

WinAPI Search

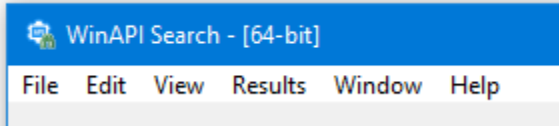
Quick Start Manual

Table of Contents

- [Intro](#)
- [Installation](#)
- [Uninstallation](#)
- [Basic Operation](#)
 - [Main Concepts](#)
 - [“Look In” Filter](#)
 - [Dependency Walker](#)
 - [Regex Filters](#)
 - [“Import Libs” Filter](#)
 - [PE Header Filters](#)
 - [Error Code Search](#)
 - [Error Message Search](#)
 - [Undecorate C++ Symbols](#)
- [Settings](#)
 - [General](#)
 - [Error Search](#)
 - [Message Boxes](#)
- [Bug Reports](#)
- [Contact & Downloads](#)

Intro

Since this application was primarily designed for developers & Windows OS researchers, I will not dwell too much on the details of the UI. Keep in mind though that there are two “flavors” of this app: 64-bit and 32-bit. Due to the nature of this app, those two “flavors” are quite different from one another. You would normally use the appropriate bitness of this app to match the bitness of the binary file(s) that you want to research. You can tell them apart by looking at the app’s title bar:



Installation

You can download the WinAPI Search app from [here](#). The app does not need to be installed in a traditional Windows-sense. It is what you’d call a *portable executable*, which means that you can run it from any place you want. Simply un-ZIP two executable files from the downloaded package to start using the app.

What I would suggest though is that you place both `WinApiSearch32.exe` and `WinApiSearch64.exe` into the same folder. This way they can interact with each other and you will be able to run either bitness of the app from its menu commands and keyboard shortcuts.

Uninstallation

Uninstallation of the WinAPI Search app is also quite simple. Because there’s no installer involved in the installation, to remove it simply delete `WinApiSearch32.exe` and `WinApiSearch64.exe` files and you’re done!

In case you also want to remove all the System Registry keys that the app used during its operation, before deleting the files, like I suggested above, while the app is still running, go to Help -> Maintenance -> “Delete App’s Temp Files & Registry Settings” option. It will delete all those things for you. After that you can go ahead and delete the two files as I suggested above.

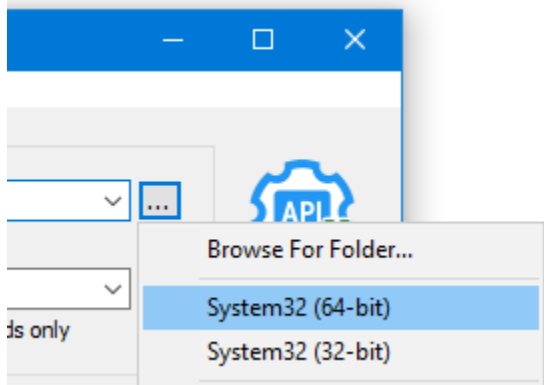
Basic Operation

Let me walk you through and show the features of the WinAPI Search app:

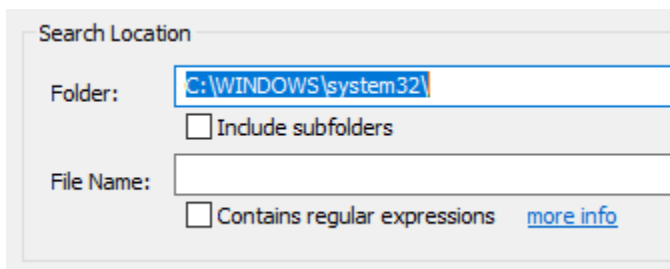
- **Main Concepts**

As its name suggests, the app can search for a specific Win32 API (or a function) by its name (or parts of its name) in a single or multiple set of binary files and folders. This was the main purpose of writing this app.

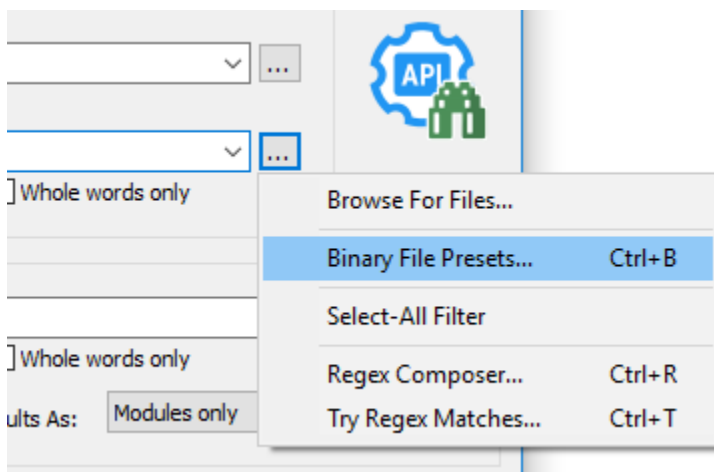
Example: Select where you want to search. Say, I'll go with the entire system folder for 64-bit apps:



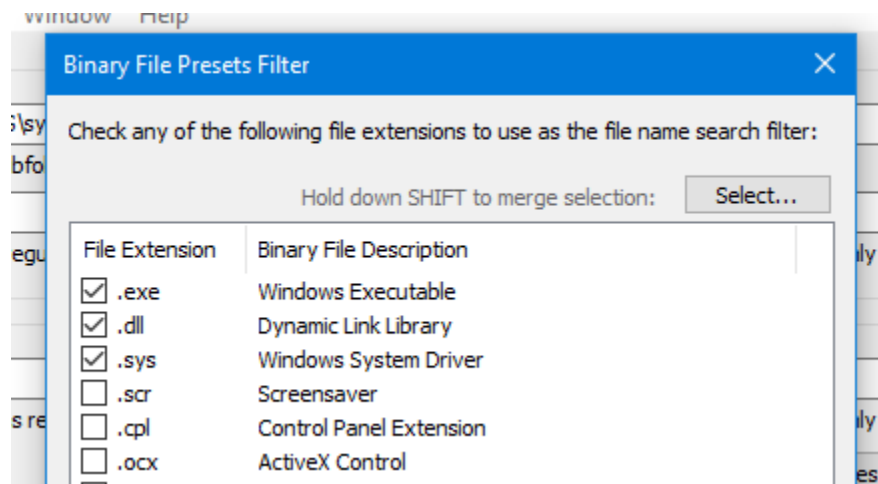
Then specify the File Name(s) that you want to search. Or you can leave it blank to search all files:



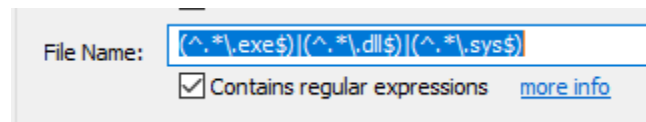
Keep in mind though that in this case WinAPI Search app will attempt to open all available files as if they were binary PE files, which may slow things down. So instead, I will specify which binary files I want to search:



and then select DLLs, EXEs and SYS files that I think will satisfy my search:

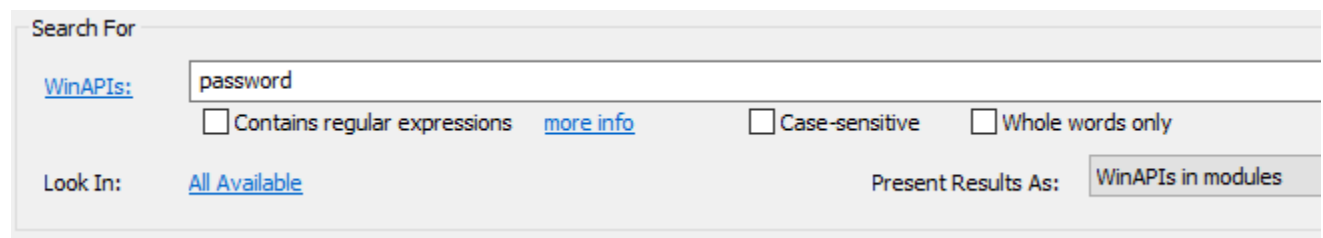


That will create the following Regex filter for me:



(By the way, to learn more about Regex syntax, click “more info” link underneath.)

Then say, I want to know which APIs contain word “password” in them. To speed things up, I won’t be using Regex for this simple filter:



Then click Search and wait for results that will be shown in the list below:

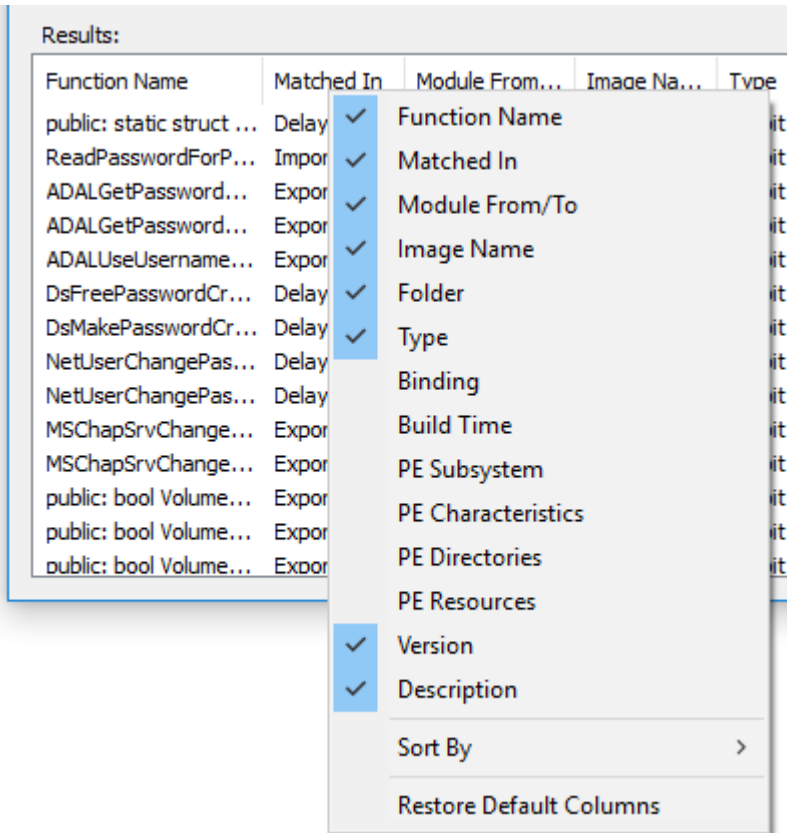
Search For					
WinAPIs: password					
<input type="checkbox"/> Contains regular expressions more info <input type="checkbox"/> Case-sensitive <input type="checkbox"/> Whole words only					
Look In: All Available			Present Results As: WinAPIs in modules		

Results: C:\WINDOWS\system32\MapControlCore.dll					
Function Name	Matched In	Module From/To	Image Name	Type	Version ^
I_NetServerPasswordSet2	Export	logoncli.dll	logoncli.dll	64-bit	
I_NetServerTrustPasswordsGet	Export	logoncli.dll	logoncli.dll	64-bit	
MsvpMakeSecretPasswordNT5	Import	NtlnShared.dll	LsaIso.exe	64-bit	
MsvpCredentialToCachePasswords	Import	NtlnShared.dll	LsaIso.exe	64-bit	

The list will show (from left to right): WinAPI “function name” that matched your search filters; where it “matched in”, meaning, an export, import table, etc.; “Module From/To” in connection with the previous column (for instance, if it’s an “Export” than this is the column the API is exported from, or if it’s an “import” this is a module

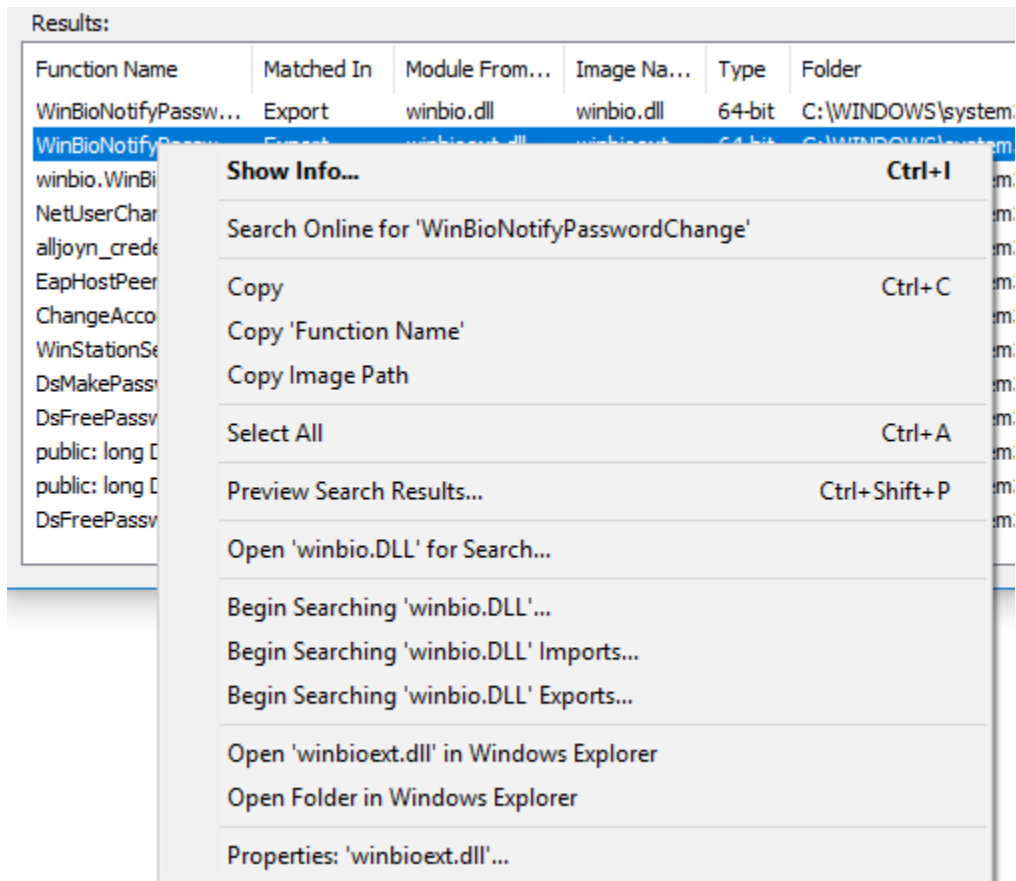
the API is imported from, etc.) The “image name” is the file name that matched API is located in. Then it will show the “type”, meaning bitness of the image module, “version” is the version of the file, and “description” is the textual description of the file.

By the way, if you right-click on the search result list header, you can select which columns you want to show there:



Additionally, you can sort by a specific column if you click on its header in the list. Clicking it again will reverse the order of search.

Note that you can right-click any entry in the Results list to see additional context menu options for it. (This menu will change depending on what is selected and which list column you clicked in.) Here’s an example:

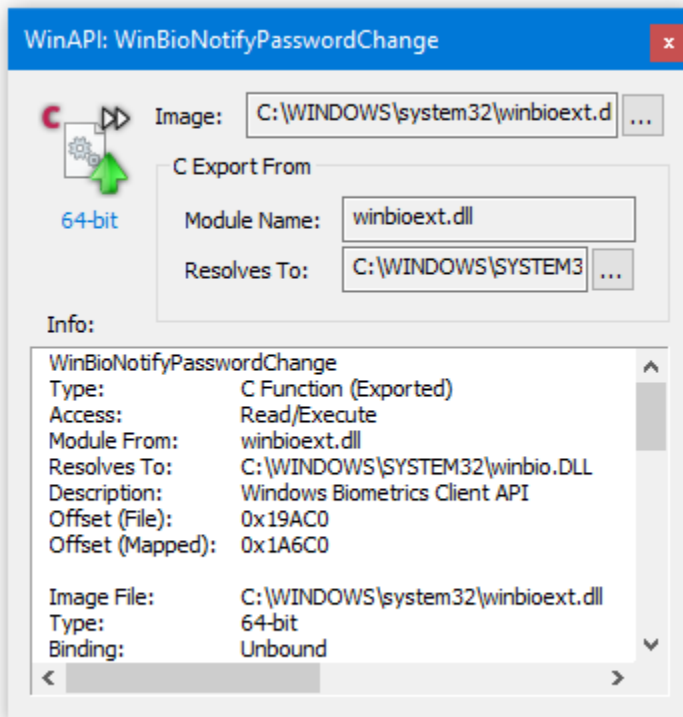


Most options in this menu are self-explanatory. Here are just a few to boot:

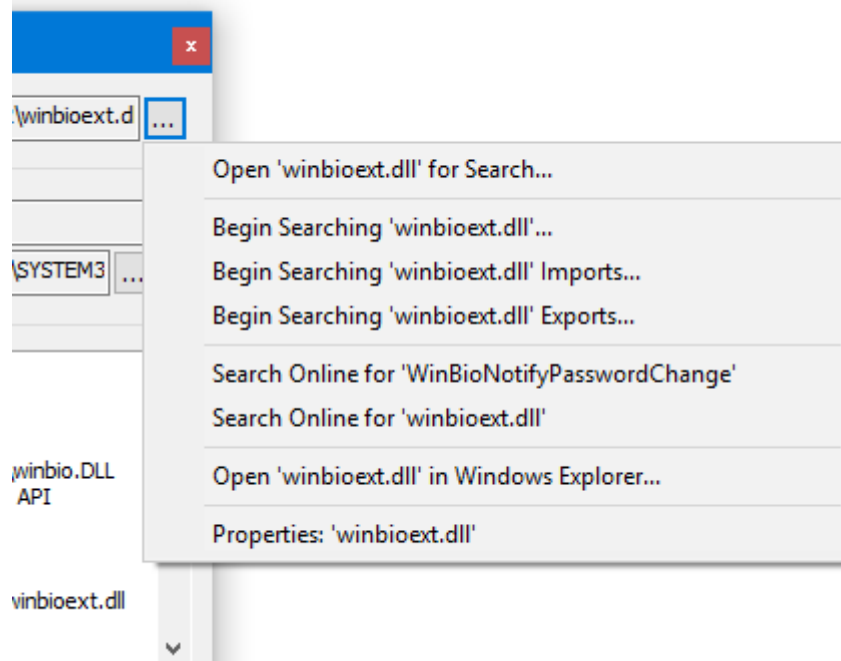
- **Search Online for 'Item'** – will change depending on which column you clicked. It will open your web browser and initiate an online search for that specific item. This is especially handy after clicking on the function name column to search for information about that function. (By default the app uses Google. But you can change that in Settings by going to Edit -> [Settings](#).)
- **Copy 'Column Name'** – will change depending on which column you clicked. It will copy whatever is displayed in that column onto Windows Clipboard.
- **Open 'module' for Search** – will change depending on the column you clicked. It will open another instance of WinAPI Search app with that particular module selected for search.
- **Begin Searching 'module'** – will change depending on the item you clicked. It will open another instance of WinAPI Search app and begin searching that module automatically.

In case the main Results list becomes too unmanageable, right-click it and select **"Preview Search Results"**, or just hit Ctrl+Shift+P on the keyboard. It will open the search results list in a read-only HTML document in your web browser for easier reading.

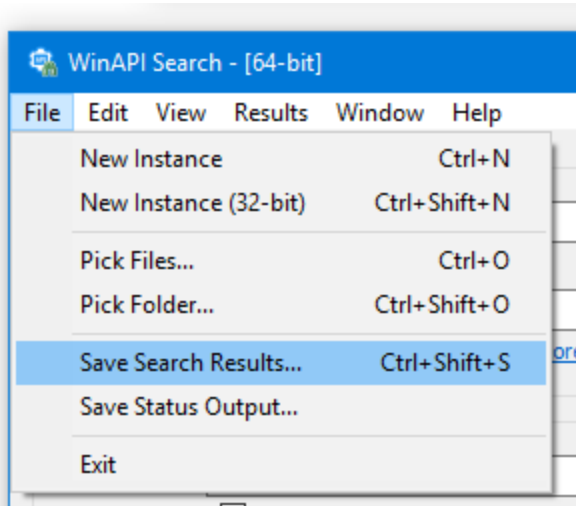
Lastly, the **"Show Info"** option will display a new popup window with a full report about a specific function that was selected in the Results list. You can also achieve the same by double-clicking any Results list item. (Also note that you can open more than one "Show Info" window, side by side.) Such window may look like this:



Note that in this popup window you can also click both “...” buttons for additional context menu options:

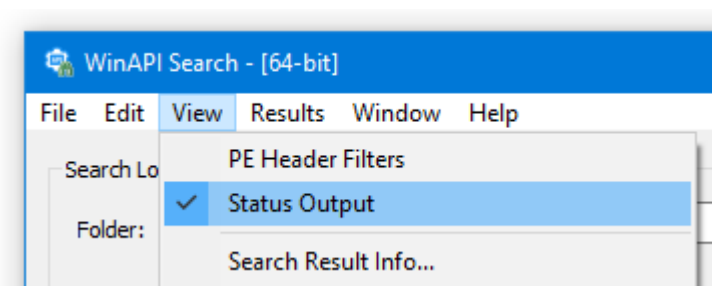


If you want to save the search results in a file, go back to the main WinAPI Search window, and select File -> Save Search Results:

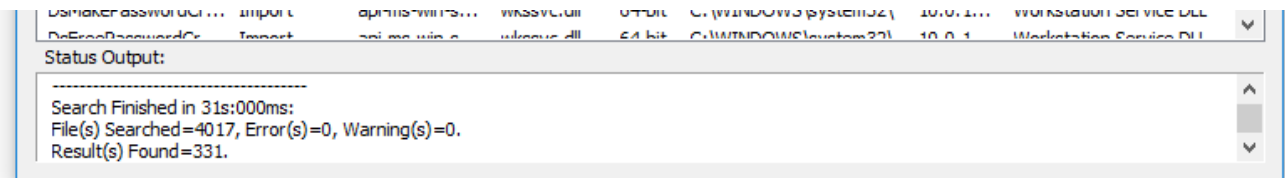


You will have an option to save it as HTML file, or as a plain text file.

Additionally, if you want to monitor the diagnostic output during the search, you will need to display the Status Output window. You can do so from the main menu by going to View -> Status Output:

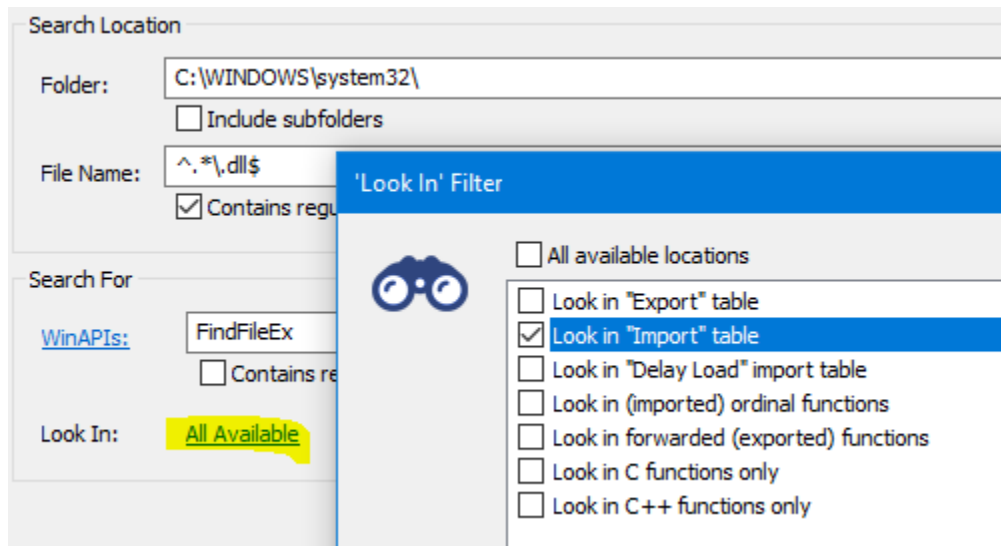


This will display it at the bottom of the main window:

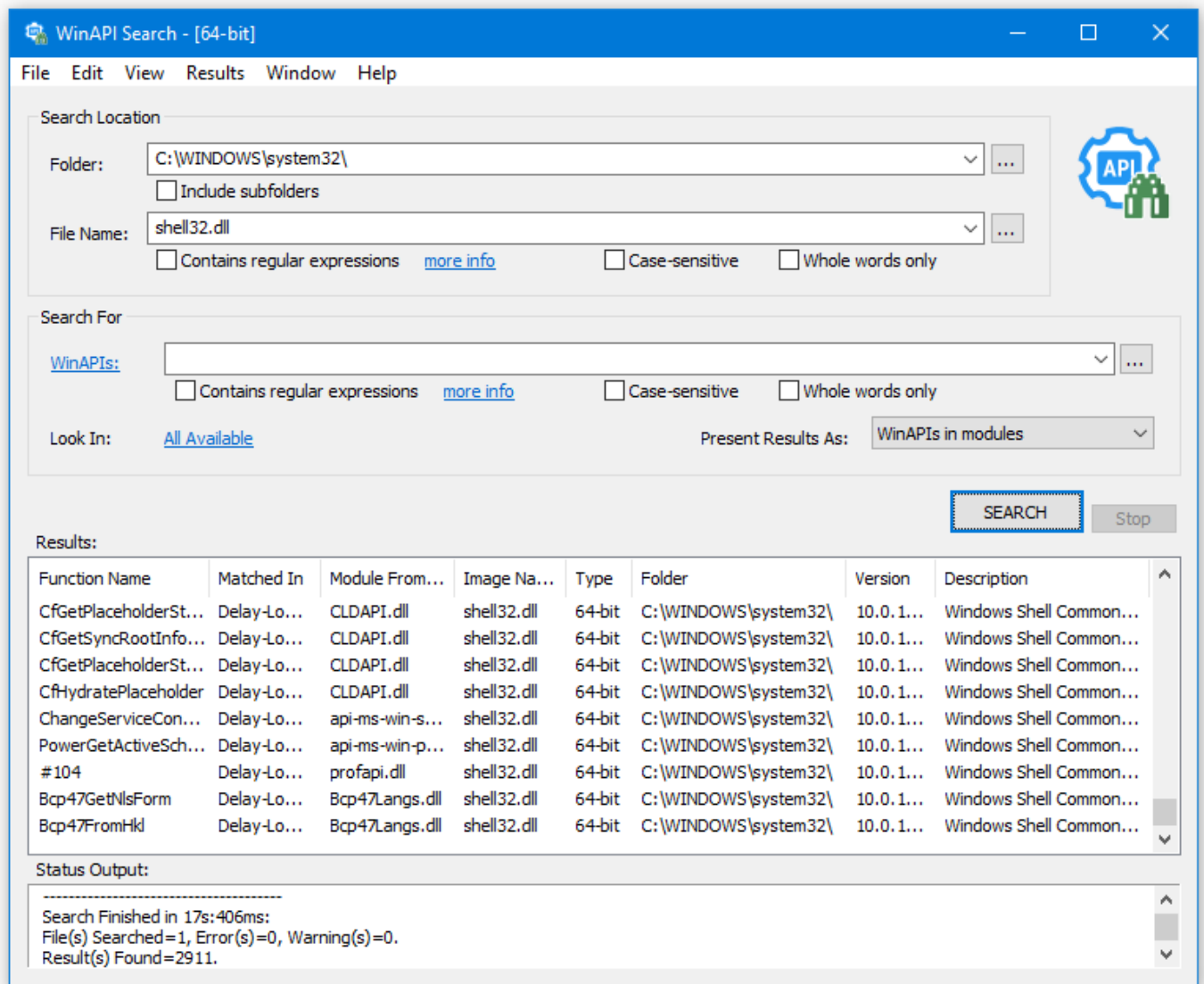


- **“Look In” Filter**

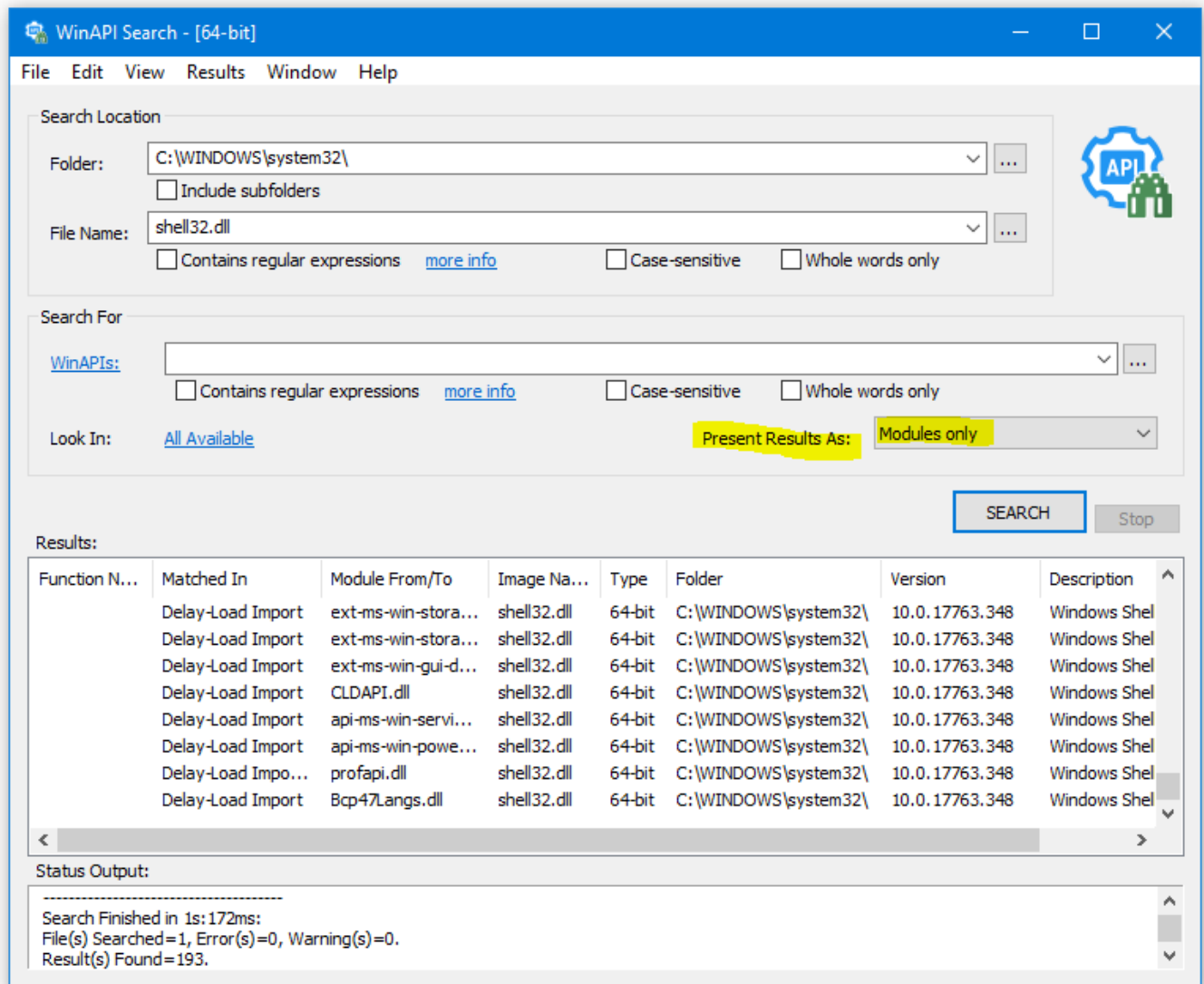
You can also use the WinAPI Search app to search for specific function(s), used in a certain way. Say, if you want to know which of the system libraries imports FindFileEx function, you would select it first (as was described above), but before hitting Search, you would also click on the “Look In” link and specify “Look in import table” option:



- Or, you can use WinAPI Search app as a replacement for **Dependency Walker**. Say, if you want to get all APIs from, say, shell32.dll, you would select that DLL first and then leave all other filters blank. It will produce the results pretty similar to what Dependency Walker may do, but with many more additional options:



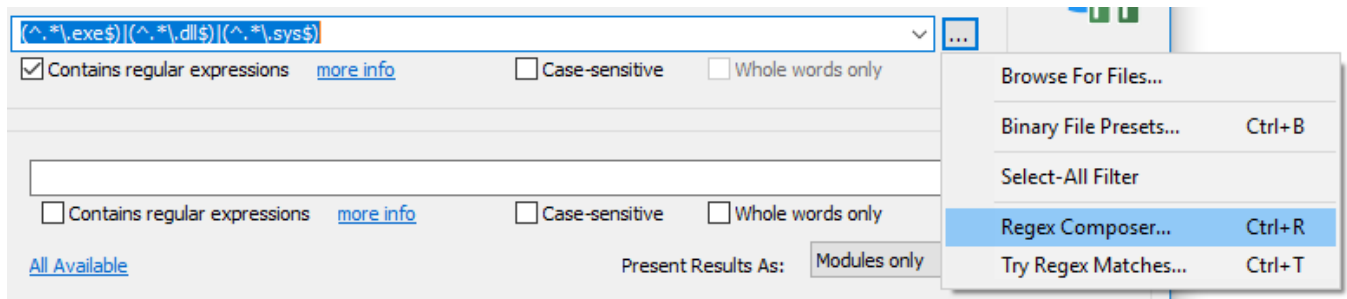
Or, if you don't care about any specific APIs and just want modules in shell32.dll, clear all filters, and then in “**Present Results As**” select “Modules Only”. It will produce a shorter list of modules imported by shell32.dll:



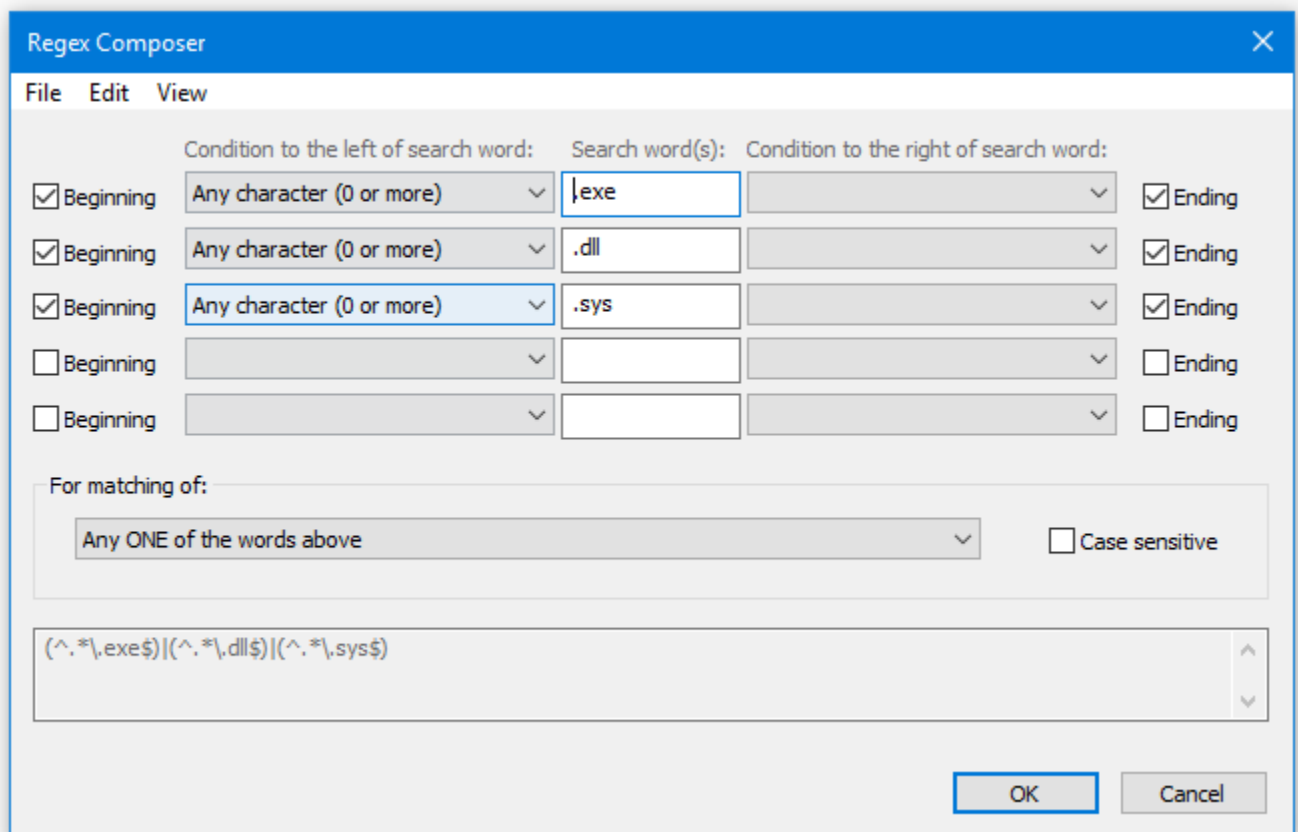
- **Regex Filters**

Most of the search filters in the app support Regular Expressions (or Regex.) The explanation of the Regex syntax is beyond the scope of this short manual. I'll just say that the WinAPI Search app supports full ECMAScript Regex syntax. In case you're new to Regex, most edit boxes have “more info” links below them that will open a popup window with a short tutorial about supported Regex.

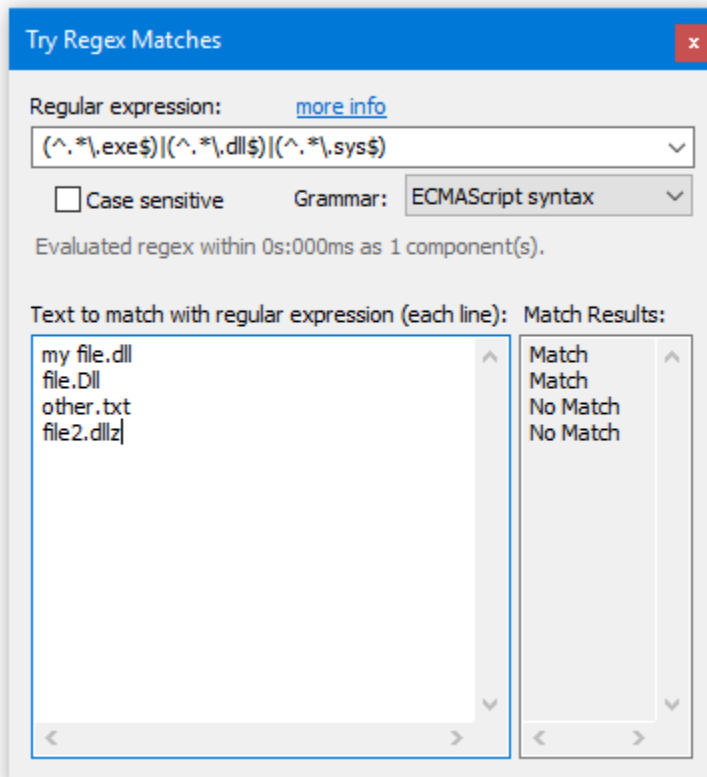
I personally don't like messing with Regex too much, and instead prefer using the “**Regex Composer**” that is available from many “...” button context menus. You can also invoke it by hitting Ctrl+R on the keyboard:



It will show this nifty window that can allow you to specify the Regex in a *visual* way, without worrying about the correct Regex syntax and proper escaping:



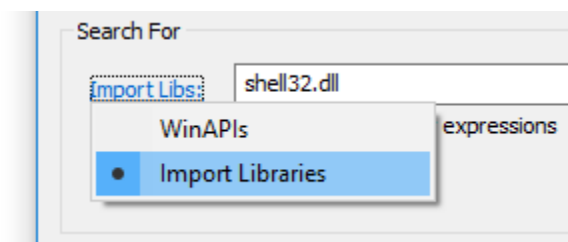
And, one more thing about Regex. In case you're not sure if your particular Regex has any effect or not, you can try it out using the "**Try Regex Matches**" option, that is also available from many "..." button context menus, or from hitting Ctrl+T on the keyboard. It will show a popup window that will help you to try out which text your Regex can match with. Here's an example:



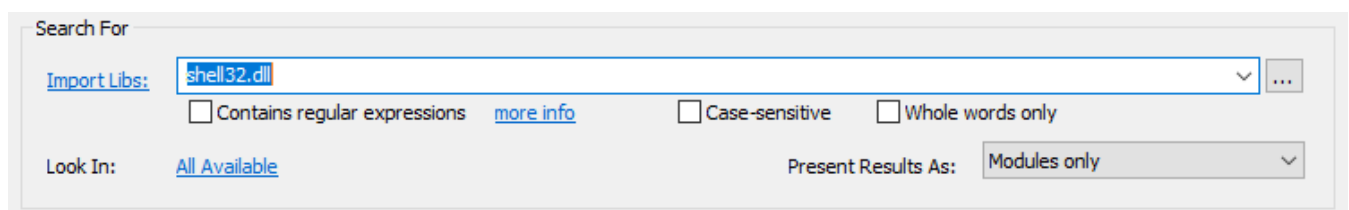
By typing search patterns into the bottom window on the left you can see if the Regex on top produces a match in the column on the bottom right. The matches are done by each line, in real time as you type.

- **“Import Libs” Filter**

You don’t have to necessarily search by an API (or function) name. You can search by the name of an imported library too. For instance, if you want to find all system modules that import shell32.dll, first select “Import Libs” in the “Search For” section:



Then specify “shell32.dll” as a search word, and click Search:



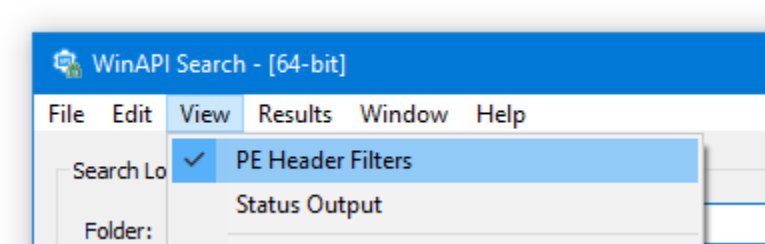
The resulting list will contain module names that import shell32.dll library:

Results:							
Function N...	Matched In	Module From/To	Image Name	Type	Folder	Version	Description
	Imported Library	SHELL32.dll	wudriver.dll	64-bit	C:\WINDO...	10.0.17763.1	Windows ...
	Imported Library	SHELL32.dll	wusa.exe	64-bit	C:\WINDO...	10.0.17763.1	Windows ...
	Imported Library	SHELL32.dll	wwanconn.dll	64-bit	C:\WINDO...	10.0.17763.1	Wireless ...
	Imported Library	SHELL32.dll	wwanmm.dll	64-bit	C:\WINDO...	10.0.17763.1	WWan Me...
	Imported Library	SHELL32.dll	Wwanpref.dll	64-bit	C:\WINDO...	10.0.17763.1	Wireless ...
	Imported Library	SHELL32.dll	XpsGdiConverter.dll	64-bit	C:\WINDO...	10.0.17763.1	XPS to GD...
	Imported Library	SHELL32.dll	xpsrchvw.exe	64-bit	C:\WINDO...	10.0.17763.134	XPS Viewer
	Delay-Load Impo...	SHELL32.dll	xwizards.dll	64-bit	C:\WINDO...	10.0.17763.1	Extensible...
	Imported Library	SHELL32.dll	zipfldr.dll	64-bit	C:\WINDO...	10.0.17763.107	Compress...

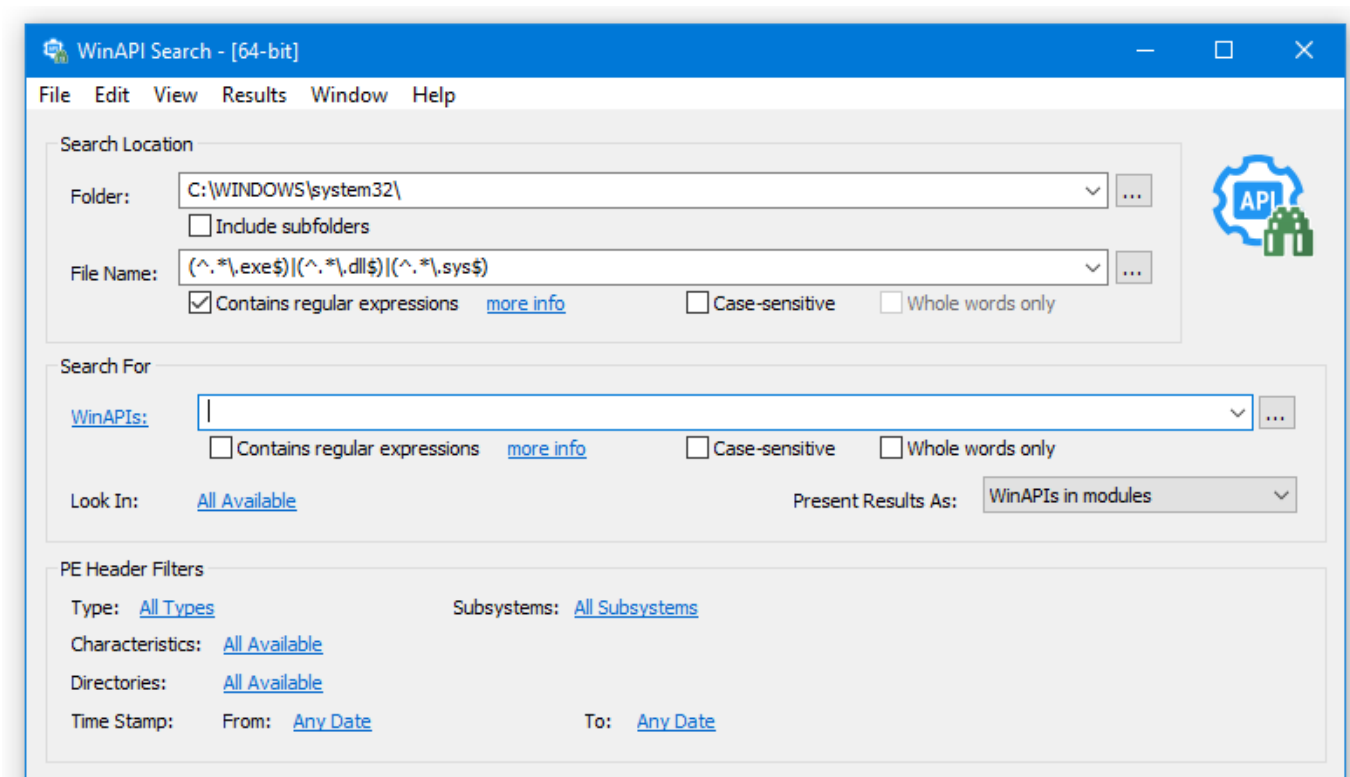
These are just some random possibilities of how you can use the WinAPI Search app.

- **PE Header Filters**

The WinAPI Search app also supports searching for binary files with specific PE header details. To do that you need to check *PE Header Filters* in the *View* menu:

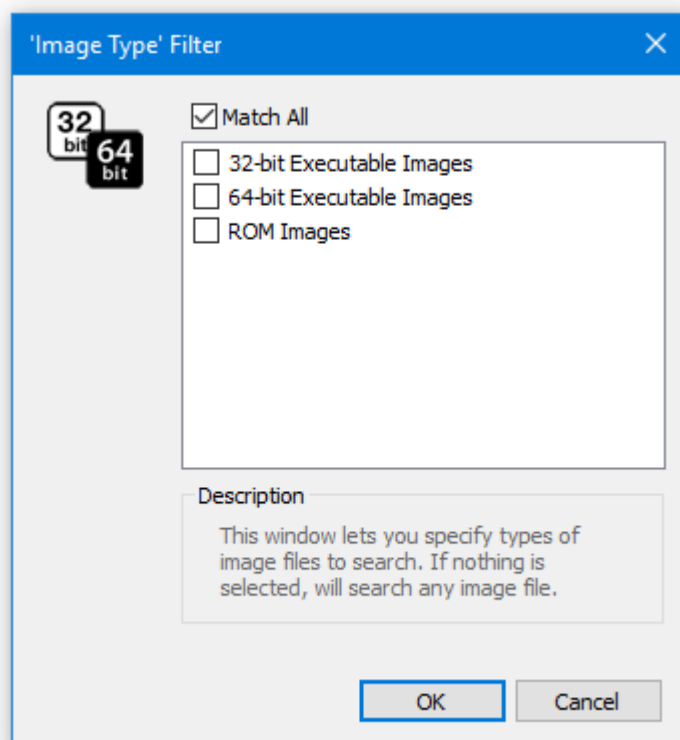


After that the main window of the app will be expanded to include additional filters at the bottom:

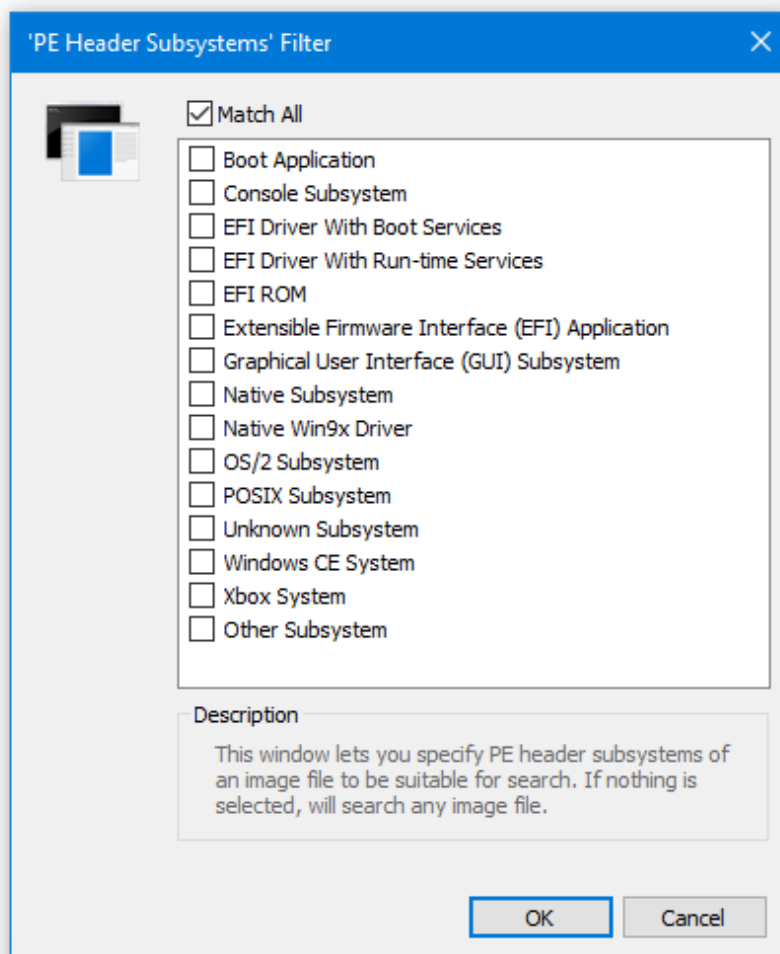


The “PE Header Filters” section provides additional filters to search by. Let me quickly explain them for you:

- **Type:** allows to search for binary files of certain bitness and type. You can select one or more options. If nothing is selected, this filter will be off (or in other words, it will accept all binary types.)

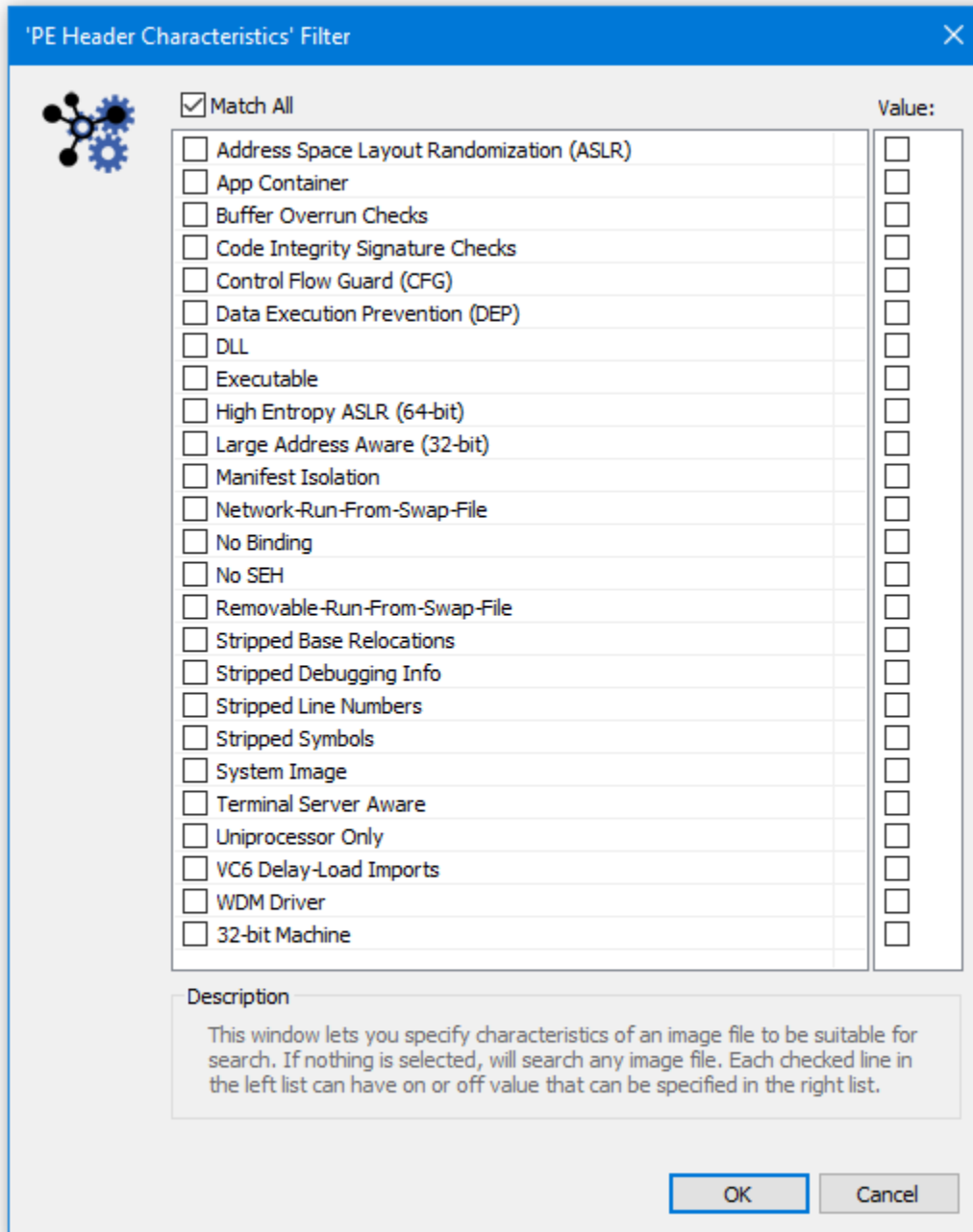


- **Subsystem:** allows you to search for binary files with specific types of PE header subsystem. Just as the filter above, you can specify one, or multiple options here. Selecting none will “match all” in this category:



To learn what each of the items in the list means, first highlight it and then refer to the short description at the bottom. Additionally you can right-click any item in the list and pick “Search Online For ...” to get more information.

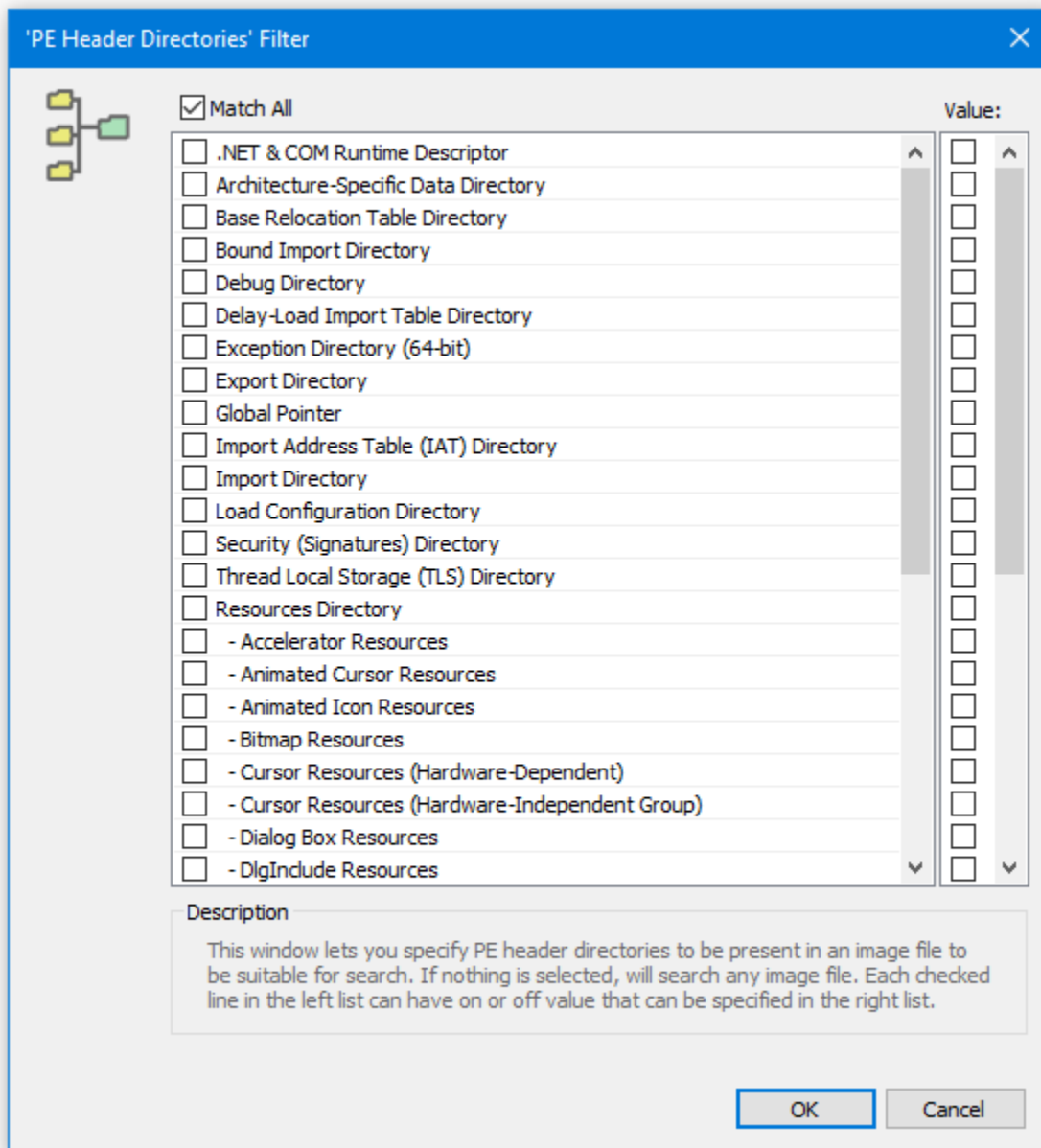
- **Characteristics:** allows you to search for binary files that support certain PE header characteristics. The list on the left gives you an option to check which characteristics you want to use in your filter. The list labeled “Value” on the right allows you to specify whether you need that characteristic to be on (when checked) or off (when unchecked.)



To learn what each of the items in the list means, first highlight it and then refer to the short description at the bottom. Additionally you can right-click any item in the list and pick “Search Online For ...” to get more information.

- **Directories:** allows you to search for binary files that contain certain PE header directories. The list on the left gives you an option to check which directories you want to use in your filter. The list labeled “Value” on the right allows you to specify whether you need that directory to be present (when checked) or absent (when unchecked.)

The “Resources Directory” filter is a special filter that allows you to specify if you want to match binary files that contain certain resources. If it’s checked by itself in the list on the left, the filter will match any resources. If any of its sub-items are checked as well, the filter will only use those specific resource types for the search.

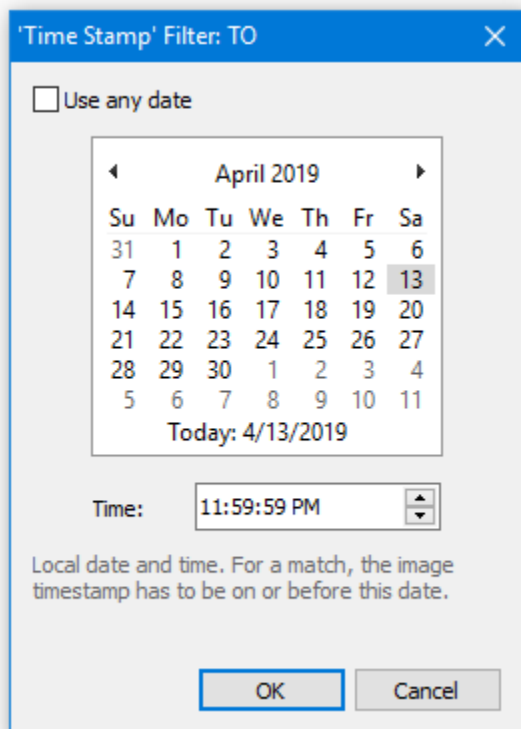


To learn what each of the items in the list means, first highlight it and then refer to the short description at the bottom. Additionally you can right-click any item in the list and pick “Search Online For ...” to get more information.

- **Time Stamp**: allows you to search for binary files with specific time-stamp of when they were compiled. (Note that executables compiled for Windows 10 [may not use time-stamps anymore](#).)

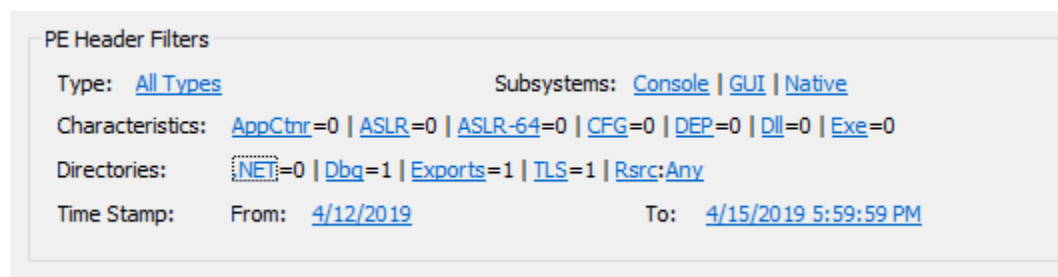
This filter is presented as a range FROM and TO. If any of the end dates of the range are not specified, this creates an open range that matches everything on that side.

The date and time for this filter must be provided for the local time zone (where the WinAPI Search app is running.)



Each of the filters above allows searching using additive conditions.

When all filters are specified, their brief annotation will be included in the main WinAPI Search window:

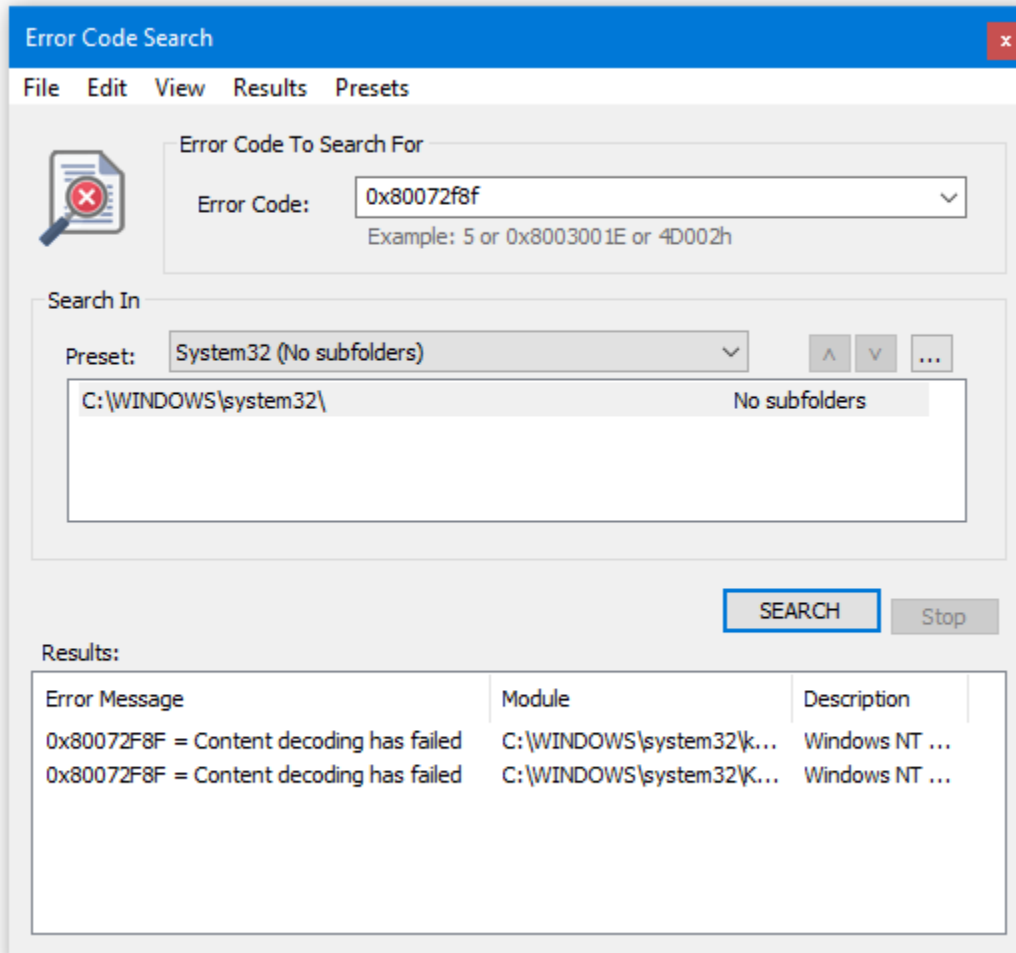


Lastly, if you want to clear out all PE filters at once, go to Edit -> "Clear All PE Header Filters", or hit Ctrl+Shift+D shortcut on the keyboard. Additionally, if you hide the PE Header Filters via the View menu, they will not be used for the search.

- **Error Code Search**

This little popup turned out to be quite handy. Its main function is to search for error codes in system (and other) binary files. How often have you run into some 0x80072f8f error and had no way of knowing what it means?

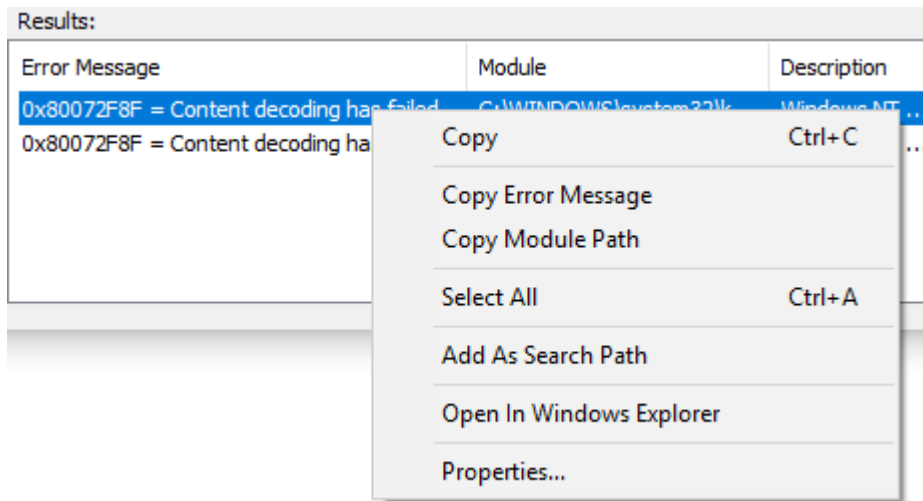
In the main WinAPI Search window, go to View -> “Error Code Search” and type in the error code you want to find, click Search and let that window find it for you. Here’s an example:



Note that you can open multiple “Error Code Search” windows.

Additionally you’re not just limited to searching the System32 folder only. You can provide additional search folders (such as Program Files folder, and others.) This popup also supports saving your search locations in presets, as well as saving your search results into a text file. You can also customize the output columns in the Results list by right-clicking on its header.

More options are also available after a right-click on an item in the Results list:



Also note that this popup is much more powerful than the “Error Lookup” tool included in the Visual Studio. Unlike that tool, this popup can search through multiple system modules that may contain your specific error code. Also note that due to that specific design, this popup may find multiple instances of the same error code in multiple binary modules.

- **Error Message Search**

This window is an opposite of the “Error Code Search”. It is less useful for general public and is more appropriate for Windows developers. It allows you to search for error codes using their textual representation. For instance, if you’re developing your Windows application or a system component while using [Win32 Error Codes](#), or HRESULT values to return errors to the callers, you might be struggling to find needed error codes for a specific context.

Here’s an example. Say, you’re coding a function that displays something on the screen and you’re working on a branch of code that receives an invalid monitor handle. What error code would you use? This popup will help you find it.

In the main WinAPI Search window, go to View -> “Error Message Search” and specify the text that your error message should have. I would suggest being less specific (because there are not that many error messages available.) My preferred way is to hit Ctrl+R to bring up the Regex Composer. Then provide what you’re looking for. In our case we need to find an error code that contains both words “monitor” and “invalid”. It may look like this:

	Condition to the left of search word:	Search word(s):	Condition to the right of search word:	
<input type="checkbox"/> Beginning		monitor		<input type="checkbox"/> Ending
<input type="checkbox"/> Beginning		invalid		<input type="checkbox"/> Ending
<input type="checkbox"/> Beginning				<input type="checkbox"/> Ending
<input type="checkbox"/> Beginning				<input type="checkbox"/> Ending
<input type="checkbox"/> Beginning				<input type="checkbox"/> Ending

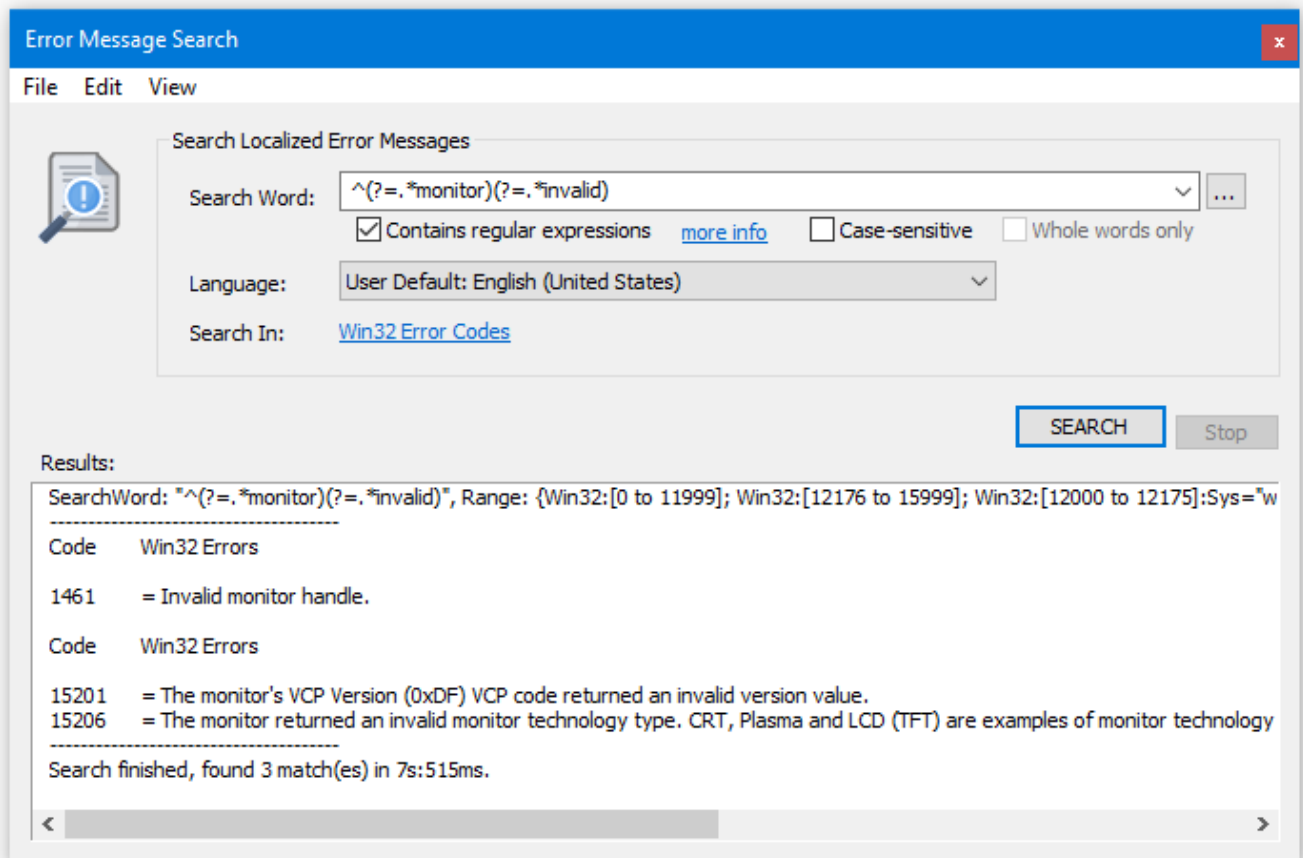
For matching of:

Combination of ALL words above, in any order ☐ Case sensitive

`^(?=.*monitor)(?=.*invalid)`

OK Cancel

This will create a nice Regex for us. Then I will specify the quickest search criteria, i.e. “Win32 Error Codes” and hit Search. In a little bit you will have your results:

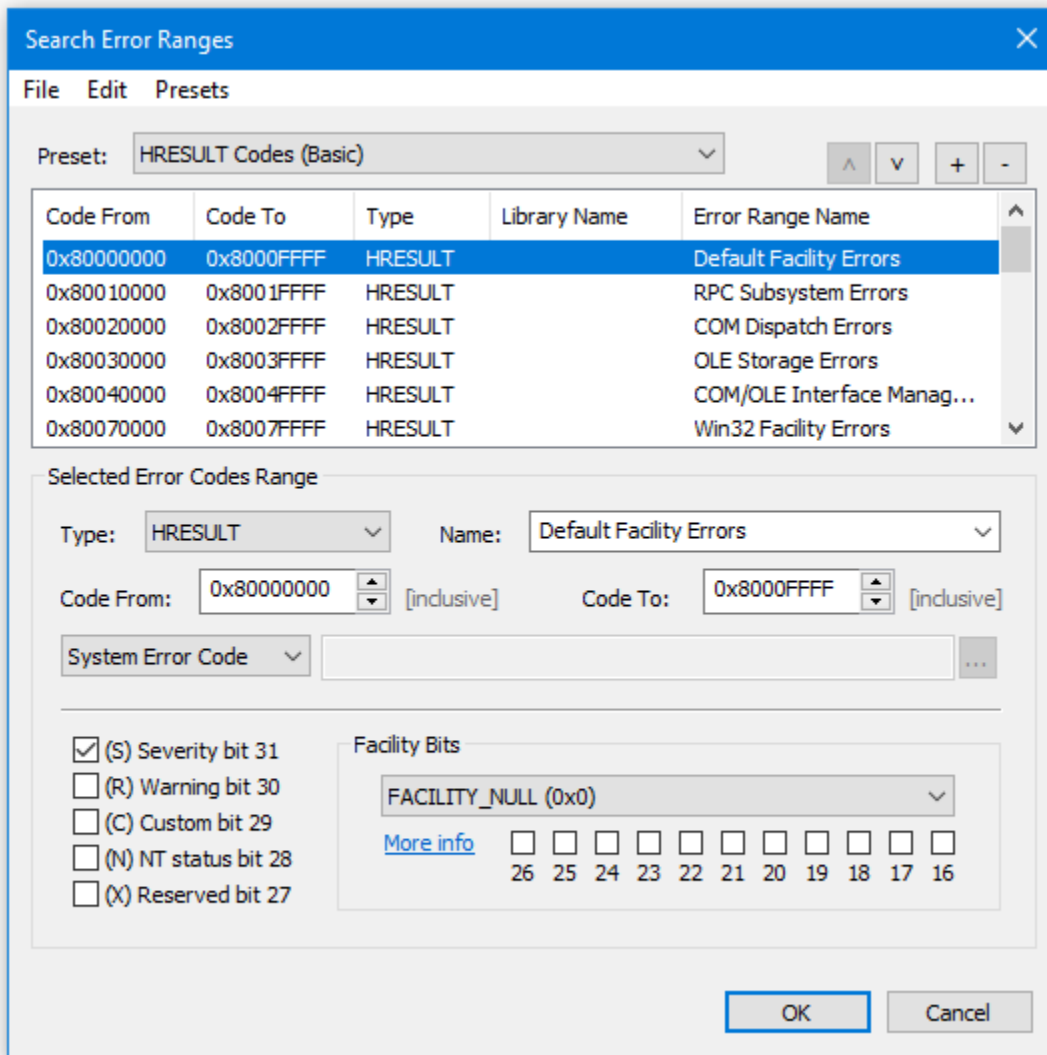


See that this window found error code 1461 that matches exactly what we need. Now you can specify it in your code by its numerical value to have a good (and localizable) error code in your function branch.

Note that this popup additionally supports searching through other (installed) languages, available through the “Languages” drop-down box. Additionally you can save the search results in the text file (via the File menu), or copy them to the Clipboard.

And just like with the “Error Code Search” popup, you can open more than one of these windows.

You can also broaden the scope of your search by clicking the “Search In” link. It provides a vast array of search locations for you to choose from, such as when you want to search the HRESULT message scope:



Using this nested popup, you can select more than just a Win32 Error Code range.

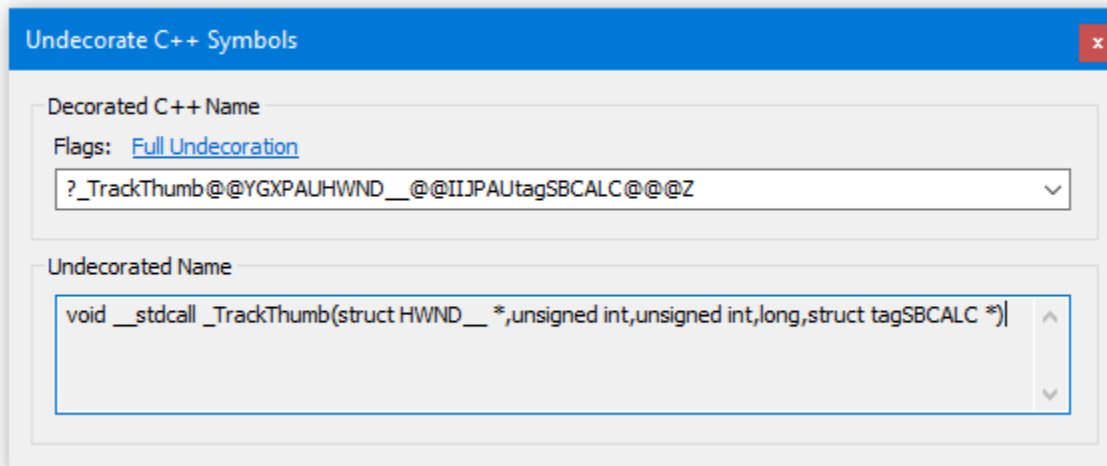
Also note that this nested popup allows you to create your own search presents, as well as to edit provided error code search ranges. I will let the reader discover the rest, as all these nuances are outside of the scope of this basic manual.

Lastly, please note though that the search done by this window is performed *live*, without caching of any results. So by specifying a very large scope you may be slowing down your search process.

- **Undecorate C++ Symbols**

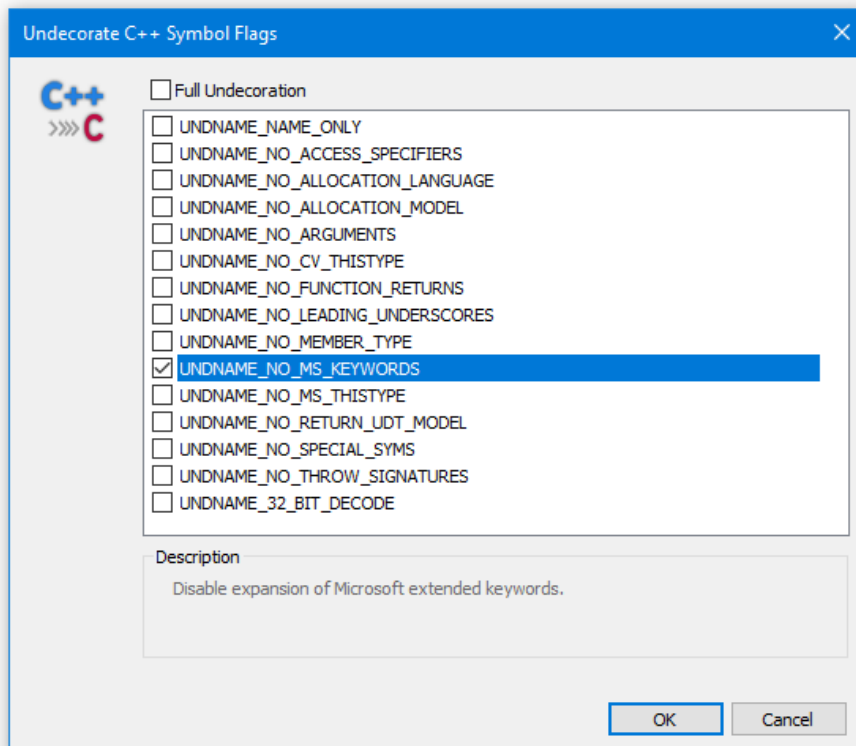
This is a very simple window (mostly for developers and researchers) that allows to *demangle* (or un-decorate, in the Microsoft lingo) previously *mangled* C++ function and variable names.

You can open it up by going to View -> “Undecorate C++ Symbols” and then paste your *mangled* symbol in the field on the top. The app will automatically *demangle* it for you in the field below:



And just like with any other popup windows in this app, you can open more than one window of this kind.

The “Flags” link allows you to specify how exactly you want your symbols to be *demangled*. The following Microsoft-specific options are available:



(This window’s default flags selection can be changed in the app’s settings, if you go to Edit -> [Settings](#).)

By right-clicking the items in the list and by selecting “Search Online For ...” you can look up more information on each available option.

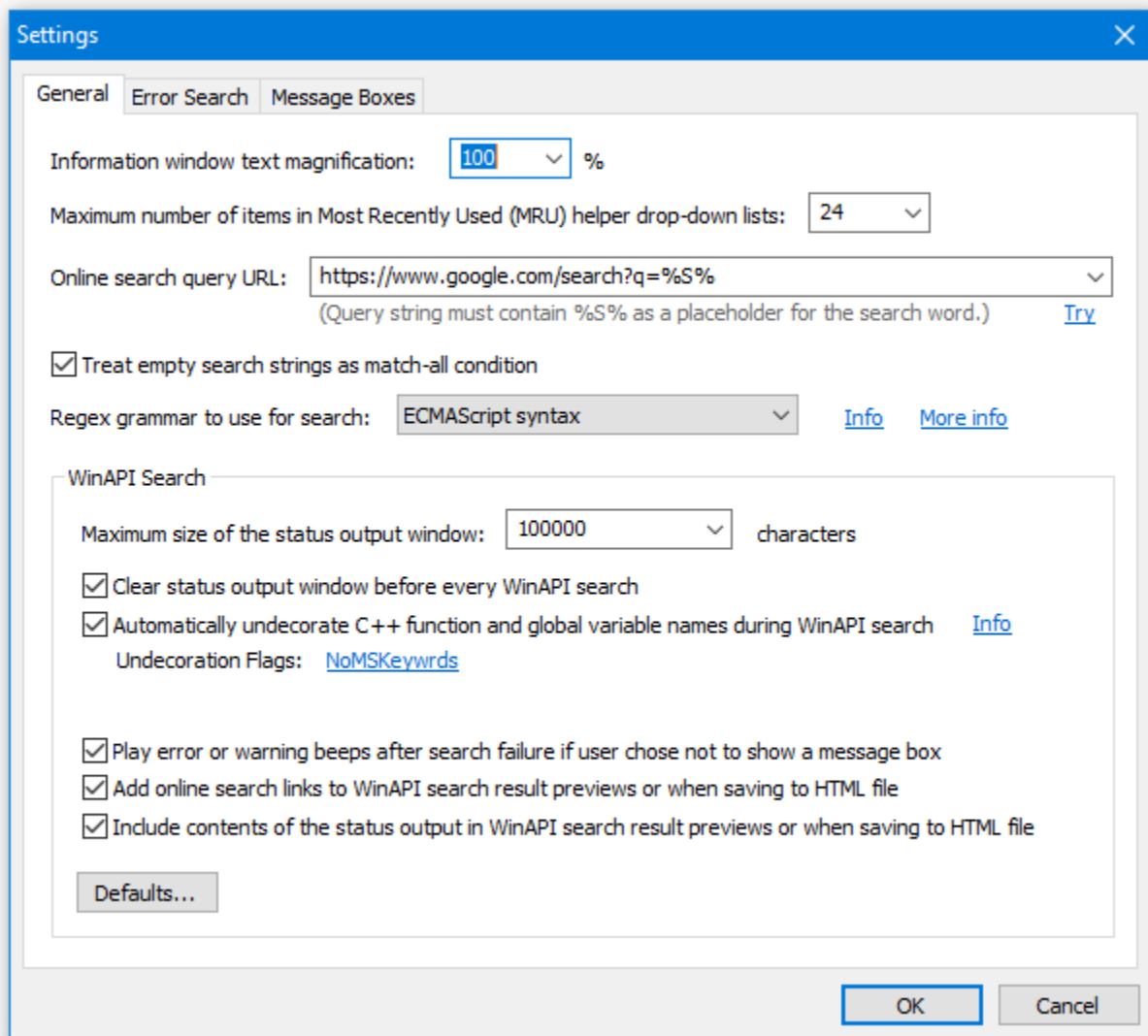
Lastly, note that this window supports the latest Microsoft-specific *mangling* scheme. It does not support any non-Microsoft symbol *mangling* options.

Settings

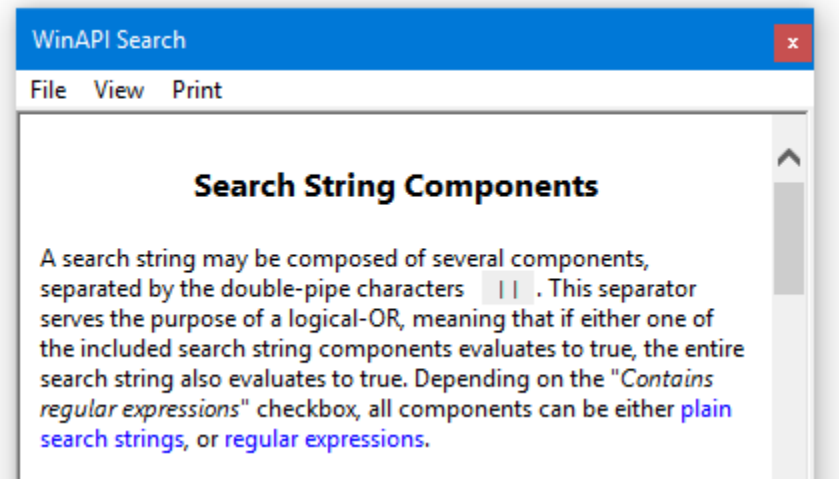
Settings for the WinAPI Search app are stored in the System Registry and are shared among all running instances of the app under the same Windows user account. The app loads its settings upon its startup and later caches them. This means that if you change any settings while another instance of the WinAPI Search app is running, the new settings won't take effect until that instance of the app is re-started.

To access settings hit Ctrl+K on the keyboard, or go to Edit -> Settings. The settings window is broken down into the following tabs, or pages:

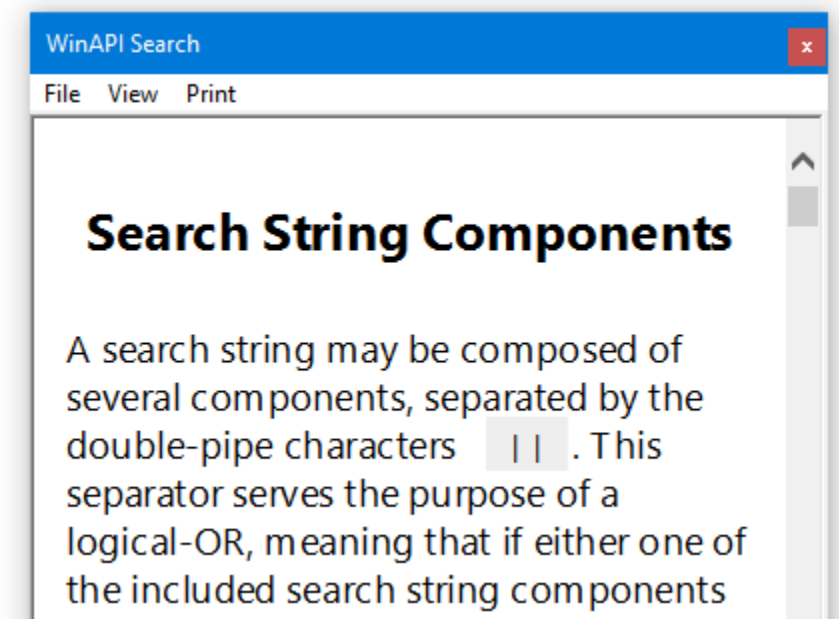
General



- **Information window text magnification** – defines how much to magnify help windows. They are displayed, for instance, when you click “more info” link to show help for the search string use. So you can make it go from:

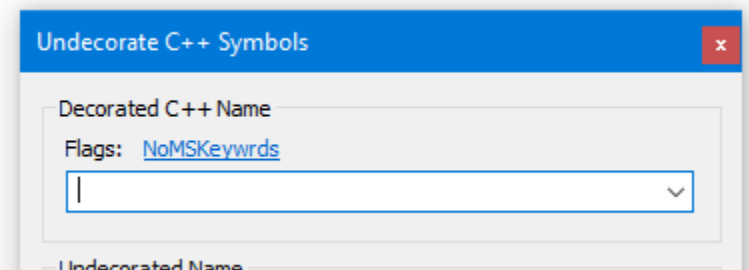


to something like this:

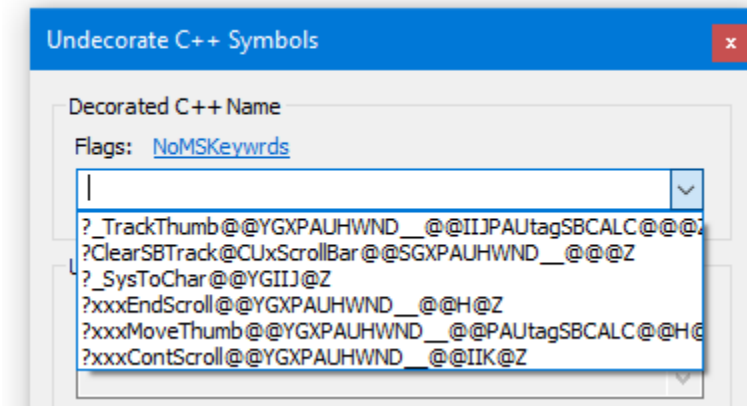


Also note that this setting affects default magnification of those helper windows. To adjust the magnification while such window is displayed on the screen, use Ctrl+Plus, Ctrl+Minus or Ctrl+0 keyboard shortcuts.

- **Maximum number of items in MRU helper drop-down lists** – defines maximum size (or depth) of the helper Most Recently Used (or MRU) drop-down lists. You can find them throughout this app's UI. For instance here:



If you click on the down-arrow on the right side of that white text field, it will show entries that were previously used in it, so you can select them from that list instead of typing them:



By the way, the order with which those items are shown is from the newest on top, to the oldest at the bottom.

In other words this setting controls how many entries should each of those helper drop-down lists remember. If the list gets too many entries, the old ones will be automatically deleted. Allowed range of entries is from 1 to 64.

- **Online search query URL** – defines the URL for a search engine to use in the context menu options that have the following name: “Search Online For ____”. This setting allows you to pick your favorite search engine. It uses Google by default, but, I know, not everyone likes the Big Guy. So if you click the drop-down arrow, it will have an option for Bing, or Duck Duck Go. Additionally you can type in your own search URL. Just make sure to keep the %S% placeholder that will be populated by the WinAPI Search app with the actual search text.

You can click “Try” link on the right to see if your edited URL actually works.

- **Treat empty search strings as match-all condition** – when checked will interpret an empty search string, as “get everything that exists” condition. If unchecked, empty search strings will not be allowed.
- **Regex grammar to use for search** – this box allows you to pick the Regex grammar to use. But I’ll be honest with you, don’t change it. The app by default is coded to use ECMAScript syntax, so if you change it, the [Regex Composer](#) will stop working properly. Change this setting only if you absolutely want to do manual Regex entry and for some reason don’t like the default ECMAScript syntax.
- **Maximum size of the status output window** – this field holds the maximum number of characters that the output window will hold before it begins deleting the old content. This window is located at the bottom of the main WinAPI Search window. (To show it, you may need to go to View -> Status Output.)



- **Clear status output window before every WinAPI search** – when checked will clear the status window (shown above) before every search. Otherwise, a new search may contain status output from a previous search.

- **Automatically undecorate C++ function and global variable names during WinAPI search** – when checked, will instruct the app to automatically *demangle* (or [un-decorate](#) in Microsoft lingo) all *mangled* symbols in function names, that were found during the search. So your results may look like this:

Results:		
Function Name	Matched In	Module From/To
public: class DirectUI::Element * DirectUI::Element::GetTopLevel(void)	Delay-Load Imp...	DUI70.dll
public: static class DirectUI::Value * DirectUI::Value::CreateGraphic(st...	Delay-Load Imp...	DUI70.dll
public: long DirectUI::DUIXmlParser::SetXMLFromResourceWithTheme(...	Delay-Load Imp...	DUI70.dll
public: long DirectUI::DUIXmlParser::SetXMLFromResource(unsigned s...	Delay-Load Imp...	DUI70.dll
public: long DirectUI::DUIXmlParser::SetPreprocessedXML(unsigned sh...	Delay-Load Imp...	DUI70.dll
public: long DirectUI::DUIXmlParser::SetXML(unsigned short const *,st...	Delay-Load Imp...	DUI70.dll
public: long DirectUI::DUIXmlParser::GetSheet(unsigned short const *,...	Delay-Load Imp...	DUI70.dll

(Note that *demangling* happens only after the search has finished and may take some time.)

If this setting is off, the list will contain original *mangled* symbols:

Results:		
Function Name	Matched In	Module From/To
?GetTopLevel@Element@DirectUI@@QEAAPEAV12@XZ	Delay-Load Imp...	DUI70.dll
?CreateGraphic@Value@DirectUI@@SAPEAV12@PEAUHICON__@@_...	Delay-Load Imp...	DUI70.dll
?SetXMLFromResourceWithTheme@DUIXmlParser@DirectUI@@QEAAJ...	Delay-Load Imp...	DUI70.dll
?SetXMLFromResource@DUIXmlParser@DirectUI@@QEAAJPEBGPEAU...	Delay-Load Imp...	DUI70.dll
?SetPreprocessedXML@DUIXmlParser@DirectUI@@QEAAJPEBGPEAUH...	Delay-Load Imp...	DUI70.dll
?SetXML@DUIXmlParser@DirectUI@@QEAAJPEBGPEAUHINSTANCE__...	Delay-Load Imp...	DUI70.dll
?GetSheet@DUIXmlParser@DirectUI@@QEAAJPEBGPEAPEAVValue@2	Delay-Load Imp...	DUI70.dll

In this case, to *demangle* a function name right-click its list entry and go to “Show Info”.

- **Undecoration Flags** – when you click the link on the right, it will show a popup window that allows you to specify Microsoft-specific symbol *demangling* options used by default in the “[Undecorate C++ Symbols](#)” window.
- **Play error or warning beeps after search failure if user chose not to show a message box** – this setting, when checked, will instruct the app to play a system error beep or a warning if the user chose not to display a message box (in the “[Message Boxes](#)” page in settings.) This is just a fallback option to warn you of an error that took place during the search. In this case a prudent thing to do is to check the status output window for details. On the other hand, if those error beeps are bothering you, uncheck this option in settings.
- **Add online search links to WinAPI search results previews or when saving to HTML file** – this setting affects the Results -> “Preview Search Results” option, as well as the option for saving the WinAPI search results in HTML file. When checked, it will add hyperlinks to search online for function and module names, as such:

Function Name	Matched In	Module From/To	Image Name	Type	Folder	Version	Description
expf	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
floor	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
floorf	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
log	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
ceilf	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
memcmp	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll

When this setting is not checked, those links will not be added:

Function Name	Matched In	Module From/To	Image Name	Type	Folder	Version	Description
expf	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
floor	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
floorf	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
log	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
ceilf	Import	msvcrt.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll

- **Include contents of the status output in WinAPI search result previews or when saving to HTML file** – just like the setting above, it affects the option in Results -> “Preview Search Results” or when saving the WinAPI search results in HTML file. When checked, it will add the contents of the status output window into the report. It will be placed at the bottom. Here’s an example:

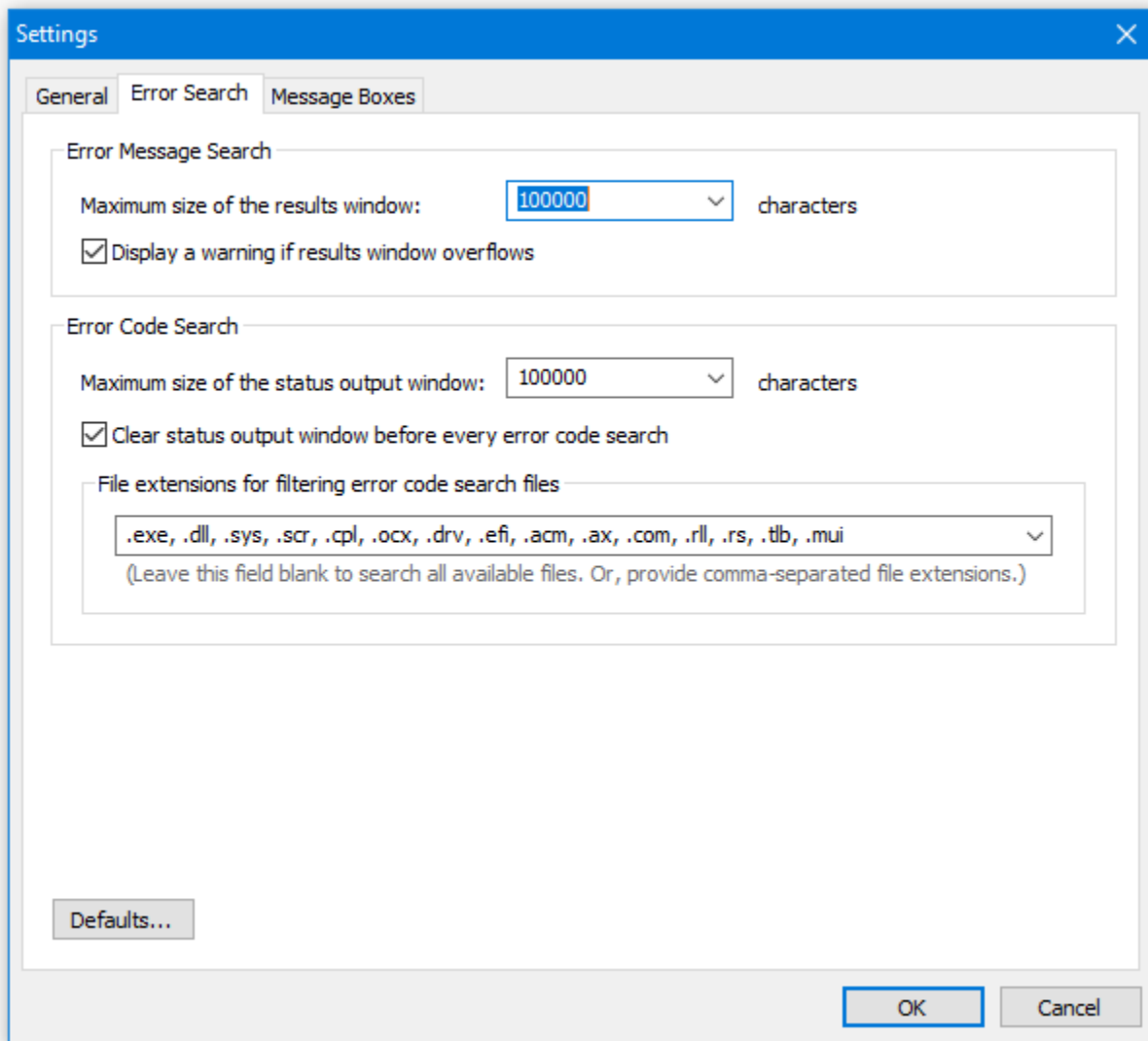
RegSetValueExW	Import	api-ms-win-core-registry-l1-1-0.dll	shell32.dll	64-bit	C:\Windows\System32\	10.0.17134.228	Windows Shell Common Dll
--------------------------------	--------	---	-----------------------------	--------	----------------------	----------------	--------------------------

Status Output:

```
Initiating Search:
SearchFolder="C:\Windows\System32\" (No subfolders), SearchFiles={"shell32.dll",
Regex=No, CaseSensitive=No, WholeWords=No}, SearchFor="WinAPIs", SearchString={"",
Regex=No, CaseSensitive=No, WholeWords=No}, LookIn="C", PE Filter:{Type="All",
Subsystems="All", Characteristics="All", Directories="All", TimeFrom="All",
TimeTo="All"}}, PresentAs="APIs-in-Modules"
-----
Search was aborted...
```

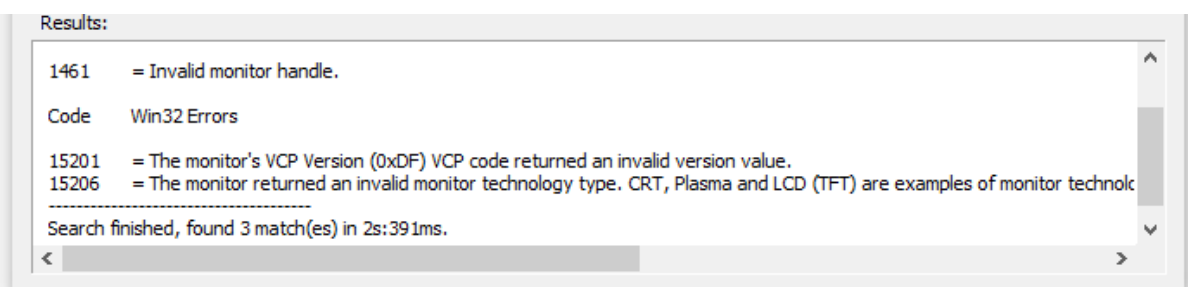
- **Defaults** – will reset everything on the page to defaults.

Error Search

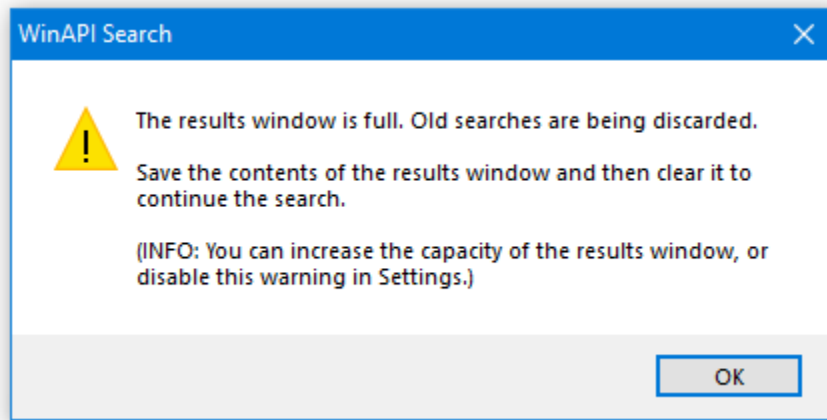


- **Error Message Search**

- **Maximum size of the results window** – defines the maximum number of characters that the results window of the [Error Message Search](#) popup will hold before it begins deleting the old content.



- **Display a warning if results window overflows** – when checked will display a warning message box if the results window (shown above) goes over its character limit. It may look like this:

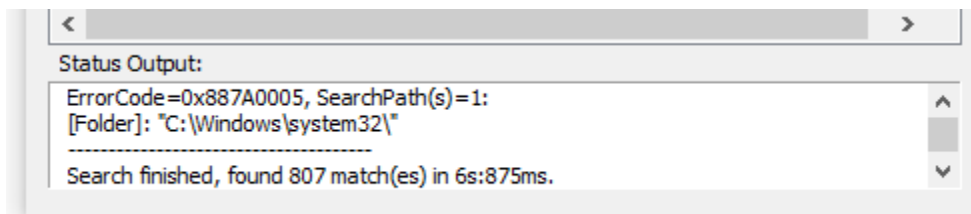


When this warning is displayed, the search will be paused.

If you disable this setting, the results window will silently erase all old content that doesn't fit the limit that is specified by the setting that was described above.

- **Error Code Search**

- **Maximum size of the status output window** – defines the maximum number of characters that the status output window of the [Error Code Search](#) popup will hold before it begins deleting the old content. (That window is located at the bottom. To display it you may need to go to View -> Status Output.)

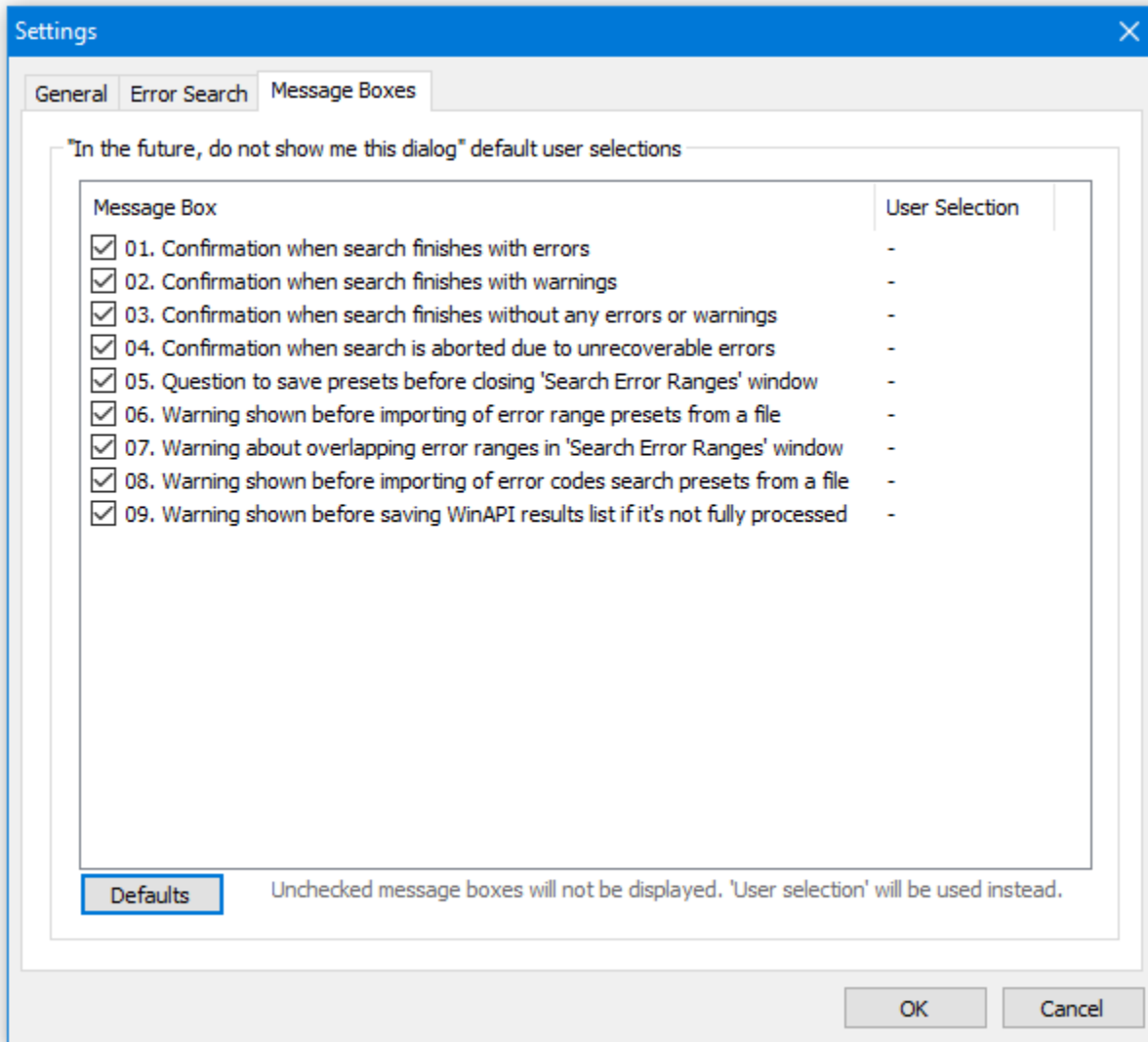


- **Clear status output window before every error code search** – when checked will clear the status window (shown above) before every search. Otherwise, a new search may contain status output from a previous search.
- **File extensions for filtering error code search files** – contains a list of (binary) file extensions to scan when searching for error codes. The drop-down list contains the most commonly used file extensions, but you're not limited to those file extensions only. You can type in your own, or leave this field blank to search all files that are present in the search folders. If you specify your own file extensions, make sure to begin them with a dot, followed by the text name of the extension, as in: `.exe`. Separate multiple extensions with commas.

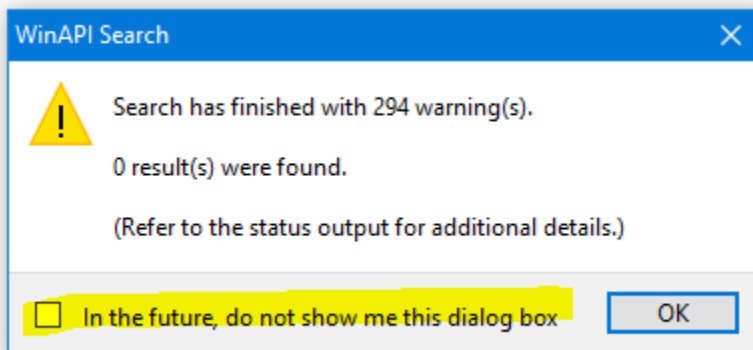
The purpose of this setting is to speed up the error code search, as there's generally no need to open non-binary files since those do not contain error code definitions.

- **Defaults** – will reset everything on the page to defaults.

Message Boxes



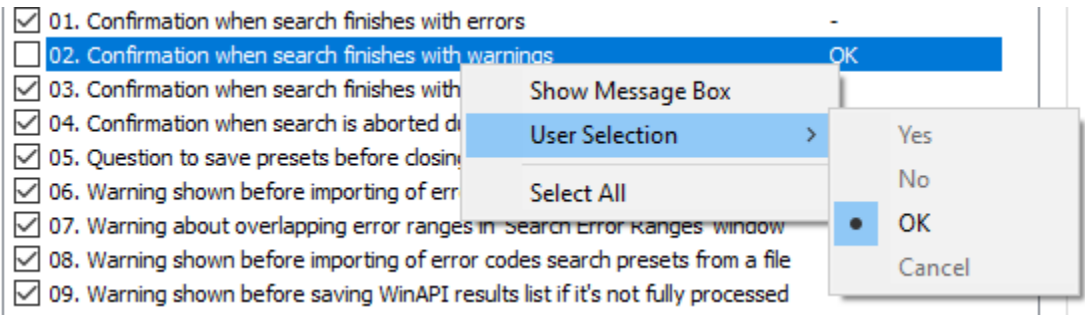
- **Checked list** – this list contains names of message boxes that have a checkbox that, in English, is named: *"In the future, do not show me this dialog box"*. It may be displayed throughout operation of the WinAPI Search app:



If the user checks that box (marked with yellow in the screenshot above), that popup message will not be shown again. This window in settings allows reverting such selection or adjusting it even further.

The checkboxes in the left side of the list control whether or not those specific message boxes will be displayed to the user. (Checked = displayed.) While "User Selection" column contains default message box button selection that

will be selected by default if the message box is not shown. If it's a message box with more than just an OK button choice, you can select it in that column. To do that, double-click that column to cycle through all available options, or right-click it to see the context menu with options:



- **Defaults** – will reset everything on the page to defaults.

Bug Reports

As we all know, bugs are an inevitable part of any software. And this one is not an exception. So if you encounter one, please follow these steps to report it:

1. Make sure to reproduce the bug.
2. Then, right after that, while WinAPI Search app is still running, go to Help -> Bug Report -> "Export Event Log" and provide some location on your drive. Give it some name and click Save to generate it.
3. The app will generate a bug report. You can open it in your web browser to make sure that it doesn't contain any of your personal information. (We DO NOT need any of that!)
4. Then go back to Help -> Bug Report -> Submit Bug Report. This will open our website. Please write a short message in English to explain what the bug is. You will not be able to attach your exported report at this point.
5. When we reply to your message, we will provide you with an email address to send in the report that you exported earlier. So for now, please hang on to it.

That is it.

Thank you for reporting software bugs to us!

Contact & Downloads

As was [described above](#), you can contact us to submit your bug reports. For any other inquiries you can reach us at the following URL:

<https://dennisbabkin.com/contact>

Note that although we encourage you to send us suggestions for improvement, we **DO NOT** provide tech support for our free software. Please respect our time and do your own research before contacting us.

Additionally, if you want to download your free copy of this app, or want to include it in your magazine (periodical), please use the following link:

<https://dennisbabkin.com/utilities/#WinApiSearch>

Note that we **DO NOT** advise downloading this app from ANY other location as it may contain adware.

Thank you for using WinAPI Search app!