to a null-terminated string to be used as the new search text.

Return Value

This function has no return value.

Remarks

Optionally call this function prior to a call to [Everything_Query](#).
[Everything_Query](#) executes the IPC Query using this search string.

Example

```
// Set the search string to abc AND 123
Everything_SetSearch("abc 123");

// Execute the IPC query.
Everything_Query(TRUE);
```

See Also

- [Everything_GetSearch](#)
- [Everything_Query](#)

Everything_SetSort

The **Everything_SetSort** function sets how the results should be ordered.

Syntax

```
void Everything_SetSort(  
    DWORD dwSortType  
);
```

Parameters

dwSortType

The sort type, can be one of the following values:

EVERYTHING_SORT_NAME_ASCENDING	(1)
EVERYTHING_SORT_NAME_DESCENDING	(2)
EVERYTHING_SORT_PATH_ASCENDING	(3)
EVERYTHING_SORT_PATH_DESCENDING	(4)
EVERYTHING_SORT_SIZE_ASCENDING	(5)
EVERYTHING_SORT_SIZE_DESCENDING	(6)
EVERYTHING_SORT_EXTENSION_ASCENDING	(7)
EVERYTHING_SORT_EXTENSION_DESCENDING	(8)
EVERYTHING_SORT_TYPE_NAME_ASCENDING	(9)
EVERYTHING_SORT_TYPE_NAME_DESCENDING	(10)
EVERYTHING_SORT_DATE_CREATED_ASCENDING	(11)
EVERYTHING_SORT_DATE_CREATED_DESCENDING	(12)
EVERYTHING_SORT_DATE_MODIFIED_ASCENDING	(13)
EVERYTHING_SORT_DATE_MODIFIED_DESCENDING	(14)
EVERYTHING_SORT_ATTRIBUTES_ASCENDING	(15)
EVERYTHING_SORT_ATTRIBUTES_DESCENDING	(16)
EVERYTHING_SORT_FILE_LIST_FILENAME_ASCENDING	(17)

EVERYTHING_SORT_FILE_LIST_FILENAME_DESCENDING	(18)
EVERYTHING_SORT_RUN_COUNT_ASCENDING	(19)
EVERYTHING_SORT_RUN_COUNT_DESCENDING	(20)
EVERYTHING_SORT_DATE_RECENTLY_CHANGED_ASCENDING	(21)
EVERYTHING_SORT_DATE_RECENTLY_CHANGED_DESCENDING	(22)
EVERYTHING_SORT_DATE_ACCESSED_ASCENDING	(23)
EVERYTHING_SORT_DATE_ACCESSED_DESCENDING	(24)
EVERYTHING_SORT_DATE_RUN_ASCENDING	(25)
EVERYTHING_SORT_DATE_RUN_DESCENDING	(26)

Return Value

This function has no return value.

Remarks

The default sort is `EVERYTHING_SORT_NAME_ASCENDING` (1). This sort is free.

Using fast sorts is recommended, fast sorting is instant.

To enable fast sorts in Everything:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **Indexes** tab.
- Check **fast sort** for desired fast sort type.
- Click **OK**.

It is possible the sort is not supported, in which case after you have received your results you should call `*Everything_GetResultListSort` to determine the sorting actually used.

This function must be called before [Everything_Query](#).

Example

```
// set the search.
Everything_SetSearch("123 ABC");

// sort by size descending.
Everything_SetSort(EVERYTHING_SORT_SIZE_DESCENDING);

// execute the query
Everything_Query(TRUE);
```

Requirements

Requires Everything 1.4.1 or later.

See Also

- [Everything_Query](#)

Everything_SortResultsByPath

The **Everything_SortResultsByPath** function sorts the current results by path, then file name. SortResultsByPath is CPU Intensive. Sorting by path can take several seconds.

Syntax

```
void Everything_SortResultsByPath(void);
```

Parameters

This functions has no parameters.

Return Value

This function has no return value.

Remarks

The default result list contains no results.

Call [Everything_Query](#) to retrieve the result list prior to a call to Everything_SortResultsByPath.

For improved performance, use [Everything/SDK/Everything_SetSort|Everything_SetSort]]

Example

```
// set the search text to abc AND 123
Everything_SetSearch(\"abc 123\");

// execute the query
Everything_Query(TRUE);

// sort the results by path
Everything_SortResultsByPath();
```

See Also

- [Everything_Query](#)

Everything_UpdateAllFolderIndexes

The **Everything_UpdateAllFolderIndexes** function requests Everything to rescan all folder indexes.

Syntax

```
BOOL Everything_UpdateAllFolderIndexes(void);
```

Parameters

No parameters.

Return Value

The function returns non-zero if the request to rescan all folder indexes was successful.

The function returns 0 if an error occurred. To get extended error information, call [Everything_GetLastError](#)

Error code	Meaning
EVERYTHING_ERROR_IPC	Please make sure the Everything search client is running in the background.

Remarks

Everything will begin updating all folder indexes in the background.

Example

```
// Request all folder indexes be rescanned.
Everything_UpdateAllFolderIndexes();
```

Function Information

Requires Everything 1.4.1 or later.

See Also

IPC

Inter Process Communication (IPC) allows applications to communicate with Everything under windows.

The Everything IPC allows applications to query and get search results from the "*Everything*" database.

You can only communicate with Everything from the local machine.

Use [ETP](#) if you would like to communicate with Everything over a network.

IPC SDK

[Everything-SDK.zip](#)

Examples

- [C](#)

See Also

- [ETP](#)
- [Clarion module](#)
- [ES Command line interface](#)
- [SDK](#)

IPC C Example

A C example to send a search query to Everything using WM_COPYDATA and retrieve the results.

Note: requires Everything running in the background.

Sending a query

```
#define COPYDATA_QUERYCOMPLETE 0

BOOL sendquery(HWND hwnd,DWORD max_results,WCHAR *search_string,BOOL regex,BOOL match)
{
    EVERYTHING_IPC_QUERY *query;
    DWORD len;
    DWORD size;
    HWND everything_hwnd;
    COPYDATASTRUCT cds;

    everything_hwnd = FindWindow(EVERYTHING_IPC_WNDCLASS,0);
    if (everything_hwnd)
    {
        len = (int)wcslen(search_string);

        size = sizeof(EVERYTHING_IPC_QUERY) - sizeof(WCHAR) + len*sizeof(WCHAR);

        query = (EVERYTHING_IPC_QUERY *)HeapAlloc(GetProcessHeap(),0,size);
        if (query)
        {
            query->max_results = max_results;
            query->offset = 0;
            query->reply_copydata_message = COPYDATA_QUERYCOMPLETE;
            query->search_flags = (regex?EVERYTHING_IPC_REGEX:0) | (match?EVERYTHING_IPC_MATCH:0);
            query->reply_hwnd = hwnd;

            CopyMemory(query->search_string,search_string,(len+1)*sizeof(WCHAR));

            cds.cbData = size;
            cds.dwData = EVERYTHING_IPC_COPYDATAQUERY;
            cds.lpData = query;

            if (SendMessage(everything_hwnd,WM_COPYDATA,(LPARAM)hwnd,(LPARAM)cds))
            {
                return TRUE;
            }
        }
    }
    return FALSE;
}
```

```

        HeapFree(GetProcessHeap(), 0, query);

        return TRUE;
    }
    else
    {
        // Everything IPC service not running
    }

    HeapFree(GetProcessHeap(), 0, query);
}
}
else
{
    // the everything window was not found.
    // we can optionally RegisterWindowMessage("EVERYTHING_IPC_CREATED")
    // wait for Everything to post this message to all top level windows
}

return FALSE;
}

```

Retrieving results

```

void DisplayPathName(TCHAR *path, TCHAR *filename)
{
    // ...
    // do something with path and filename
    // ...
}

LRESULT CALLBACK window_proc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    DWORD i;
    EVERYTHING_IPC_LIST *list

    switch(uMsg)
    {
        case WM_COPYDATA:
        {
            COPYDATASTRUCT *cds = (COPYDATASTRUCT *)lParam;

            switch(cds->dwData)
            {
                case COPYDATA_QUERYCOMPLETE:

                    list = (EVERYTHING_IPC_LIST *)cds->lpData;

                    for(i=0; i<list->numitems; i++)
                    {
                        DisplayPathName(EVERYTHING_IPC_ITEMPATH(list
                    }

                    return TRUE;
                }

                break;
            }
        }
    }

    return DefWindowProc(hwnd, msg, wParam, lParam);
}

```

Python

An Everything SDK Python example.

Created with Python 3.7.3.

Note: requires Everything running in the background.

test.py

```
import ctypes
import datetime
import struct

#defines
EVERYTHING_REQUEST_FILE_NAME = 0x00000001
EVERYTHING_REQUEST_PATH = 0x00000002
EVERYTHING_REQUEST_FULL_PATH_AND_FILE_NAME = 0x00000004
EVERYTHING_REQUEST_EXTENSION = 0x00000008
EVERYTHING_REQUEST_SIZE = 0x00000010
EVERYTHING_REQUEST_DATE_CREATED = 0x00000020
EVERYTHING_REQUEST_DATE_MODIFIED = 0x00000040
EVERYTHING_REQUEST_DATE_ACCESSED = 0x00000080
EVERYTHING_REQUEST_ATTRIBUTES = 0x00000100
EVERYTHING_REQUEST_FILE_LIST_FILE_NAME = 0x00000200
EVERYTHING_REQUEST_RUN_COUNT = 0x00000400
EVERYTHING_REQUEST_DATE_RUN = 0x00000800
EVERYTHING_REQUEST_DATE_RECENTLY_CHANGED = 0x00001000
EVERYTHING_REQUEST_HIGHLIGHTED_FILE_NAME = 0x00002000
EVERYTHING_REQUEST_HIGHLIGHTED_PATH = 0x00004000
EVERYTHING_REQUEST_HIGHLIGHTED_FULL_PATH_AND_FILE_NAME = 0x00008000

#dll imports
everything_dll = ctypes.WinDLL ("C:\\EverythingSDK\\DLL\\Everything32.dll")
everything_dll.Everything_GetResultDateModified.argtypes = [ctypes.c_int, ctypes.POINTER(ctypes.c_int)]
everything_dll.Everything_GetResultSize.argtypes = [ctypes.c_int, ctypes.POINTER(ctypes.c_int)]

#setup search
everything_dll.Everything_SetSearchW("test.py")
everything_dll.Everything_SetRequestFlags(EVERYTHING_REQUEST_FILE_NAME | EVERYTHING_REQUEST_PATH)

#execute the query
everything_dll.Everything_QueryW(1)

#get the number of results
num_results = everything_dll.Everything_GetNumResults()

#show the number of results
print("Result Count: {}".format(num_results))

#convert a windows FILETIME to a python datetime
#https://stackoverflow.com/questions/39481221/convert-datetime-back-to-windows-64-bit-filetime
WINDOWS_TICKS = int(1/10**-7) # 10,000,000 (100 nanoseconds or .1 microseconds)
WINDOWS_EPOCH = datetime.datetime.strptime('1601-01-01 00:00:00',
                                             '%Y-%m-%d %H:%M:%S')
POSIX_EPOCH = datetime.datetime.strptime('1970-01-01 00:00:00',
                                           '%Y-%m-%d %H:%M:%S')
EPOCH_DIFF = (POSIX_EPOCH - WINDOWS_EPOCH).total_seconds() # 11644473600.0
WINDOWS_TICKS_TO_POSIX_EPOCH = EPOCH_DIFF * WINDOWS_TICKS # 116444736000000000.0

def get_time(filetime):
    """Convert windows filetime winticks to python datetime.datetime."""
    winticks = struct.unpack('<Q', filetime)[0]
    microsecs = (winticks - WINDOWS_TICKS_TO_POSIX_EPOCH) / WINDOWS_TICKS
    return datetime.datetime.fromtimestamp(microsecs)
```



```
#create buffers
filename = ctypes.create_unicode_buffer(260)
date_modified_filetime = ctypes.c_ulonglong(1)
file_size = ctypes.c_ulonglong(1)

#show results
for i in range(num_results):

    everything_dll.Everything_GetResultFullPathNameW(i,filename,260)
    everything_dll.Everything_GetResultDateModified(i,date_modified_filetime)
    everything_dll.Everything_GetResultSize(i,file_size)
    print("Filename: {}\nDate Modified: {}\nSize: {} bytes\n".format(ctypes.wstr
```

Output

```
Filename: C:\Users\python\Desktop\test.py
Date Modified: 2019-06-30 19:38:03.171133
Size: 2976 bytes
```

See Also

<https://stackoverflow.com/questions/252417/how-can-i-use-a-dll-file-from-python>

<https://stackoverflow.com/questions/39481221/convert-datetime-back-to-windows-64-bit-filetime>

<https://stackoverflow.com/questions/2330587/how-to-convert-ctypes-c-long-to-pythons-int>

<https://stackoverflow.com/questions/45178140/returning-string-from-c-function-with-ctypes-gives-large-int-not-char-pointer>

Visual Basic

A Visual Basic example.

Note: requires Everything running in the background.

[Download solution](#)

Form1.vb

```
Public Class Form1

    Public Declare Unicode Function Everything_SetSearchW Lib "d:\dev\everything\sdk
    Public Declare Unicode Function Everything_SetRequestFlags Lib "d:\dev\everythin
    Public Declare Unicode Function Everything_QueryW Lib "d:\dev\everything\sdk\dll
    Public Declare Unicode Function Everything_GetNumResults Lib "d:\dev\everything\
    Public Declare Unicode Function Everything_GetResultFileNameW Lib "d:\dev\everyt
    Public Declare Unicode Function Everything_GetLastError Lib "d:\dev\everything\s
    Public Declare Unicode Function Everything_GetResultFullPathNameW Lib "d:\dev\ev
    Public Declare Unicode Function Everything_GetResultSize Lib "d:\dev\everything\
    Public Declare Unicode Function Everything_GetResultDateModified Lib "d:\dev\eve

    Public Const EVERYTHING_REQUEST_FILE_NAME = &H1
    Public Const EVERYTHING_REQUEST_PATH = &H00000000
    Public Const EVERYTHING_REQUEST_FULL_PATH_AND_FILE_NAME = &H00000000
    Public Const EVERYTHING_REQUEST_EXTENSION = &H00000000
    Public Const EVERYTHING_REQUEST_SIZE = &H00000001


```

```

Public Const EVERYTHING_REQUEST_DATE_CREATED = &H00000002
Public Const EVERYTHING_REQUEST_DATE_MODIFIED = &H00000004
Public Const EVERYTHING_REQUEST_DATE_ACCESSED = &H00000008
Public Const EVERYTHING_REQUEST_ATTRIBUTES = &H00000010
Public Const EVERYTHING_REQUEST_FILE_LIST_FILE_NAME = &H00000020
Public Const EVERYTHING_REQUEST_RUN_COUNT = &H00000040
Public Const EVERYTHING_REQUEST_DATE_RUN = &H00000080
Public Const EVERYTHING_REQUEST_DATE_RECENTLY_CHANGED = &H00000100
Public Const EVERYTHING_REQUEST_HIGHLIGHTED_FILE_NAME = &H00000200
Public Const EVERYTHING_REQUEST_HIGHLIGHTED_PATH = &H00000400
Public Const EVERYTHING_REQUEST_HIGHLIGHTED_FULL_PATH_AND_FILE_NAME = &H00000800

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    Everything_SetSearchW(TextBox1.Text)
    Everything_SetRequestFlags(EVERYTHING_REQUEST_FILE_NAME Or EVERYTHING_REQUEST_ATTRIBUTES)
    Everything_QueryW(1)

    Dim NumResults As UInt32
    Dim i As UInt32
    Dim filename As New System.Text.StringBuilder(260)
    Dim size As UInt64
    Dim ftdm As UInt64
    Dim DateModified As System.DateTime

    NumResults = Everything_GetNumResults()

    ListBox1.Items.Clear()

    If NumResults > 0 Then
        For i = 0 To NumResults - 1

            Everything_GetResultFullPathNameW(i, filename, filename.Capacity)
            Everything_GetResultSize(i, size)
            Everything_GetResultDateModified(i, ftdm)

            DateModified = System.DateTime.FromFileTime(ftdm)

            '
            ListBox1.Items.Insert(i, filename.ToString() & " size: " & size)
            ListBox1.Items.Insert(i, System.Runtime.InteropServices.Marshal.PtrToStringUTF8(System.IntPtr.From(ftdm)))

        Next
    End If

End Sub

End Class

```

Search History

Search history is a list of all your previous searches.

Search history is disabled by default.

When search history is enabled, you can list search suggestions by pressing Ctrl + Space in the search edit.

How do I enable search history?

Search history is disabled by default, to enable search history:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **History** tab.
- Check **Enable search history**.
- Click **OK**.

How do I show a list of search suggestions?

Press Ctrl + Space to display the list of search suggestions.

-or-

Disable search as you type to always show suggestions:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **UI** tab.
- Uncheck **Search as you type**.
- Click **OK**.

-or-

Always show search suggestions:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **History** tab.
- Check **Always show search suggestions**.
- Click **OK**.

-or-

Click the down arrow on the right side the search edit.

How do I add a search to the search history?

Press Enter from the search edit or perform an action on a result.

Where is the search history saved?

Search history is saved to:

```
%APPDATA%\Everything\Search History.csv
```

If [Store settings and data in %APPDATA%\Everything](#) is disabled, the Search History.csv is stored in the same location as Everything.exe.

This file is updated when closing the Everything Search window.

How do I clear my search history?

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **History** tab.
- In the Search History group box, click **Clear Now**.
- Click **OK** to clear.
- Click **OK**.

Browsing Search History

Previous searches can be shown from clicking on the dropdown button to the right of the search box.

With Everything 1.4 you can now browse and edit your entire search history (if enabled).

To browse and edit your Search History:

- In "Everything", from the View menu, from the Goto menu, click Show All History.

Searching

Use the [search edit](#) to type in a search.

Type in a partial filename, files and folders matching your partial filename search will be instantly displayed in the [result list](#).

For example, to find files and folders with the text **reports** in the filename, search for:

```
reports
```

To limit the results to a drive, simply search for that drive, for example, to find files and folders on your **D:** drive, search for:

```
d:
```

Include the path to limit the results to a folder, for example, to find files and folders in the folder **D:\Downloads**, search for:

```
d:\downloads\
```

Use double quotes to escape spaces, for example, to find files and folders in **C:\Program Files**, search for:

```
"c:\program files\"
```

Use \ to search for partial paths, for example, to find files and folders in a folder ending with **documents**, search for:

```
documents\
```

Use a space to combine searches, for example, to find files and folders containing the text **order**, located in the folder starting with **work**, search for:

```
\work order
```

Syntax

Operators

<i>space</i>	AND
	OR
!	NOT
< >	Grouping
" "	Search for an exact phrase

Wildcards

*	Matches zero or more characters.
?	Matches one character.

Wildcards match the *whole* filename. Disable [Match whole filename when using wildcards](#) to match wildcards anywhere in the filename.

Macros

quot:	Literal double quote "
apos:	Literal apostrophe '
amp:	Literal ampersand &
lt:	Literal less than <
gt:	Literal greater than >
#<n>:	Literal unicode character <n> in decimal.
#x<n>:	Literal unicode character <n> in hexadecimal.
audio:	Search for audio files.
zip:	Search for compressed files.
doc:	Search for document files.
exe:	Search for executable files.
pic:	Search for picture files.
video:	Search for video files.

Modifiers

Functions and regular search terms can be prefixed with a modifier.

ascii: utf8: noascii:	Enable or disable fast ASCII case comparisons.
case: nocase:	Match or ignore case.
diacritics: nodiacritics:	Match or ignore accent marks.
file: files: nofileonly:	Match files only.
folder: folders: nofolderonly:	Match folders only.
path: nopath:	Match the full path and file name or just the filename.
regex: noregex:	Enable or disable regex.

wfn:	Match the whole filename or match anywhere in the filename.
wholefilename:	
nowfn:	
nowholefilename:	
exact:	
wholeword:	Match whole words or match anywhere in the filename.
ww:	
nowholeword:	
noww:	
wildcards:	Enable or disable wildcards.
nowildcards:	

Functions

album:<album>	Search for the ID3 or FLAC album.
artist:<artist>	Search for the ID3 or FLAC artist.
attrib:<attributes> attributes:<attributes>	Search for files and folders with the specified file attributes.
bitdepth:<bitdepth>	Find images with the specified bits per pixel.
child:<filename>	Search for folders that contain a child file or folder with a matching filename.
childcount:<count>	Search for folders that contain the specified number of subfolders and files.
childfilecount:<count>	Search for folders that contain the specified number of files.
childfoldercount:<count>	Search for folders that contain the specified number of subfolders.
comment:<comment>	Search for the ID3 or FLAC comment.
content: ansicontent: utf8content: utf16content: utf16becontent:	Search file content.
count:<max>	Limit the number of results to max.
dateaccessed:<date> da:<date>	Search for files and folders with the specified date accessed.
datecreated:<date> dc:<date>	Search for files and folders with the specified date created.
datemodified:<date> dm:<date>	Search for files and folders with the specified date modified.
daterun:<date> dr:<date>	Search for files and folders with the specified date run.
depth:<count> parents:<count>	Search for files and folders with the specified folder depth.

dimension:<width>x<height>	Find images with the specified width and height.
dupe: namepartdupe: attribdupe: dadupe: dcdupe: dmdupe: sizedupe:	Search for duplicated files.
empty:	Search for empty folders.
endwith:<text>	Filenames (including extension) ending with text.
ext:<list>	Search for files with a matching extension in the specified semicolon delimited extension list.
filelist:<list>	Search for a list of file names in the specified pipe () delimited file list.
filelistfilename:<filename>	Search for files and folders belonging to the file list filename.
frn:<frnlist>	Search for files and folders with the specified semicolon delimited File Reference Numbers.
fsi:<index>	Search for files and folders in the specified zero based internal file system index.
genre:<genre>	Search for the ID3 or FLAC genre.
height:<height>	Search for images with the specified height in pixels.
len:<length>	Search for files and folders that match the specified filename length.
orientation:<type>	Search for images with the specified orientation (landscape or portrait).
parent:<path> infolder:<path> nosubfolders:<path>	Search for files and folders in the specified path, excluding subfolders.
recentchange:<date> rc:<date>	Search for files and folders with the specified recently changed date.
root:	Search for files and folders with no parent folder.
runcount:<count>	Search for files and folders with the specified run count.
shell:<name>	Search for a known shell folder name , including subfolders and files.
size:<size>	Search for files with the specified size in bytes.
startswith:<text>	Search for filenames starting with text.
title:<title>	Search for the ID3 or FLAC title.
type:<type>	Search for files and folders with the specified type.
width:<width>	Search for images with the specified width in pixels.

Function syntax

<code>function:value</code>	Equal to value.
<code>function:<=value</code>	Less than or equal to value.
<code>function:<value</code>	Less than value.
<code>function:=value</code>	Equal to value.
<code>function:>value</code>	Greater than value.
<code>function:>=value</code>	Greater than or equal to value.
<code>function:start..end</code>	Is in the range of values from start to end.
<code>function:start-end</code>	Is in the range of values from start to end.

Size Syntax:

`size[kb|mb|gb]`

Size Constants:

<code>empty</code>	
<code>tiny</code>	<code>0 KB < size <= 10 KB</code>
<code>small</code>	<code>10 KB < size <= 100 KB</code>
<code>medium</code>	<code>100 KB < size <= 1 MB</code>
<code>large</code>	<code>1 MB < size <= 16 MB</code>
<code>huge</code>	<code>16 MB < size <= 128 MB</code>
<code>gigantic</code>	<code>size > 128 MB</code>
<code>unknown</code>	

Date Syntax:

`year`

`month/year` or `year/month` depending on locale settings

`day/month/year`, `month/day/year` or `year/month/day` depending on locale settings

`YYYY[-MM[-DD[Thh[:mm[:ss[.sss]]]]]]`

`YYYYMM[DD[Thh[mm[ss[.sss]]]]]`

Date Constants:

`today`

`yesterday`

`<last|past|prev|current|this|coming|next><year|month|week>`

`<last|past|prev|coming|next><x><years|months|weeks>`

`<last|past|prev|coming|next><x><hours|minutes|mins|seconds|secs>`

`january|february|march|april|may|june|july|august|september|october|november|december`

jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec
sunday|monday|tuesday|wednesday|thursday|friday|saturday
sun|mon|tue|wed|thu|fri|sat
unknown

Attribute Constants:

A	Archive
C	Compressed
D	Directory
E	Encrypted
H	Hidden
I	Not content indexed
L	Reparse point
N	Normal
O	Offline
P	Sparse file
R	Read only
S	System
T	Temporary
V	Device

Search options

Search options can be toggled from the [Search menu](#).

Enabled search options are displayed in the right of the status bar. Double click the status bar part to disable the search option. Right click the status bar to show a list of search options.

Match Case

Match or ignore case.

Match Whole Words

Match whole words or allow the match to occur any where in the filename.

Match Path

match against the full path and filename or just the filename.

Match Diacritics.

Match or ignore accent marks.

Enable Regex

Enable or disable [regular expressions](#).

Everything

Audio

Compressed

Document

Executable

Folder

Picture

Video

Match the specified type only. See [Filters](#) for more information.

Advanced Search

The screenshot shows the 'Advanced Search' dialog box with the following settings:

- File names containing...**
 - all these words:** blue lake photo
 - ☐ Match case ☐ Match whole words ☐ Match diacritics
 - this exact phrase:** summer holiday
 - ☐ Match case ☐ Match whole words ☐ Match diacritics
 - any of these words:** (empty field)
 - ☐ Match case ☐ Match whole words ☐ Match diacritics
 - none of these words:** small
 - ☐ Match case ☐ Match whole words ☐ Match diacritics
- A word or phrase in the file:** (empty field)
 - ☐ Match case ☐ Match whole words ☐ Match diacritics
 - Content type: Auto detect
- Located in:** C:\Users\limited\Pictures (with a 'Browse...' button)
 - ☒ Include subfolders
- Date modified:** 24/05/2016 to 10/10/2016
- Size:** 1 MB to 100 MB

Buttons: OK, Cancel

Use the Everything advanced search to help build a more complex search.

To show the advanced search dialog:

- In "**Everything**", from the **Search** menu, click **Advanced Search**.

Content Searching

Warning: content searching is extremely slow.

File content is not indexed.

Please combine content: functions with other filters for the best performance.

Content search functions:

<code>content:<text></code>	Search file content using the associated iFilter . If no iFilter exists UTF-8 content is used.
<code>ansicontent:<text></code>	File contents are treated as ANSI text.
<code>utf8content:<text></code>	File contents are treated as UTF-8 text.
<code>utf16content:<text></code>	File contents are treated as UTF-16 (Unicode) text.
<code>utf16becontent:<text></code>	File contents are treated as UTF-16 (Big Endian) text.

Example, find emails, modified this week, containing the text "banana":

```
*.eml dm:thisweek content:banana
```

* Note: content: is used last in the search above, this means Everything will only search the content of files that matched the previous search `*.eml dm:thisweek`

ID3 Tags

The following search functions can be used to search for ID3 tags and FLAC tags:

<code>track:<track></code>	Track number or track range.
<code>year:<year></code>	Tear or year range.
<code>title:<title></code>	Song title.
<code>artist:<artist></code>	Song Artist.
<code>album:<album></code>	Album name.
<code>comment:<comment></code>	Track comment.
<code>genre:<genre></code>	Track genre.

ID3v1 is fully supported.

ID3v2, ID3v2.3 and ID3v2.4 are loosely supported.

ID3 tags and FLAC tags are not indexed. Searching for ID3 tags and FLAC tags is slow, combine with other searches for the best performance.

Only ID3 tags in mp3 files are currently supported.

Examples:

```
year:2002..2005
genre:electronic
regex:album:^[a-n]
wildcards:title:red*
track:>10
year:>=2000
```

Image information

The following functions can be used to search for image dimensions

width:<width>	The width of the image in pixels.
height:<height>	The height of the image in pixels.
dimensions:<width>x<height>	The width and height of the image in pixels. Use a x to separate the width and height.
orientation:<type>	<type> can landscape or portrait.
bitdepth:<bitdepth>	Find images with the specified bits per pixel.

Image information is not indexed. Searching for image information is slow, combine with other searches for the best performance.

Only jpg, png, gif and bmp files are supported.

Examples:

```
width:>2560
width:800..1920
height:600..1080
dimensions:800x600..1920x1080
```

Duplicated Files

The following functions can be used to search for duplicated files.

dupe:	Find files and folders with the same filename.
attribdupe:	Find files and folders with the same attributes. Sort by attributes for the best results.
dadupe:	Find files and folders with the same date accessed. Sort by date accessed for the best results.
dcdupe:	Find files and folders with the same date created. Sort by date created for the best results.
dmdupe:	Find files and folders with the same date modified. Sort by date modified for the best results.
namepartdupe:	Find files and folders with the same name part (excluding extension).
sizedupe:	Find files and folders with the same size. Sort by size for the best results.

Duplicates are found in the ENTIRE index, not the current results.
Search and sort results by the same dupe type for the best results.

Everything does not check file contents. Use the dupe finding functions as a guide only.

Examples:

```
dupe: .mp4  
size:>1gb sizedupe:
```

Filters

Filters are predefined searches. Only one filter can be active at a time. Filters can be toggled from the [Search menu](#).

To create a new filter:

- In **Everything**, from the **Search** menu, click **Add to filters....**
- Type in a **Name**.
- Click **OK**.

To edit an existing filter:

- In **Everything**, from the **Search** menu, click **Organize filters....**
- Select a filter.
- Click **Edit**.

To create a search macro:

- In **Everything**, from the **Search** menu, click **Organize filters....**
- Select a filter.
- Click **Edit**.
- Set macro to your search shortcut, for example: photos
- The photos: search will now be replaced with this filter's search.
- Click **OK**.
- Click **OK**.

To create a filter keyboard shortcut:

- In **Everything**, from the **Search** menu, click **Organize filters....**
- Select a filter.
- Click **Edit**.
- Click to the right of Keyboard shortcut and press a new keyboard shortcut.
- Click **OK**.
- Click **OK**.

Filters can be accessed from the Search menu, filter bar (View -> Filters), right clicking the status bar, filter macro or filter keyboard shortcut.

Bookmarks

Bookmarks can save the current search, filter, sort and index.

To create a new bookmark from the current search:

- In **Everything**, from the **Bookmarks** menu, click **Add to bookmarks....**
- Type in a **Name**.

- Click **OK**.

To organize bookmarks:

- In **Everything**, from the **Bookmarks** menu, click **Organize bookmarks....**

Home Search

The home search is the default search when you first open an Everything search window.

The home search can be opened by pressing [Alt + Home](#).

To change the home search settings:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the ["Home"](#) tab.
- Set the desired search, search options, sort order and view options.
- Click **OK**.

Macros

Custom macros can be defined by filters and bookmarks.

To create a custom filter macro:

- Create a [new filter](#) or [edit an existing one](#).
- Type in a macro name, for example:

foo

- Click **OK**.
- Click **OK**.
- Searching for foo: will now be replaced by the filter's search.

Regex

Regex overrides the search syntax. Search operators, wildcards, macros, modifiers and functions do not work in regex mode.

Regex must be enabled from the [Search menu](#) or prefix the search with `regex:`

When using the regex: modifier, please escape `|` and space with double quotes.

<code>a b</code>	Matches a or b
<code>gr(a e)y</code>	Matches gray or grey
<code>.</code>	Matches any single character
<code>[abc]</code>	Matches a single character a, b or c
<code>[^abc]</code>	Matches any single character except a, b or c
<code>[a-z]</code>	Matches a single character in the range a to z
<code>[a-zA-Z]</code>	Matches a single character in the range a to z or A to Z
<code>^</code>	Matches the start of the filename

\$	Matches the end of the filename
*	Matches the preceding element zero or more times
?	Matches the preceding element zero or one times
+	Matches the preceding element one or more times
{x}	Matches the preceding element x times
{x, }	Matches the preceding element x or more times
{x, y}	Matches the preceding element between x and y times
\	Escape special character

Search Commands

The following search commands are special searches that can be activated by typing in the search and pressing ENTER.

Search	Action
about:	Show the About dialog.
about:config	Open your Everything.ini
about:credits	Show the Everything credits dialog.
about:home	Open the home search .
about:licence about:license	Show the Everything licence agreement.
about:options about:preferences	Show the Everything options .
/close	Close the Everything search window.
/closeall	Close all Everything search windows.
/command <id>	Run the specified command ID .
/config_save	Show a save as dialog to backup the Everything.ini .
/config_save <filename>	Backup the Everything.ini to the specified filename.
/config_load	Show a open file dialog to load the Everything.ini .
/config_load <filename>	Load the Everything.ini from the specified filename.
/debug /console	Enable debug mode.
/debug_log	Enable debug mode and start debug logging.
/help	Show the help window.

<code>/monitor_pause</code>	Stops NTFS, ReFS, file lists and folder monitors. Monitors are not started again until specified with <code>/monitor_resume</code> .
<code>/monitor_resume</code>	Starts NTFS, ReFS, file lists and folder monitors.
<code>/quit</code> <code>/exit</code>	Exit Everything
<code>/rebuild</code> <code>/reindex</code>	Force a rebuild
<code>/restart</code>	Restart Everything
<code>/verbose</code>	Enable verbose debugging.
<code>/update</code>	Update all folder indexes now.
<code>/update <folder index></code>	Update the specified folder index by path now.
<code>/<ini option></code>	Show the current setting for the specified ini option in the statusbar. Use the TAB key to auto complete.
<code>/<ini option>=<value></code>	Set the specified ini option to the specified value. The new value is shown in the status bar if successful.

For example, to change the status bar selected item format to full path and filename, type in the following search and press ENTER:

```
/statusbar_selected_item_format=$f
```

Changing some [ini options](#) will require a restart.

Limiting results

To limit the number of results shown, use the count: search function.

For example to limit the number of results shown to 100, include at the start of your search:

```
count:100
```

Examples

Search for ABC and 123:

```
ABC 123
```

Search for ABC or 123:

```
ABC|123
```

Search for everything except ABC:

```
!ABC
```


Search for uppercase ABC

```
case:ABC
```

Search for mp3 files:

```
*.mp3
```

Search for mp3 files on the D: or E: drive:

```
d:|e: *.mp3
```

Search for jpg or png files on the D: drive:

```
d: *.jpg|*.png
```

Search for files or folders with no extension:

```
!.
```

Search for files only:

```
file:
```

Search for folders only:

```
folder:
```

Limit the search to a single folder:

```
parent:c:\windows
```

Find files larger than 1MB:

```
size:>1mb
```

Instantly find files that have changed today:

```
dm:today
```

Find files and folders modified this week:

```
dm:thisweek
```

Find files and folders modified from the 1st August 2014 to 31st August 2014:

```
dm:1/8/2014-31/8/2014
```

or:

```
dm:8/1/2014-8/31/2014
```

Depending on your locale settings.

Find folders in D:\music that do not contain an mp3 file:

```
d:\music\ !child:mp3
```

Find filenames with non-ASCII characters:

```
regex: [^\x00-\x7f]
```

See also

- [Search menu](#)
- [Search command line options](#)
- [Search options](#)
- [Search keyboard shortcuts](#)

Supported Languages

The following translations will work with any version of "Everything" with multiple language support.

English is used for new untranslated features.

Language	Translator	Version
Afrikaans	Samuel Murray	1.2.1.371
Bosanski	Nedim Laletovic	1.2.1.371
Čeština	jvh	1.4.1.1000
Deutsch	Michael Banz & Thomas Knoop	1.4.1.1000
English (AU)	David Carpenter	1.4.1.1000
English (UK)	David Carpenter	1.4.1.1000
English (US)	David Carpenter	1.4.1.1000
Español	Sacher Collado Benitez (ScB)	1.4.1.1000
Español (Venezuela)	Ricardo A. Rivas	1.4.1.1000
Estonia	Olav Mägi	1.2.1.371
Français	Jikoo and Colok and Pierre le Lidgeu	1.3.4.686
Hrvatski	de4dcod3	1.2.1.371
Italiano	Stefano Piccardi	1.4.1.1000
Magyar	Zsolt Péter Basák (Shiki) & Zoltán Szabó (Bird)	1.2.1.371
Nederlands	Jeroen Westera	1.4.1.1000
Norsk bokmål	Imre Kristoffer Eilertsen	1.4.1.1000
O'zbekcha	Zafar Shamsiddinov	1.4.1.1000

Polski	Zbigniew Zienko & superjohn & Piotr J. Balwierz	1.4.1.1000
Português (Brasil)	Igor Rückert	1.4.1.1000
Română	Alexandru Bogdan Munteanu	1.4.1.1000
Shqip	Sazan Kërçuku	1.2.1.371
Slovenčina	Sepp Winkler	1.2.1.371
Srpski	ozzii	1.4.1.1000
Suomi	VeikkoM	1.4.1.1000
Svenska	Åke Engelbrektson	1.4.1.1000
Tiếng Việt	Lê Sỹ Đạt	1.2.1.371
Türkçe	Burak Yavuz (BouRock) & Koray KIRCAOGLU	1.4.1.1000
Ελληνικά	geogeo.gr	1.4.1.1000
Македонски	01 Vlatce	1.2.1.371
Русский	Dmitry Yerokhin	1.4.1.1000
Српски	ozzii	1.4.1.1000
עברית	Stamimail & shai horn	1.4.1.1000
عربي	alyzeedus dwadeh	1.2.1.371
فارسی	Reza Jelodar	1.2.1.371
தமிழ்	அனிஷ் பிரபு தி.(Anish Prabu T.), Ashwin and Aarthi	1.4.1.1000
ภาษาไทย (TH)	นิติกร คำภักดิ์	1.4.1.1000
한국어	jeiea & dis1	1.4.1.1000
日本語	koga & miyano	1.4.1.1000
简体中文	WanderMax & xbeta	1.4.1.1000
繁體中文	GT Wang & Carl Yuan & rexx & xbeta	1.4.1.1000

Download language pack for "*Everything*"

- [Everything.Ing.zip](#)

Installing the language pack

- Download the language pack [Everything.Ing.zip](#)
- Unzip **Everything.Ing** from the language pack into the folder where "Everything" is installed.
- Restart "**Everything**".
- In "**Everything**", from the **Tools** menu, click **Options**.

- Click the **General** tab.
- Select your language from the **Language** dropdown list.
- Click **OK**.
- In the "language change" popup, click **OK** to restart "**Everything**".

See also

- [Translating](#)

Translating

Please check [Supported languages](#) for a list of supported translations.

Download the English translation template from [Template.zip](#)

Unzip the template to an empty folder.

Open the file `Template.txt` with notepad.

Translate each string in the template to the new language.

Syntax

Each string will appear on a single line and has the following syntax:

```
key = "value"
```

For example: the string in the English translation template:

```
LOADING_DATABASE = "Loading database..."
```

For example: Translating the `LOADING_DATABASE` string to Italian:

```
LOADING_DATABASE = "Caricamento della base dati..."
```

Comments

// Start a line comment (the rest of the line is ignored)

; Start a line comment (the rest of the line is ignored)

Comments are ignored by the language compiler.

Escaping characters

Escape	Description
<code>\n</code>	new line
<code>\r</code>	line return

\t	tab
\"	escape a double quote (")
\\	a single back-slash (\)

printf

%s and %d must be in the same order as they appear in the English translation template.

Menu Keys

place a & in front of the letter for the menu key.

This only applies to strings in the English translation template that have a &

Saving

Please save the template as UTF-8

Updating

I will notify you by email when translations require updating.

An up to date English translation template is always available from [Template.zip](#)

New and changed translation strings are added to the end of the template.

If you have an existing translation you will need to add the updated translations from [Template.zip](#) to the end of your translation template.

Your translation is compatible with previous and future versions of Everything.

Strings that have not yet been translated will appear in English.

Changes

Some strings may vary between Everything versions. Usually help information is expanded upon or alt shortcut keys added.

Please check the Changes.txt in the [Template.zip](#) file for a list of changes that may need updating in your existing translation template.txt.

Language compiler

Please download the language pack compiler if you would like to test your translation with "*Everything*".

An up to date language pack compiler is always available from [makeIng.zip](#)

Instructions

- Download the language pack compiler and extract to an empty folder.
- Copy your translation template to this folder.
- Run the compiler.
- Exit "*Everything*".
- Copy the generated `Everything.lng` to your "*Everything*" installation folder.
- Restart "*Everything*".
- Select the correct language in "*Everything*"

Note: makeIng.exe compiles *.txt .

Warnings

Ignore warnings for duplicated definitions.

The last definition will be used for duplicates.

You can optionally remove the first references of the duplicates to remove these warnings.

Warnings for undefined strings should be fixed.

Warnings for empty strings should be fixed.

Credit

Please add your name to the credit list for your translation.

Your name will be added to the English translation template to be updated for the next release of "*Everything*".

Your name will also be included on the website.

You can optionally have your name linked to an email address or website.

Help & Support

Please contact [David Carpenter](#) for assistance.

Everything 1.4 Changes

Please check the latest English translation template from [Template.zip](#) for a list of changes to strings from Everything 1.3. These strings should be updated in your translation template.

Website

To translate the voidtools website, please see [makevoidtools](#).

Thanks

Thanks to those that have translated "*Everything*".

Thank you for supporting "*Everything*"!

See also

- [Supported languages](#)

Troubleshooting

Missing files

Corrupt files and folders are automatically excluded by "*Everything*". Please check your disk(s) for errors.

To check your disk(s) for errors:

- In **My Computer**, for each volume:
- **Right click** and select **Properties**.
- Click the **Tools** tab.
- Click **Check now....**
- Select **Automatically fix file system errors**.
- Click **Start**.
- If prompted to schedule the disk check the next time you restart the computer select Yes.

Rebuilding the database

To force "*Everything*" to rebuild the database:

- In "*Everything*", from the the **Tools** menu, click **Options**.
- Click the **Indexes** tab.
- Click **Force rebuild**.
- Click **OK**.

Recreate the USN Journal

To force the NTFS driver to rebuild the USN Journal:

- In "*Everything*", from the the **Tools** menu, click **Options**.
- Click the **NTFS** tab.
- Select the NTFS volume .
- Uncheck **Enable USN Journal Logging**.

- Click **Apply**.
- Check **Enable USN Journal Logging**.
- Click **OK**.

HTTP Server does not start

This is most commonly caused by another service using port 80.

Try to change the HTTP server port.

For example: to change the HTTP server port to 8080:

- In "Everything", from the Tools menu, click Options.
- Under the HTTP tab, change the HTTP Server Port to 8080.
- Click OK.

To connect to this server:

- In your web browser, enter `http://localhost:8080`
- Change localhost to the server name for remote access.

Debug mode.

The debug console can be toggled with the Ctrl + ` key.

See viewtopic.php?f=6&t=1713 for more information.

Support

Please check the [forums](#) or [contact voidtools](#) for support.

Uninstalling Everything

An uninstaller is created when Everything is installed.

To completely uninstall Everything with the uninstaller:

- From the **Control panel**, open **Programs and Features**.
- Right click **Everything**.
- Select **Uninstall/Change**
- Select the language of the uninstaller if the multilingual installer was used to install Everything.
- Click **Uninstall**.

To completely uninstall the portable version of Everything:

- From a command prompt, run:

```
Everything.exe -uninstall
```

- Delete Everything.exe.

System modifications

The follow settings modify the system:

- Store settings and data in %APPDATA%\Everything
- Start Everything on system startup
- Everything service
- Show shell context menus
- Start menu shortcuts
- Desktop shortcuts
- Quick Launch shortcuts
- URL protocol
- EFU file association.

To change these settings:

- In **Everything**, from the **Tools** menu, click **Options**.
- Click the **General** tab.

All system modifications made by Everything are removed when Everything is uninstalled.

Files created by Everything

Everything may create the following files in %LOCALAPPDATA%\Everything:

- Everything.db
- Everything.db.tmp

These files are deleted when Everything is uninstalled for all users.

Everything may create the following files in %APPDATA%\Everything:

- Everything.ini
- Bookmarks.csv
- Filters.csv
- Search History.csv
- Run History.csv
- Everything.ini.tmp
- Errorlog.txt
- Bookmarks.csv.tmp
- Filters.csv.tmp
- Search History.csv.tmp
- Run History.csv.tmp
- Logs\HTTP_Server_Log.txt
- Logs\ETP_Server_Log.txt

These files are deleted when Everything is uninstalled for the current user.

If [Store settings and data in %APPDATA%\Everything](#) is disabled, these files are stored in the same location as Everything.exe.

These files have different filenames if [Multiple Instances](#) is used.

If you use [Named instances](#) and wish to uninstall, you will have to uninstall each instance separately with Everything.exe -uninstall -instance <Instance Name>

Manual uninstall

- Make sure Everything and the Everything service is not running.
- Delete the folder C:\Program Files\Everything

- Delete the folder C:\Program Files (x86)\Everything
- Delete the folder %APPDATA%\Everything
- Delete the registry value Everything in HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- Delete the registry value Everything in HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
- Delete the registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Everything
- Delete the registry key HKEY_CLASSES_ROOT\Folder\shell\Search Everything...
- Delete the folder %APPDATA%\Microsoft\Windows\Start Menu\Programs\Everything
- Delete the folder %USERPROFILE%\Start Menu\Programs\Everything
- Delete the shortcut %USERPROFILE%\Desktop\Search Everything.Ink
- Delete the shortcut %APPDATA%\Microsoft\Internet Explorer\Quick Launch\Search Everything.Ink
- Delete the registry key HKEY_CLASSES_ROOT\ES
- Delete the registry key HKEY_CLASSES_ROOT\efu
- Delete the registry key HKEY_CLASSES_ROOT\Everything.FileList
- Delete the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Everything
- Delete the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\Everything
- Delete the registry key HKEY_LOCAL_MACHINE\SOFTWARE\voidtools\Everything
- Remove the empty registry key HKEY_LOCAL_MACHINE\SOFTWARE\voidtools

See also

- [Multiple Instances](#)

What's New

New features and changes in Everything 1.4

- [Size, Dates and Attributes Indexing](#)
- [Fast Sorting](#)
- [Thumbnails](#)
- [Preview Pane](#)
- [ReFS Support](#)
- [Advanced Search](#)
- [Multi-file Renaming](#)
- [Content Searching](#)
- [Fast Reindexing](#)

New features and changes in Everything 1.3

- [Everything service](#)
- [Recent changes](#)
- [Run history](#)
- [Search history](#)
- [File Lists](#)
- [Folder Indexing](#)

- [Match Diacritics](#)
- [Filters](#)
- [Bookmarks](#)
- [Filter bar](#)
- [Search functions](#)

Improvements

- [HTTP](#)
- [ETP](#)
- [Options](#)

[About voidtools](#)

[Language](#)

© 2021 voidtools - [Privacy](#)