



DLL Export Viewer v1.66

Copyright (c) 2005 - 2016 Nir Sofer

Web Site: <http://www.nirsoft.net>

Description

This utility displays the list of all exported functions and their virtual memory addresses for the specified DLL files. You can easily copy the memory address of the desired function, paste it into your debugger, and set a breakpoint for this memory address. When this function is called, the debugger will stop in the beginning of this function.

For example: If you want to break each time that a message box is going to be displayed, simply put breakpoints on the memory addresses of message-box functions: `MessageBoxA`, `MessageBoxExA`, and `MessageBoxIndirectA` (or `MessageBoxW`, `MessageBoxExW`, and `MessageBoxIndirectW` in unicode based applications) When one of the message-box functions is called, your debugger should break in the entry point of that function, and then you can look at call stack and go backward into the code that initiated this API call.

Versions History

- Version 1.66
 - Fixed to display the dll filename in the window title when dragging a file from Windows Explorer.
- Version 1.65
 - The dll filename is now displayed in the window title.
- Version 1.63
 - Added `/cfg` command-line option, which instructs DLL Export Viewer to use a config file in another location instead of the default config file, for example:
`dllexp.exe /cfg "%AppData%\dllexp.cfg"`
- Version 1.62
 - Fixed the flickering while browsing the exported functions list.
- Version 1.61
 - Fixed issue: The properties and the other windows opened in the wrong monitor, on multi-monitors system.

- Version 1.60
 - Added 'Put Icon On Tray' option.
- Version 1.55
 - Added 'Explorer Context Menu' option. If it's turned on - 'Open With DLL Export Viewer' menu item is added to the context menu of Windows Explorer when you right click on a dll file.
- Version 1.50
 - When API export entry is forwarded function that points to another function in another dll (like in wsock32.dll and some functions in kernel32.dll and kernelbase.dll), DLL Export Viewer now displays the forwarder string that specifies external function name, instead of the memory address.
- Version 1.45
 - Unnamed Functions are now displayed in Ordinalxxx format. (When 'Display Unnamed Functions' option is turned on)
- Version 1.42
 - Added 'Add Header Line To CSV/Tab-Delimited File' option. When this option is turned on, the column names are added as the first line when you export to csv or tab-delimited file.
 - You can now send the exported functions list to stdout by specifying an empty filename ("") in the command-line of all save parameters. For example: dllexport.exe /from_files "c:\windows\system32\shell32.dll" /stab "" >> c:\temp\funclist.txt
- Version 1.41
 - Fixed focus and tab order issues in the Options window.
- Version 1.40
 - Added more command-line options to control the dll scan settings.
- Version 1.36
 - Added /sort command-line option.
- Version 1.35
 - Added option to undecorate/unmangle C++ function names that mangled by Microsoft compilers. When this option is selected, all mangled functions will be displayed with parameters list and return value type. Works on computers with Windows XP or greater OS.
- Version 1.30
 - Added Drag & Drop support.
 - Added support for 'Open With' menu/dialog-box of Windows. If you choose Dll Export Viewer in the 'Open With' dialog-box, it'll now be opened with the right dll that you right-clicked.

- Fixed the Esc key problem in the 'Select Dlls' dialog-box.
- Version 1.27
 - New option: Show found items during the scan process.
- Version 1.26
 - Added option: Automatically remove Ascii/Unicode Postfix ('A' and 'W') On Google Search. It means that if the function name, for example, is CreateWindowExW, the google search will be 'CreateWindowEx'.
- Version 1.25
 - Added function name filter - Don't show functions that contains the specified strings.
- Version 1.22
 - The export options dialog-box is now resizable.
- Version 1.21
 - Added AutoComplete support for filename text-boxes.
- Version 1.20
 - Added new option: Scan subfolders.
 - Fixed bug: The main window lost the focus when the user switched to another application and then returned back to DLL Export Viewer.
- Version 1.16
 - Added support for saving as comma-delimited text file.
- Version 1.15
 - Added command-line support
- Version 1.12
 - The configuration is now saved to a file, instead of the Registry.
- Version 1.11
 - Added function name filter (Show only functions that contains the specified strings)
- Version 1.10
 - Added support for 64-bit DLL files.
 - New option: Display method and properties of COM type libraries.
 - Function counter is now displayed while scanning the files.
 - A tooltip is displayed when a string in a column is longer than the column length.
- Version 1.02 - New Option: Search Function In Google.
- Version 1.01 - New Option: Copy WinDbg BP Commands. You can use this option to easily set multiple breakpoints in [WinDbg debugger](#). Just select the desired API functions, execute this option, and then paste the bp commands into the command window of WinDbg.

- Version 1.00 - First Release.

Start using DLL Export Viewer

This utility doesn't require any installation process or additional DLLs, in order to start using it, just run the executable file - dllexp.exe

When DLL Export Viewer is loaded, you have to choose one of the following options:

- **Load all functions from standard system DLLs:** This is the default option. If you select it, the exported API functions of standard Windows DLLs (kernel32.dll, user32.dll, and so on...) will be displayed.
- **Load functions from the specified DLL file:** If you select this option, you have to specify the DLL file that you want to load in the text-box below this option. You can also specify a wildcard for loading multiple DLL files. If for some reason, you want to view all API functions on your system, you can specify something like 'c:\windows\system32*.dll' - but I must warn you... You'll get a very long functions list, probably more than 50,000 functions !
- **Load functions from the DLL files specified in the following text file:** If you select this option, the specified text file should contain a list of DLL files, separated by Enter characters (CR-LF). All exported functions from the specified DLLs will be loaded.
- **Load functions from all DLLs attached to the selected process:** This is the most useful option if you want to use this utility for debugging. Select the process that you are currently debugging, and the exported functions of all DLLs attached to the selected process will be displayed.

System Requirements

This utility works properly on all 32-bit/64-bit versions of Windows, including Windows 98/ME, Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows 7, Windows 8, and Windows 10. If you want to use it on Windows NT, you should add psapi.dll into your system32 folder.

Command-Line Options

/cfg <Filename>	Start DLL Export Viewer with the specified configuration file. For example: dllexport.exe /cfg "c:\config\dllexport.cfg" dllexport.exe /cfg "%AppData%\dllexport.cfg"
/from_std	Load all functions from standard system DLLs.
/from_files <Filename>	Load all functions from the specified DLL file or wildcard.
/from_textfile <Filename>	Load all functions from the DLL files specified in the text file.
/from_process <Process ID>	Load all functions from the DLL files loaded by the specified process.
/filter <Filter String>	Show only functions that contain the specified strings
/nofilter	Don't use the functions filter.
/subfolders	Scan subfolders (Only for /from_files)
/nosubfolders	Don't scan subfolders (Only for /from_files)
/DisplayUnnamed <0 1>	Display Unnamed Functions. 0 = No, 1 = Yes
/ScanComTypeLibs <0 1>	Scan COM type libraries. 0 = No, 1 = Yes
/ScanExports <0 1>	Scan exported functions. 0 = No, 1 = Yes
/ShowComMethods <0 1>	Show COM methods. 0 = No, 1 = Yes
/ShowComProperties <0 1>	Show COM properties. 0 = No, 1 = Yes
/UseIncludeOnlyStr <0 1>	Show only functions that contain the following strings. 0 = No, 1 = Yes
/IncludeOnlyStr <String>	Specifies the include string when UseIncludeOnlyStr = 1
/UseExcludeStr <0 1>	Don't show functions that contain the following strings. 0 = No, 1 = Yes
/ExcludeStr <String>	Specifies the excluded string when UseExcludeStr = 1
/RemovePostfixOnSearch <0 1>	Remove Ascii/Unicode Postfix On Google Search. 0 = No, 1 = Yes

<code>/ShowFoundDuringScan <0 1></code>	Show found items during the scan process. 0 = No, 1 = Yes
<code>/Undecorate <0 1></code>	Undecorate C++ function names. 0 = No, 1 = Yes
<code>/stext <Filename></code>	Save the functions list into a regular text file.
<code>/stab <Filename></code>	Save the functions list into a tab-delimited text file.
<code>/scomma <Filename></code>	Save the functions list into a comma-delimited text file.
<code>/stabular <Filename></code>	Save the functions list into a tabular text file.
<code>/shtml <Filename></code>	Save the functions list into HTML file (Horizontal).
<code>/sverhtml <Filename></code>	Save the functions list into HTML file (Vertical).
<code>/sxml <Filename></code>	Save the functions list to XML file.
<code>/sort <column></code>	<p>This command-line option can be used with other save options for sorting by the desired column. If you don't specify this option, the list is sorted according to the last sort that you made from the user interface. The <code><column></code> parameter can specify the column index (0 for the first column, 1 for the second column, and so on) or the name of the column, like "Function Name" and "Address". You can specify the '~' prefix character (e.g: "~Function Name") if you want to sort in descending order. You can put multiple <code>/sort</code> in the command-line if you want to sort by multiple columns.</p> <p>Examples: <code>dllexp.exe /from_files</code> <code>"c:\windows\system32\shell32.dll" /shtml</code> <code>c:\temp\funclist.html /sort "~Function Name"</code> <code>dllexp.exe /from_files</code> <code>"c:\windows\system32\shell32.dll" /shtml</code></p>

	c:\temp\funclist.html /sort 1 /sort ~2
/nosort	When you specify this command-line option, the list will be saved without any sorting.

Examples:

```
dllexp.exe /from_files "c:\windows\system32\s*.dll"
```

```
dllexp.exe /from_files "c:\windows\system32\sh*.dll" /filter "shell open" /shtml  
"c:\temp\func.html"
```

```
dllexp.exe /from_process 2020 /nofilter
```

```
dllexp.exe /from_std
```

License

This utility is released as freeware. You are allowed to freely distribute this utility via floppy disk, CD-ROM, Internet, or in any other way, as long as you don't charge anything for this. If you distribute this utility, you must include all files in the distribution package, without any modification !

Disclaimer

The software is provided "AS IS" without any warranty, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The author will not be liable for any special, incidental, consequential or indirect damages due to loss of data or any other reason.

Translating this utility to other languages

In order to translate this utility to other language, follow the instructions below:

1. Run DLL Export Viewer with /savelangfile parameter:
dllexp.exe /savelangfile
A file named dllexp_lng.ini will be created in the folder of DLL Export Viewer utility.
2. Open the created language file in Notepad or in any other text editor.

3. Translate all string entries to the desired language. Optionally, you can also add your name and/or a link to your Web site. (TranslatorName and TranslatorURL values) If you add this information, it'll be used in the 'About' window.
4. After you finish the translation, Run DLL Export Viewer, and all translated strings will be loaded from the language file.
If you want to run this utility without the translation, simply rename the language file, or move it to another folder.

Feedback

If you have any problem, suggestion, comment, or you found a bug in my utility, you can send a message to nirsofer@yahoo.com