

PE File Browser

by

Software Verify

Copyright © 2017-2025 Software Verify Limited



PE File Browser

PE File contents inspector

by Software Verify Limited

Welcome to the PE File Browser software tool.

PE File Browser is a software tool that allows you to inspect the contents of PE files (.exe and .dll files).

We hope you will find this document useful.

DbgHelp Browser Help

Copyright © 2015-2025 Software Verify Limited

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: May 2025 in United Kingdom.

L

Table of Contents

	Foreword	1
Part I	How to get PE File Browser	2
Part II	What does PE File Browser do?	4
Part III	Menu	6
1	File	
2	Inspect	
3	Software Updates	
4	Help	11
Part IV	The user interface	13
1	The data display	14
	File Header	
	Optional Header	16
	Optional Header Directories	
	Additional Information	
	Net Header	
	Net NetaData	
	Imported Modules and Functions	
	Delay Loaded Modules	
	All Dependent Modules	27
	Imported Functions	28
	Exported Functions	
	Sections / Segments	
	Data Bounds	
	FPO Data	
	Version Information	
	Manifest	
	Digital Signatures	
	Thread Local Storage	
	Resources	39
	Debug Information	39
	PDB Symbols	40
	COFF Symbols	
	Coueview Symbols	
	Misc Debug Data	
2	View Memory Dialog	
-	Soarch Momory Dialog	
3	Search memory Dialog	

Part V Command Line Interface

		Contents	П
1	I Alphabetic Reference		51
2	2 Usage Reference		52
	Index		0
	IIIdex		v

1	Foreword



1 How to get PE File Browser

PE File Browser is free for commercial use. PE File Browser can be downloaded for Software Verify's website at https://www.softwareverify.com/product/pe-file-browser/.

This help manual is available in Compiled HTML Help (Windows Help files), PDF, and online.

Windows Help	https://www.softwareverify.com/documentation/chm/peFileBrowser.chm
PDF	https://www.softwareverify.com/documentation/pdfs/peFileBrowser.pdf
Online	https://www.softwareverify.com/documentation/html/peFileBrowser/index.html

Whilst PE File Browser is free for commercial use, PE File Browser is copyrighted software and is not in the public domain.

You are free to use the software at your own risk.

You are not allowed to distribute the software in any form, or to sell the software, or to host the software on a website.

Contact

Contact Software Verify at:

Software Verify Limited Suffolk Business Park Eldo House Kempson Way Bury Saint Edmunds IP32 7AR United Kingdom

 email
 sales@softwareverify.com

 web
 https://www.softwareverify.com

 blog
 https://www.softwareverify.com/blog

 twitter
 http://twitter.com/softwareverify

Visit our blog to read our articles on debugging techniques and tools. Follow us on twitter to keep track of the latest software tools and updates.



2 What does PE File Browser do?

PE File Browser allows you to inspect the contents of a PE format file (.exe or .dll).

32 bit and 64 bit

PDB files created by 32 bit and 64 bit software are supported. On 64 bit Operating systems if a 64 bit PDB file is opened the 64 bit version PE File Browser is automatically started.

History

PE File Browser has been an internal tool at Software Verify for many years. We recently decided to make it a bit more user friendly and to make it available for public use.



3 Menu

The main menu contains three menus, File, Software Updates and Help.



3.1 File

The File menu controls loading of PE files, clearing the display and exiting the program.

Open PE File	
Close	~
Exit	

File menu > Open PE File... > loads a PE File (exe or dll) and displays information about the PE file.

File menu > Close > clear all results, unloads the PE file.

File menu > Exit > closes PE File Browser.

3.2 Inspect

The Inspect menu allows you to view arbitrary memory, or to search for memory.



Inspect menu **> View memory... >** view memory at an arbitrary address. The View Memory Dialog is displayed.

Inspect menu **> Search memory... >** search for a text string or a sequence of bytes. The Search Memory Dialog is displayed.

3.3 Software Updates

The Software Updates menu controls how often software updates are downloaded.

If you've been notified of a new software release to PE File Browser or just want to see if there's a new version, this feature makes it easy to update.

Check for software updates... Configure software updates... Set software update directory...

Software Updates menu **> Check for software updates >** checks for updates and shows the software update dialog if any exist

An internet connection is needed to be able to make contact with our servers.

Before updating the software, close the help manual, and end any active session by closing target programs.

If no updates are available, you'll just see this message:



Software Update dialog

If a software update is available for PE File Browser you'll see the software update dialog.



• **Download and install** > downloads the update, showing progress

Downloading PE File Browser 1.78	×		
Downloading PE File Browser 1.78			
	Stop		
5552 KB of 16831 KB (33.0%)			

Once the update has downloaded, PE File Browser will close, run the installer, and restart.

You can stop the download at any time, if necessary.

- Don't download... > Doesn't download, but you'll be prompted for it again next time you start PE File Browser
- Skip this version... > Doesn't download the update and doesn't bother you again until there's an
 even newer update
- Software update options... > edit the software update schedule

Problems downloading or installing?

If for whatever reason, automatic download and installation fails to complete:

• Download the latest installer manually from the software verify website.

Make some checks for possible scenarios where files may be locked by PE File Browser as follows:

- Ensure PE File Browser and its help manual is also closed
- Ensure any error dialogs from the previous installation are closed

You should now be ready to run the new version.

Software update schedule

PE File Browser can automatically check to see if a new version of PE File Browser is available for downloading.

Software Updates menu **> Configure software updates >** shows the software update schedule dialog

The update options are:

- never check for updates
- check daily (the default)
- check weekly
- check monthly

The most recent check for updates is shown at the bottom.

Software update schedule	?	×
OutputDebugStringChecker can check for software updat	tes on a regular b	oasis.
C Never check for updates		
C Check for updates every day		
Check for updates once a week		
C Check for updates once a month		
Most recent check for updates was performed on 2022-	07-23	
	ок са	ancel

Software update directory

It's important to be able to specify where software updates are downloaded to because of potential security risks that may arise from allowing the TMP directory to be executable. For example, to counteract security threats it's possible that account ownership permissions or antivirus software blocks program execution directly from the TMP directory.

The TMP directory is the default location but if for whatever reason you're not comfortable with that, you can specify your preferred download directory. This allows you to set permissions for TMP to deny execute privileges if you wish.

Software Updates menu **> Set software update directory >** shows the Software update download directory dialog

Software update download directory	?	×
Software updates will be downloaded to the location specified below:		
C: \Users\stephen\AppData\Local\Temp	Browse	
	Rese	t
OK	Cance	1

An invalid directory will show the path in red and will not be accepted until a valid folder is entered.

Example reasons for invalid directories include:

- the directory doesn't exist
- the directory doesn't have write privilege (update can't be downloaded)
- the directory doesn't have execute privilege (downloaded update can't be run)

When modifying the download directory, you should ensure the directory will continue to be valid. Updates may no longer occur if the download location is later invalidated.

• **Reset >** reverts the download location to the user's TMP directory

The default location is c:\users\[username]\AppData\Local\Temp

3.4 Help

The Help menu controls displaying this help document and displaying information about PE File Browser.

Help menu > About PE File Browser... > displays information about PE File Browser.

Help menu > Readme and Version History... > displays the readme and version history.

Help menu > Help Topics... > displays this help file.

Help menu > Help PDF... > displays this help file in PDF format.

Help menu **> Help on softwareverify.com... >** display the Software Verify documentation web page where you can view online documentation or download compiled HTML Help and PDF help documents.

Help menu **> Blog... >** display the Software Verify blog.

Help menu **> Library... >** display the Software Verify library - our best blog articles grouped by related topics.

Help menu > Contact customer support... > displays the options for contacting customer support.



Click a link to contact customer support.

Help menu **> How do I?... >** displays the options for asking us how to do a particular task.





4 The user interface

Enter topic text here.

4.1 The data display

The PE File Browser user interface is shown below.

PE File Browser x64 [E:\om\c\testApp File Help	ps\testDelayLoad1\Debug_x64\t	estDelayLoad1.exe]	-		×
Header	Name	Value			-
- Optional Header	File Name				
Optional Header Directories	PDB Name				
	File Signature	PE			
	File Type	Executable			
Modules	Machine Type	x64			
Imported Modules and Functions Delay Loaded Modules	Timestamp	Wed Jun 24 10:24:00 2020			
All Dependent Modules	Num symbols	0			
Functions	Pointer to symbols	0x0000000000000000			
Exported Functions	Sections	7			
Data Sections	Size of optional header	0x000000f0			
- Data Bounds	Characteristic	Relocations stripped (image has been bound using bind.exe or similar)			
	Characteristic	Executable File			
Version Information	Characteristic	64 bit machine			
Digital Signatures	Characteristic	Large address aware			
Symbols PDB Symbols COFF Symbols	<			3	>

The general principle guiding what data we choose to display and how we group the data is as follows:

- Is the data logically grouped inside the PE file? This covers data that is in the structure of the PE file, such as File Headers, Optional Headers, Sections, etc.
- Is the data logically grouped from the perspective of the user of the software? This covers data such as debugging symbols.

There have been a few areas where we've blended data together to either compress the display (the File Header section has a few additional entries) or to synthesize extra information not expressly provided by the PE file format but which can be determined from the data in the PE file (the Additional Information section is an example of this).

4.1.1 File Header

The File Header display shows the contents of the PE file File Header.

PE File Browser x64 [E:\om\c\testApps\native\testDelayLoad1\Debug_x64\testDelayLoad1.exe]			-		×
File Software Updates Help					
Header	Name	Value			-
- Optional Header	File Name				
Optional Header Directories	PDB Name				
- Net Information	File Signature	PE			
Net Header	File Type	Executable			
Modules	Machine Type	x64			
	Timestamp	Wed May 10 22:07:34 2023			
All Dependent Modules	Num symbols	0			
Functions	Pointer to symbols	0x000000000000000			
Exported Functions	Sections	7			
Data 	Size of optional header	0x000000f0			
Data Bounds	Characteristic	Relocations stripped (image has been bound using bind.exe or similar)			
- x64 Exception Handling	Characteristic	Executable File			
	Characteristic	64 bit machine			
Digital Signatures	Characteristic	Large address aware			
Thread Local Storage Resources	4			,	
Symbols 🗸				-	

File Name

The name of the PE file. This is often blank.

PDB Name

The name of the corresponding PDB file. This is often blank.

File Signature

This can be many values, but for a PE file, will be the value "PE".

File Type

This can be one of three values:

- DLL. A loadable module.
- Executable. An executable program.
- System. A system file.

Machine Type

This can be one of many values, although we typically only expect "i386" and "x64". Valid values are:

- Unknown
- i386
- x64
- MIPS R3000
- MIPS R4000
- MIPS R10000
- Alpha
- Power PC
- Hitatchi SH3
- Hitatchi SH4
- ARM
- MIPS WinCE v2
- Hitatchi SH3E
- Thumb
- IA64 (Merced/Itanium)
- MIPS 16
- MIPS FPU

• MIPS FPU 16

• ALPHA64

Timestamp

This is the time the PE file was created. This is often not set.

Num Symbols

The number of symbols embedded in the PE file.

Pointer to Symbols

Pointer to any symbols embedded in the PE file.

Sections

The number of different named sections in the PE file.

Size of optional header

The size of the optional header in the PE file.

Characteristics

A PE file can have many characteristics. The remainder of the File Header lists only the characteristics that have been set, one per line.

The following characteristics can be set. Many values can apply at the same time. Many of these characteristics are obsolete.

- Relocations stripped (image has been bound using bind.exe or similar)
- Executable File
- DLL
- System File
- UniProcessor systems only
- 32 bit machine
- 64 bit machine
- Line numbers stripped
- Local symbols stripped
- Debug info stripped (held in separate .DBG file)
- Aggressive working set trim
- Large address aware
- Machine is LSB precedes MSB
- Run from swap if image is on removable media
- Run from swap if image is on the network
- Machine is MSB precedes LSB

4.1.2 Optional Header

The Optional Header display shows the contents of the PE file Optional Header.

File Software Updates Heip Header File Header Optional Header Optional Header Optional Header Additional Information Net Header Net MetaData Modules Size of code 0x0000000 Net MetaData Modules Size of code 0x00000000 Net MetaData Net MetaData Modules Exported Functions Exported Functions Exported Functions Section alignment 0x0000000 4.00 KB MetaData MetaData Modules Section alignment 0x0000000 Headed Addues <ul< th=""><th>PE File Browser x64 [E:\om\c\testA</th><th>testDelayLoad1\Debug_x64\testDelayLoad1.exe] —</th><th></th><th>×</th></ul<>	PE File Browser x64 [E:\om\c\testA	testDelayLoad1\Debug_x64\testDelayLoad1.exe] —		×
- File Header - Optional Header/ Optional Header Directories - Magic Number 0x000020b 64 bit - Optional Header Directories - Linker Version 10.0 Visual Studio 2010 - Net Information - Size of code 0x00062200 392.50 KB - Net MetaData Size of code 0x00010400 113.00 KB Modules Size of uninitialised data 0x0000000 0 B - Inported Modules and Functions Entry point RVA 0x0000000 4.00 KB - Imported Functions Image base 0x0000010000 Fixed load address - Linge base 0x0000010000 KB	File Software Updates Help Header	me Value Description		^
Optional Header Drectories Linker Version 10.0 Visual Studio 2010 - Additional Information Size of code 0x00062200 392.50 KB - Net Information Size of code 0x0001c400 113.00 KB - Net MetaData Size of uninitialised data 0x0000000 0 B Modules Entry point RVA 0x00002290 Entry point RVA - Linported Functions Base of code 0x0000000 Fixed load address - Imported Functions Image base 0x000000000 Fixed load address Section alignment 0x0001000 4.00 KB	- File Header	ic Number 0x000020b 64 bit		
Addiconal Information Size of code 0x00062200 392.50 KB Net Header Size of initialised data 0x0001c400 113.00 KB Modules Size of uninitialised data 0x0000000 0 B Imported Modules and Functions Entry point RVA 0x00002290 Functions Base of code 0x000001000 4.00 KB Imported Functions Image base 0x00000140000000 Fixed load address Exported Functions Section alignment 0x00001000 4.00 KB	Optional Header Directories	er Version 10.0 Visual Studio 2010		
Imported Functions Size of initialised data 0x0001c400 113.00 KB Imported Modules and Functions Size of uninitialised data 0x0000000 0 B Imported Modules and Functions Entry point RVA 0x00002290 Impendent Modules Base of code 0x00001000 4.00 KB Imported Functions Image base 0x00000140000000 Fixed load address Exported Functions Section alignment 0x00001000 4.00 KB		of code 0x00062200 392.50 KB		
Interview Size of uninitialised data 0x0000000 0 B Imported Modules Functions Entry point RVA 0x00002290 All Dependent Modules Base of code 0x00001000 4.00 KB Functions Image base 0x00000140000000 Fixed load address Exported Functions Section alignment 0x0001000 4.00 KB		of initialised data 0x0001c400 113.00 KB		
Imported Functions Entry point RVA 0x00002290 All Dependent Modules Base of code 0x00001000 4.00 KB Functions Image base 0x00000140000000 Fixed load address Exported Functions Section alignment 0x0001000 4.00 KB	Modules	of uninitialised data 0x00000000 0 B		
All Dependent Modules Base of code 0x00001000 4.00 KB Functions Image base 0x000000140000000 Fixed load address Exported Functions Section alignment 0x00001000 4.00 KB	Imported Modules and Functions	y point RVA 0x00002290		
Functions Image base 0x00000014000000 Fixed load address _Imported Functions Section alignment 0x00001000 4.00 KB	- All Dependent Modules	e of code 0x00001000 4.00 KB		
Exported Functions Section alignment 0x00001000 4.00 KB	Functions Imported Functions	ge base 0x000000140000000 Fixed load address		
	Exported Functions	ion alignment 0x00001000 4.00 KB		
Upta	Data — Sections / Segments	alignment 0x0000200 512 B		
Data Bounds O.S. version 5.2 Windows XP	- Data Bounds	version 5.2 Windows XP		
- x64 Exception Handling Image version 0.0	- x64 Exception Handling	ge version 0.0		
Version Information Subsystem version 5.2 Windows XP		system version 5.2 Windows XP		
- Digital Signatures Win32 version 0	- Digital Signatures	32 version 0		
Thread Local Storage Size of image 0x00085000 532.00 KB	Thread Local Storage	of image 0x00085000 532.00 KB		~
Symbols v <	Symbols		3	>

Magic Number

This is the magic number stored in the PE file to identify the file type.

Valid values are:

- 0x0107 ROM
- 0x010B 32 bit
- 0x020B 64 bit

Linker Version

This is the version of the linker used to build the image.

The description field indicates which version of Visual Studio may have been used to build this file.

If Visual Studio 2012 is set to build using Visual Studio 2010 libraries this field will indicate Visual Studio 2010.

If a different compiler/linker is used (Embarcadero, Delphi, MingW) the description field will be incorrect.

Size of code

Size of the executable code in the PE file.

Size of initialised data

Size of the initialised data in the PE file.

Size of uninitialised data

Size of the uninitialised data in the PE file.

Entry point RVA

The relative virtual address of the entry point to the executable. This is an offset from the start of the PE file.

This is DIIMain for a DLL, or the start of the program for an EXE.

Base of code

This is the relative virtual address of the start of the code in the PE file.

Base of data

This is the relative virtual address of the start of the data in the PE file.

Section alignment

Sections are aligned at boundaries that are a multiple of this value after the PE file is mapped into memory.

File alignment

Sections are aligned at boundaries that are a multiple of this value before the PE file is mapped into memory.

O.S. version

The minimum version of the Windows operating system that this software can execute.

Image version

A user definable version number.

Subsystem version

The windows subsystem version number.

Win32 version The Win32 version number

Size of image

The size of the parts of the PE file that the DLL loader needs to be concerned with.

Size of headers

The size of the PE header and the section (object) table.

Subsystem

The Windows subsystem. Valid values are:

- Unknown
- Native
- Windows GUI
- Windows Character UI
- OS2 Character UI
- Posix Character UI
- Windows 95/98 Driver
- Windows CE GUI
- EFI Application
- EFI Boot Service Driver
- EFI Runtime Driver
- EFI ROM
- XBox
- Windows Boot Application

Characteristics

These are characteristics that affect how the PE file is treated. This is a bitmask. Many values can apply at the same time.

Valid characteristics are:

- · Call when DLL is first loaded into a process's address space
- Call when a thread terminates

- Call when a thread starts up
- Call when DLL exits
- Unknown(0x0010)
- Unknown(0x0020)
- Relocatable
- Fixed load address
- Force code integrity check
- Data execution prevention compatible
- No isolation
- No SEH
- Do not bind
- WDM driver
- Reserved(0x4000)
- Terminal server aware

Stack reserve

This is the amount of memory reserved for the first thread's stack.

Stack commit

This is the amount of memory committed for the first thread's stack.

Heap reserve

This is the amount of memory reserved for the process heap.

Heap commit

This is the amount of memory committed for the process heap.

Loader flags

The Loader flags value is obsolete. This value is normally set to 0.Known values appear to be related to debugging support.Invoke a breakpoint instruction before starting the processInvoke a debugger on the process after it's been loaded

4.1.3 Optional Header Directories

The Optional Header Directories display shows the contents of the PE file Optional Header Directories.

PE File Browser x64 [E\com\c\testApps\native\testDelayLoad1\Debug_x64\testDelayLoad1.exe] File							×
Header	Name	RVA	Address	Size			
- File Header	Export	0x00000000					
Optional Header Directories	Import	0x00081000	0x000000140081000	0x0000050			
- Additional Information	Resource	0x00084000	0x000000140084000	0x00000459			
	Exception	0x0007d000	0x00000014007D000	0x00002ad8			
.Net MetaData	Security / Certificate table	0x00000000					
Modules	Base Relocation	0x0000000					
- Imported Modules and Functions	Debug	0x00064ce0	0x000000140064CE0	0x0000001c			
- All Dependent Modules	Architecture Specific	0×00000000	0.0000000110001020	0.0000012			
Functions	Global Deinter	0+00000000					
Imported Functions	Thread Level Channel	0.00000000					
Exported Functions	Inread Local Storage	0x0000000					
- Sections / Segments	Load Configuration	0x00000000					
Data Bounds	Bound Import	0x0000000					
FPO Data	Import Address Table	0x000814e0	0x00000001400814E0	0x00000490			
	Delay Load Import Table	0x00083000	0x000000140083000	0x0000060			
Manifest	CLR / COM Runtime Table	0x00000000					
Digital Signatures	Reserved	0x00000000					
Thread Local Storage							
Symbols M							
<pre> *</pre>							

Each optional header directory is listed, with the RVA, the actual address and size. Address and Size are only displayed if the RVA is valid (non-zero).

The following header directories are displayed.

- Export
- Import
- Resource
- Exception
- Security
- Base Relocation
- Debug
- Architecture Specific
- Global Pointer
- Thread Local Storage
- Load Configuration
- Bound Import
- Import Address Table
- Delay Load Import Table
- COM Runtime Table
- Reserved

Context Menu

A context menu provides a single option:

View data...

Clicking **View data...** opens a memory inspection dialog, allowing you to view the memory as BYTEs, WORDs, DWORDs or QWORDs.

Optional Header Directory "Exception", 0x00007FF9D74BC000 1380 bytes	×
BYTE WORD DWORD QWORD	
0x00007FF9D74BC000 00 10 00 02 3 11 00 00 70 96 00 00 30 11 00 00 #p0 0x00007FF9D74BC010 93 11 00 00 5c 96 00 00 a0 11 00 00 6c 17 00 00 0x00007FF9D74BC020 6c 97 00 00 70 17 00 00 e9 17 00 00 7c 94 00 00 1p 0x00007FF9D74BC030 70 17 00 00 c5 18 00 00 5c 96 00 00 d0 18 00 00 1p 0x00007FF9D74BC030 7c 94 00 00 b0 24 00 00 10 24 00 00 a3 24 00 00 \$\$\$\$\$\$\$ 0x00007FF9D74BC030 7c 94 00 00 b0 24 00 00 61 25 00 00 11 26 00 00 a&\$\$\$\$\$ 0x00007FF9D74BC070 0a 29 00 00 10 96 00 00 28 96 00 00 01 22 29 00 00 a&\$\$\$\$\$\$\$	^
	ose

4.1.4 Additional Information

The Additional Information displays some additional information about a PE that can be determined from a PE file, but which is not an obvious value in the PE file.

PE File Browser x86 [E:\om\c\testApps\dotnetCore\Bank\Bank\bin\Debug\netcoreapp2.1\Bank.dll]					
Header	Name	Value		^	
- Optional Header	VB6	No			
Optional Header Directories Additional Information	Native module	No			
	Mixed-mode module	No			
.Net Header .Net MetaData	Entry point	.Net entry point			
Modules					
Imported Modules and Functions Delay Loaded Modules	Module contains code	Yes			
All Dependent Modules	Module contains resources	Yes			
Functions Imported Functions	Module is only resources	No			
Exported Functions	Module is UPX compressed	No			
Data					
- Data Bounds	Module has Data Execution Prevention	No			
FPO Data		N.			
Version Information	Module has Control Flow Guard	NO			
Manifest					
Digital Signatures	Console UI	Yes			
Resources					
Symbols	Heer Win22 Service ADI	No		~	
Debug Information V	<		>		

VB6

Is this module a Visual Basic 6 application?

Native module

Is this a native executable?

Mixed-mode module

Is this a mixed mode executable (contains native code and .Net code)?

Entry point

The entry point field describes the type of entry point for the DLL.

Valid values are:

- No entry point
- .Net entry point
- Native entry point
- Unknown entry point type

Module contains code

Does this module contain executable code?

Module contains resources Does this module contain resources?

Module is only resources

Does this module only contain resources?

Module is UPX compressed

This module has been compressed using the UPX compression algorithm.

Module has Data Execution Prevention

This module implements data execution prevention security measures.

Module has Control Flow Guard

This module implements control flow guard security measures.

Console UI

Is this module a console application?

Uses Win32 Service API

This EXE/DLL uses functions in the Win32 Service API. This EXE/DLL may be part of a service.

Uses Visual Leak Detector

This EXE/DLL uses Visual Leak Detector.

Uses Address Sanitizer

This EXE/DLL uses Address Sanitizer.

4.1.5 .Net Information

The .Net Information displays some .Net specific information about a PE that can be determined from a PE file, but which is not an obvious value in the PE file.

Re File Browser x86 [E:\om\c\testApps\dotnetCore\Bank\Bank\bin\Debug\netcoreapp2.1\Bank.dll]							
File Help Software Updates Test							
Header	^	Name	Value				
File Header Optional Header		CLR Version	.Net CLR 2.0 (VS 20122022)				
- Optional Header Directories							
- Additional Information .Net Information		.Net x86 CPU	No				
.Net Header		.Net x64 CPU	No				
Modules		.Net Any CPU	Yes				
Imported Modules and Functions		.Net Any CPU 32 bit preferred	No				
- All Dependent Modules							
Functions		.Net Module	Yes				
- Exported Functions		.Net Linked Module	Yes				
Data Sections / Segments							
Data Bounds		.Net Core (Framework Dependent)	Yes				
FPO Data Version Information		.Net Core (Self Contained)	No				
Manifest							
Digital Signatures Thread Local Storage		Native Loads .Net Runtime	No				
Resources							
Debug Information	v						
< >							

CLR Version

If this module uses the .Net runtime, this entry displays which version of the CLR is required. Valid values are:

- .Net CLR 1.0 (VS 2002)
- .Net CLR 1.1 (VS 2003)
- .Net CLR 2.0 (VS 2005..2010)
- .Net CLR 2.0 (VS 2012..2017)
- .Net CLR 2.0 or later

The last value will be displayed if a CLR version that is not recognised is encountered.

.Net x86 CPU

Is this module a .Net module that is compiled for use on x86 processors?

.Net x64 CPU

Is this module a .Net module that is compiled for use on x86 processors?

.Net Any CPU

Is this module a .Net module that is compiled for use on any CPU?

.Net Any CPU 32 bit preferred

Is this module a .Net module that is compiled for use on any CPU but prefers 32 bit processors?

.Net Module

Is this module a pure .Net module (contains no native code)?

.Net Linked Module

Is this module linked to the .Net runtime?

.Net Core (Framework Dependent)

Is this module built for use with .Net Core as a framework dependent DLL?

.Net Core (Self Contained)

Is this module built for use with .Net Core as a self contained executable?

Native Loads .Net Runtime

Is this a native module that loads the .Net runtime to executable .Net code?

4.1.6 .Net Header

The .Net Header displays some information the .Net Header (if any) that is present in the PE file.

WPE File Browser x86 [E\om\<\testApps\dotnet\Bank\Bank\Bank\bin\Debug\netcoreapp2.1\Bank.dll] File Software Updates Help						×
PE File Browser x86 [E-kom/cktest/ File Software Updates Help Header -Optional Header -Net Information -Net Information -Net MetaData Modules Limported Modules and Functions -Delay Loaded Modules Al Dependent Modules Functions -Denyted Functions Data -Sections / Segments -Data Bounds -Fine Data Version Information -Marifest Version Information -Marifest Version Information	Apps\	dotnet\Bank\Bank\bin\Debug Name Framework Version CLR Header Version Size Flags .Net Entry Point Token MetaData Resources StrongNameSignature CodeManagerTable VTableFixups ExportAddressTableJumps ManagedNativeHeader	Netcoreapp2.1\Bank.d v4.0.30319 2.5 0x00000048 0x0000001 0x06000007 0x000022f8	JII] Size COMIMAGE_FLAGS_ILONLY 0x00000810 0x00000810		×
Digital Signatures Thread Local Storage Resources Symbols Debug Information	~					

Framework version The .Net Framework version

MetaData Version

The .Net Metadata version.

MetaData Flags

The .Net Metadata flags.

Size

Size of the .Net Header

Flags

CLR flags. As specified by .corflags directive in ILASM or /FLAGS with ILASM compiler.

.Net Entry Point RVA

The relative virtual address that defines the native entry point for this .Net application. This is not displayed if a .Net Entry Point Token is defined.

.Net Entry Point Token

The .Net method token that identifies the startup function for this application. This is not displayed if a .Net Entry Point RVA is defined.

MetaData

Relative virtual address (RVA) of the .Net Metadata inside the PE file.

Resources

Relative virtual address (RVA) of the managed resources inside the PE file.

StrongNameSignature

Relative virtual address (RVA) of the strong name signature for this PE file.

CodeManagerTable

Relative virtual address (RVA) of the Code Manager table.

VTableFixups

Relative virtual address (RVA) of the virtual fixup table.

ExportAddressTableJumps

Relative virtual address (RVA) of jump thunks.

ManagedNativeHeader

Relative virtual address (RVA) of the managed native header. Reserved for precompiled images.

For a more detailed explanation of the fields mentioned in this topic see Table 4-6, Chapter 4 of Expert .Net 2.0 Assembler by Serge Lidin.

4.1.7 .Net MetaData

The .Net MetaData displays some information the .Net MetaData (if any) that is present in the PE file.

🐺 PE File Browser x86 [E:\om\c\testApps\	dotnet\Bank\Bank\bin\Debu	g\netcoreapp2.1\Bank.d	11]	-	\times
File Software Updates Help					
Header	Name	Value	Size		
	Framework Version	v4.0.30319			
- Optional Header Directories	MetaData Version	1.1			
- Net Information	MetaData Flags	0x00000000			
Net Header	Number of Streams	5			
Modules	Stream 1	#~			
Imported Modules and Functions	Stream 2	#Strings			
- All Dependent Modules	Stream 3	#US			
Functions	Stream 4	#GUID			
Exported Functions	Stream 5	#Blob			
Data Sections / Segments					
- Data Bounds					
FPO Data					
Version Information					
Manifest					
Digital Signatures					
Thread Local Storage					
Symbols Debug Information					
Debug Information					

Framework version

The .Net Framework version

MetaData Version

The .Net Metadata version.

MetaData Flags

The .Net Metadata flags.

Number of Streams

How many streams are in the .Net MetaData.

Stream ??

For each metadata stream present the name of the metadata stream is shown.

There are typically no more than 5 streams. The expected names are #~, #Strings, #US, #GUID, #Blob.

4.1.8 Imported Modules and Functions

The Imported Modules and Functions display shows the modules that are linked to the PE file. The display also shows the functions imported from each module.



In the image above you can see that this PE file imports functions from 3 DLLs. The first DLL in the list, KERNEL32.DLL imports 88 functions to the PE file.

4.1.9 Delay Loaded Modules

The Delay Loaded Modules display shows the modules that are linked to the PE file but which are delay loaded (they load when the a delay loaded function is called).



In the image above you can see that this PE file delay loads two DLLs, ole32.dll and shlwapi.dll. Each DLL in this example imports 2 functions.

4.1.10 All Dependent Modules

The All Dependent Modules display shows the all the modules that the PE file depends on.

PE File Browser x64 [E:\om\c\testApps\native\testDelayLoad1\Debug_x64\testDelayLoad1.exe] -	×
ïle Software Updates Help	
Teader	~
Symbols V I	>

If you expand the entries you can see which DLLs those DLLs depend on.

Any DLLs that can't be found on the \$PATH or in the same folder as the main PE file will be displayed in red and marked as a "Missing DLL".

4.1.11 Imported Functions

The Imported Functions display shows the all the functions imported to the PE file.

💐 PE File Browser x64 [E:\om\c\testApp	s\native\testDelayLoad1\Debug_x64\testDelayLoad1.exe]		- 🗆 X
File Software Updates Help			
Header	Undecorate symbols		Filter
- Optional Header	Function		^
Optional Header Directories Additional Information	CloseHandle	KERNEL32.dll	
Net Information	CreateFileW	KERNEL32.dll	
Net Header	DecodePointer	KERNEL32.dll	
·····	DeleteCriticalSection	KERNEL32.dll	
- Imported Modules and Functions	EncodePointer	KERNEL32.dll	
- Delay Loaded Modules	EnterCriticalSection	KERNEL32.dll	
All Dependent Modules Functions	EnumSystemLocalesA	KERNEL32.dll	
- Imported Functions	ExitProcess	KERNEL32.dll	
Exported Functions	FatalAppExitA	KERNEL32.dll	
- Sections / Segments	FIsAlloc	KERNEL32.dll	
Data Bounds	FIsFree	KERNEL32.dll	
FPO Data	FlsGetValue	KERNEL32.dll	
- Version Information	FIsSetValue	KERNEL32.dll	
Manifest	FlushFileBuffers	KERNEL32.dll	
	FreeEnvironmentStringsW	KERNEL32.dll	
Resources	FreeLibrary	KERNEL32.dll	~
Symbols 🗸	<		>

Function

The name of the imported function.

Module

The name of the module the function is imported from.

Sorting

The data can be sorted by Function or Module. Click the column header to select sorting by Function or Module. Click the same header to reverse the sort direction.

Filter

If a filter string is typed into the edit field and the **Filter** button is clicked the display shows on functions that contain the filter string. The search is case sensitive.

4.1.12 Exported Functions

The Exported Functions display shows the all the functions exported from the PE file.

Software Opdates Help					
ader ^	Undecorate symbols				Fil
- Optional Header	Function	Forwarded Function	Address	Ordinal	Hint
- Additional Information	EulaDIgProc		0x00007FFB542B84E0	0x0000001	0x00000000
	RunDIIEntry		0x00007FFB542BBD70	0x0000002	0x0000001
	SymbolServer		0x00007FFB542B9B80	0x0000016	0x0000002
	SymbolServerByIndex		0x00007FFB542BA080	0x0000003	0x0000003
	SymbolServerByIndexW		0x00007FFB542BA000	0x00000017	0x00000004
Delay Loaded Modules	SymbolServerClose		0x00007FFB542B9660	0x0000018	0x00000005
- All Dependent Modules	SymbolServerDeltaName		0x00007FFB542BAF70	0x0000004	0×0000006
Imported Functions	SymbolServerDeltaNameW		0x00007FFB542BAE50	0x0000005	0×0000007
Exported Functions	SymbolServerGetIndexString		0x00007FFB542BB720	0x0000019	0x0000008
- Sections / Segments	SymbolServerGetIndexStringW		0x00007FFB542BB6A0	0x0000001a	0x0000009
Data Bounds	SymbolServerGetOptionData		0x00007FFB542BA800	0x000001b	0x0000000a
- FPO Data - x64 Exception Handling	SymbolServerGetOptions		0x00007FFB542BA7E0	0x000001c	0x000000b
Version Information	SymbolServerGetSupplement		0x00007FFB542BB290	0x0000006	0x000000c
Manifest	SymbolServerGetSupplementW		0x00007FFB542BB0A0	0x0000007	b0000000x0
Digital Signatures	SymbolServerGetVersion		0x00007FFB542BAC90	0x0000008	0x0000000e
- Inread Local Storage	Cumbal Canada Chana		0.000075585.0088550	0.00000000	0.00000006

Function

The name of the exported function.

Forwarded Function

If function is being forwarded from another DLL, it is listed in this column.

Address

The address of the exported function.

Ordinal

The ordinal of the exported function.

Hint

The ordinal hint of the exported function.

Sorting

The data can be sorted by any column. Click the column header to select sorting by that column. Click the same header to reverse the sort direction.

Filter

If a filter string is typed into the edit field and the **Filter** button is clicked the display shows on functions that contain the filter string. The search is case sensitive.

4.1.13 Sections / Segments

The Sections display shows the all the sections in the PE file.

Sections are also known as segments, often mentioned in MAP files and in the context of debugging information.

Symbol information is often provided as an offset and a segment. The symbol address is the module load address + the segment offset inside the dll, plus the symbol offset inside the segment.

Header	^	Segment 🖉	Name	Relocations	Line Numbers	Length	Virtual	Raw Size	Characte
- File Header - Optional Header		1	.text	0	0	0x000141df	0x00001000	0x00014200	Code, Exec
- Optional Header Directories		2	rdata	0	0	0x00010e64	0x00016000	0x00011000	Init data R
- Additional Information			data	0	0	0+00022264	0,00027000	0+00000600	Init data, P
Net Information		2	.data	U	0	0x00025504	0x00027000	0x0000000	init data, K
Net Header		4	.pdata	0	0	0x00000fe4	0x0004b000	0x00001000	Init data, R
.Net MetaData		5	.didat	0	0	0x000001c8	0x0004c000	0x00000200	Init data, R
Modules		6	.mrdata	0	0	0x00002d30	0x0004d000	0x00002e00	Init data, R
		7	rsrc	0	0	0x00001998	0x00050000	0x00001a00	Init data R
All Dependent Modules			i i i i i i i i i i i i i i i i i i i	•	0	0.000000.0	0.00052000	0.00000.00	Lin Lu D
unctions		8	reloc	U	U	0x000009e0	0x00052000	0x00000a00	Init data, D
- Imported Functions									
Exported Functions									
ata									
Sections / Segments									
Data Bounds									
FPO Data									
Version Information									
Manifect									
Mannesc									
Digital Signatures									
- Digital Signatures - Thread Local Storage									

Segment

The segment identifier.

Name

The section name.

Relocations

The number of relocations in this section.

Line Numbers

The number of line numbers in this section.

Physical

The physical address of this section.

Virtual

The virtual address of this section.

Size

The size of this section.

Characteristics

The characteristics of this section. This is a bitmask. Many values can apply at the same time. Some of these values are obsolete.

- Code
- Init data
- Uninit data
- Comments/info
- Not part of image
- COMDAT
- Reset speculative exception
- GP relative data
- FAR data
- Purgeable data
- 16 bit data

- Locked data
- Preloadable data
- Extended relocations
- Discardable
- Not cacheable
- Not pageable
- Shareable
- Executable
- Readable
- Writeable
- Alignment:1
- Alignment:2
- Alignment:4
- Alignment:8
- Alignment:16
- Alignment:32
- Alignment:64
- Alignment:128
- Alignment:256
- Alignment:512
- Alignment:1024
- Alignment:2048
- Alignment:4096
- Alignment:8192

The alignment values are exclusive. Only one alignment value can be applied.

Sorting

The data can be sorted by any column. Click the column header to select sorting by that column. Click the same header to reverse the sort direction.

Context Menu

A context menu provides a single option:



Clicking **View data...** opens a memory inspection dialog, allowing you to view the memory as BYTEs, WORDs, DWORDs or QWORDs. For executable code a disassembly view is provided.

Section / Segmer	nt 1, 0x00007FF9D74	B1000 22528 bytes		×
BYTE	WORD	DWORD	QWORD Disassembly	
0x0007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E 0x00007FF9D74E	31000 48 89 53 31010 89 7c 24 31020 79 50 44 31030 48 8b f3 31040 df 3b di 31050 fb 45 01 31060 60 48 8i 31070 d0 89 44 31080 d7 49 8i 31090 8b 46 44 31080 0f 47 di 31000 6f 47 di 31000 0f 47 di 31000 0f 47 di 31000 0f 47 di 31000 0f 47 di 31000 6d 10 di 31000 6d 10 di 31000 6d 10 di 31000 6d 10 di <td>24 08 48 89 6c 40 41 55 54 41 55 8b 59 58 42 85 59 54 42 8b 59 58 45 33 54 60 54 15 34 44 66 15 5a 1c 56 68 89 45 42 85 66 89 45 46 85 66 89 45 46 85 66 89 45 46 85 66 88 94 46 85 66 74 18 48 84 85 67 61 89 75 61 89 75 61 89 75 61 89 74 84 84 80 74 18 48 80 74 18 48 80 74 14 42 80 74 14 42 80 74 14 42 80 32 86 67 24 48 66</td> <td>24 10 48 89 74 24 18 48 41 56 48 83 ec 20 48 8b 6a 10 45 8b e0 48 8b ea 48 8b 59 48 8b 42 18 2b F6 85 db 74 08 41 83 f8 2b c3 89 42 18 48 8b 41 2b c3 89 42 18 48 8b 47 2b c3 89 42 18 48 8b 47 2b c3 89 42 18 48 8b 47 2b c3 89 42 18 48 8b 47 3b c3 48 8b 74 14 4c 8b c3 48 8b 2b c3 48 8b 2b c3 48 8b 4b c3 48 8b 47 4b c4 63 94 65 8b c4 8 2b c5 82 b df 3b d8 2b c4 80 c5 74 11 c4 c4 c5 c5 c5 24 40 4c 89 c4 c4 c5 c5 24 40 4c 89 c4 c5 c5 c5 26 48 8b 74 24 50 c5 c5 c5 c5 26 c5 c5</td> <td>Close</td>	24 08 48 89 6c 40 41 55 54 41 55 8b 59 58 42 85 59 54 42 8b 59 58 45 33 54 60 54 15 34 44 66 15 5a 1c 56 68 89 45 42 85 66 89 45 46 85 66 89 45 46 85 66 89 45 46 85 66 88 94 46 85 66 74 18 48 84 85 67 61 89 75 61 89 75 61 89 75 61 89 74 84 84 80 74 18 48 80 74 18 48 80 74 14 42 80 74 14 42 80 74 14 42 80 32 86 67 24 48 66	24 10 48 89 74 24 18 48 41 56 48 83 ec 20 48 8b 6a 10 45 8b e0 48 8b ea 48 8b 59 48 8b 42 18 2b F6 85 db 74 08 41 83 f8 2b c3 89 42 18 48 8b 41 2b c3 89 42 18 48 8b 47 2b c3 89 42 18 48 8b 47 2b c3 89 42 18 48 8b 47 2b c3 89 42 18 48 8b 47 3b c3 48 8b 74 14 4c 8b c3 48 8b 2b c3 48 8b 2b c3 48 8b 4b c3 48 8b 47 4b c4 63 94 65 8b c4 8 2b c5 82 b df 3b d8 2b c4 80 c5 74 11 c4 c4 c5 c5 c5 24 40 4c 89 c4 c4 c5 c5 24 40 4c 89 c4 c5 c5 c5 26 48 8b 74 24 50 c5 c5 c5 c5 26 c5	Close

4.1.14 Data Bounds

The Data Bounds display shows the all the different data and code area in the PE file.

💐 PE File Browser x64 [E:\om\c\memo	ory32	\tabserv\debug_x64\	dbgHel	p\vs14.0\symsrv.	dll]			-	×
File Software Updates Help									
Header	•	Low /	Hia	h	Туре	Init	Writeable		
File Header		0x00001000	0x00	0151df	Code	No	No		
- Optional Header		0x00016000	0x00	026e64	Data	Ves	No		
- Additional Information		0+00027000	0,00	04-264	Data	Vec	Vor		
Net Information		0x00027000	0,00	048304	Data	Tes	TES		
Net Header		0x0004b000	0x00	04bte4	Data	Yes	No		
		0x0004c000	0x00	04c1c8	Data	Yes	Yes		
Modules		0x0004d000	0x00	04fd30	Data	Yes	No		
- Delay Loaded Modules		0x00050000	0x00	051998	Data	Yes	No		
All Dependent Modules		0x00052000	0x00	0529e0	Data	Yes	No		
Functions									- 1 - 1
Imported Functions									
Exported Functions									
Data									
- Data Bounds									
FPO Data									
- x64 Exception Handling									
Version Information	1.								
Manifest									
Digital Signatures									
Resources									
Symbols									
< >									

Low

The low RVA (offset from the start of the PE file) of the data bound.

High

The high RVA (offset from the start of the PE file) of the data bound.

Туре

This indicates if this area of memory is code or data.

Init

Is this area of memory initialised?

Writeable

Is this area of memory writeable?

Sorting

The data can be sorted by any column. Click the column header to select sorting by that column. Click the same header to reverse the sort direction.

Context Menu

A context menu provides a single option:

View data...

Clicking **View data...** opens a memory inspection dialog, allowing you to view the memory as BYTEs, WORDs, DWORDs or QWORDs. For executable code a disassembly view is provided.

🔳 Data Bounds 0	, 0x00007Ff	F9D74	B1000	22462	2 byte	es													×
BYTE 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7 0x00007FF9D7	481000 481010 481020 481030 481040 481050 481050 481080 481080 481080 481080 481080	ORD 48 8 89 7 79 5 48 8 df 3 fb 4 df 3 df 4 fb 4 df 3 df 4 fb 4 df 4 df 4 df 4 df 4 df 4 df 4 df 4 df	9 5c c 24 0 48 b f1 b f1 5 0f 8 85 9 46 9 8b 6 48 9 8b 6 48 4 48 7 48 c 2b 6 48	22402 24 20 8b 48 0f 44 c0 68 68 68 68 68 68 68 68 68 68 68 68 68	08 41 59 3b 47 e6 74 89 e8 3b 7e db 89 8b	es A8 54 58 68 01 11 45 79 68 58 74 45 58 74 45 58 74 58 58 60 11 45 58 58 60 11 45 58 58 60 11 45 58 58 58 60 10 58 58 58 58 58 58 58 58 58 58	89 41 4c 76 45 5a 8b 4c 56 75 8b 08 18 48	6c 55 8b 04 33 1c 49 85 00 61 45 41 48 8b	QWO 24 48 f6 2b 68 db 00 48 18 83 83 84 d7	RD 10 56 10 8b 85 c3 44 74 4c 8b 8c 46 ff	48 48 45 59 db 89 8b 14 03 7e 5e fb 60 d0	Di 89 83 8b 48 74 42 c3 4c eb 40 58 45 48 89	sass 74 ec e0 8b 08 18 48 48 48 48 2b 0f 85 46	eml 24 20 48 42 41 48 8b c3 03 39 df 44 c0 68	bly 18 48 8b 18 8b d7 48 fb 46 3b e6 74 89	48 8b ea 2b f8 41 ff 8b 48 58 d8 01 11 45	H.\\$.H.]\$.H.t\$.H . \$ ATAUAVH.!H. yPH.YXL.j.E.&H. H.IH;IV.H.YH.B.+ I;I.GIE3I.J.T.A.I IE.DW.Z.HI.B.H.A H.IT.IDD.H.T I.Fh.EL.J.L.HH. TI.4 YV.L.H.IH. FH;IUAH.~@H9FX U.H.~X.E.^X+I;I .GI.J.T.A.WIE.DW.].+H.E.H.F`H.IT.		^
0x00007FF9D7 0x00007FF9D7	4B10E0 4B10F0 4B1100	4c 8 56 0	5 db	74 4c 8b	14 03	4c eb	8b 48	c3 03	48 fb	8b 48 70	d7 8b	49 5c	8b 24 8b	40 74	e8 4c	0f 89	L t.L. H. I. . VL.+H.[H.\\$@L.		
																		Close	

4.1.15 FPO Data

The FPO display shows the all the Frame Pointer Omission data in the PE file.

Frame Pointer Omission data is largely obsolete these days. Since Windows Vista, no Windows binaries have been shipped that use Frame Pointer Omission. However the option to use Frame Pointer Omission remains in the compiler.

andar										
File Header	Туре		Size	Locals	Params	Prolog	Registers	SEH	Uses EBP	
Optional Header	FPO	0x0002c719	13	0	0	0	0	No	No	
Optional Header Directories	FPO	0x00028cbd	35	0	4	0	0	No	No	
Additional Information	FPO	0x00028ce0	5	0	0	0	0	No	No	
.Net Information	FRO	0x00028co5	1	0	0	0	0	No	No	
Net Header	110	0,00020005		0	0	0	0	110	NO	
dules	FPO	0x00089atc	25	0	0	0	0	No	No	
Imported Modules and Functions	FPO	0x0002914b	104	0	0	0	2	No	No	
Delay Loaded Modules	FPO	0x000291b3	63	0	8	0	0	No	No	
All Dependent Modules	FPO	0x000291f2	7	0	0	0	0	No	No	
nctions	FPO	0x000291f9	23	0	4	0	0	No	No	
Imported Functions	FRO	0~00020210	2	0		0	0	No	No	
a	100	0,00023210	5	0	-	0	0	NU NU	NO	
Sections / Segments	FPO	0x00029210	3	0	4	0	0	No	No	
Data Bounds	FPO	0x00029210	3	0	4	0	0	No	No	
FPO Data	FPO	0x00029210	3	0	4	0	0	No	No	
Version Information	FPO	0x00029210	3	0	4	0	0	No	No	
Manifest Digital Signatures	FRO	0x00029210	3	0	4	0	0	No	No	
Thread Local Storage	500	0.00020210	2	0	-	0	•	NI-	No	
Resources	FPO	0x00029210	3	U	4	U	0	No	No	
mbols	FPO	0x00029210	3	0	4	0	0	No	No	

Туре

The type of frame. Valid values are:

- FPO
- Trap
- TSS
- Non FPO

RVA

The relative virtual offset of the function that has Frame Pointer Omission data

Size

This is the size of the function that has Frame Pointer Omission data.

Locals

The number of DWORD local variables this function has.

Params

The number of DWORD parameters this function has.

Prolog

The number of bytes in the function prolog.

Registers

How many registers are saved.

SEH

Does the function use Structured Exception Handling?

Uses EBP

Does the function use the EBP register?

Sorting

The data can be sorted by any column. Click the column header to select sorting by that column. Click the same header to reverse the sort direction.

4.1.16 x64 Exception Handling

The x64 Exception Handling display shows the all the exception handling data in the x64 PE file. These are RUNTIME_FUNCTION records.

x86 PE files handle exceptions using a different technique, as such there is no corresponding data in an x86 PE file. This display is not available when working with a 32 bit (x86) PE file.

addr * Function Filename Line Begin End Unwind Version File Header Optional Header Optional Header 0 0x00001030 0x00001060 0x00001060 0x00001060 0x00001060 0x00001060 0x00001060 0x00001060 0x00001060 0x00001070 0x00001070 0x00001070 0x00001070 0x00001070 0x00001070 0x000001100 0x00001100 0x000001100 0x000001100 0x00										
Implementation Imple	ader .	^ #	Function	Filename	Line	Begin	End	Unwind	Version	Fla
Optional Header Dependent Header	Optional Header	🖽 🕨 o	CFont::'scalar deleting de			0x00001030	0x0000106a	0x0009ec44	1	0x0
Additional Information C. GdiObject::scalar deleti C. GdiObject::scalar deleti C. SdiObject::scalar deleti Sdi Scalar deleti Sdi Sc	Optional Header Directories	□ □ □ ▶ 1	CBrush: 'scalar deleting d			0x00001080	0x000010ba	0x0009ec44	1	0x0
Net Information Image: 2 Collidiglet::schart defetime 0x0000100 0x0000100 0x0000100 0x0000100 0x0000100 0x0000100 0x0000100 0x0000100 0x00001100 0x00001100 0x00001100 0x00001100 0x000001130 0x000001130 0x000001130 0x000001130 0x000001140 0x000001140 0x000001140 0x000001140 0x00001140 0x00001460 0x00001460 0x00001460 0x000001460 0x00001460 0x0000	Additional Information		CGdiObiestu'scalas deleti			0+000010=0	0-00001060	0:0000==44	1	0.0
Net Header Image: Solution of the solution of th	Net Information		Codiobject: scalar deleti			000001000	0x00001010	0x0009ec44		UX
Net MetaData Image: scalar deleting 0x00001140 0x0000117a 0x0000ec44 1 Jes monted Modules and Functions Image: scalar deleting e\orm\c\3rd_src\virtualtreegridcon 41 0x00001140 0x0000117a 0x0000ec44 1 UB Dependent Modules Image: scalar deleting e\orm\c\3rd_src\virtualtreegridcon 42 0x00001400 0x0000142 0x00001460 1 1 UB Dependent Modules Image: scalar deleting e\orm\c\3rd_src\virtualtreegridcon 42 0x00001400 0x0000142 0x00001560 0x00002544 1 1	Net Header	🖽 🕨 3	CGdiObject::~CGdiObject	c:\program files (x86)\microsoft vi	77	0x000010f0	0x0000113b	0x0009efac	1	0x
defs monted Modules and Functions > > S CGridCell::Create.Object e\om\c\3rd_src\virtualtreegridcon 41 0x00001410 0x0000148c 0x0000146c 0x0000146c 1 W Dependent Modules > > 6 CGridDefaultCell::Create e\om\c\3rd_src\virtualtreegridcon 42 0x00001400 0x0000146c 0x0000146c 0x0000146c 0x0000146c 0x0000146c 0x0000150c 0x0000156c 0x0000150c	Net MetaData	🖽 🕨 4	CBitmap::`scalar deleting			0x00001140	0x0000117a	0x0009ec44	1	0x
Imported Proclems and Proclems 	ules	🖽 🕨 5	CGridCell::CreateObject	e:\om\c\3rd_src\virtualtreegridcon	41	0x00001410	0x0000148c	0x0009ec44	1	0x
Image: Dependent Modules Image: Critical End	relay Loaded Modules		CGridDefaultCell::Create	e:\om\c\3rd_src\virtualtreegridcon	42	0x00001490	0x000014c4	0x000a1f60	1	0x
ions CGridCell::SetFont e:\omegaTedFunctions 0x00001560 0	Dependent Modules		CGridCell::CGridCell	e:\om\c\3rd src\virtualtreenridcon	48	0x000014f0	0x0000154e	0x0009ec44	1	Ov
mported Functions % Conducting scalar detertime monted Functions monted Functions <td>tions</td> <td></td> <td></td> <td>ertonntetora_sretontaatareegnacona.</td> <td></td> <td>0.00001550</td> <td>0.00001540</td> <td>0.0000.44</td> <td></td> <td></td>	tions			ertonntetora_sretontaatareegnacona.		0.00001550	0.00001540	0.0000.44		
Segments 9 CGridCell:-CGridCell e\omega CGridCell:-CGridCell e\omega CGridCell:-CGridCell e\omega CGridCell:-CGridCell e\omega CGridCell:-CGridCell e\omega CGridCell:-CGridCell e\omega CGridCell:-CGridCell CGridCell:-CGridCell:-CGridCell CGridCell:-CGridC	mported Functions		CGridCell:: scalar deletin			0x00001550	0x000015ac	0x0009ec44	1	0x
Segments Image: Constraint of the sectors / Segments<	Exported Functions	🖽 🕨 9	CGridCell::~CGridCell	e:\om\c\3rd_src\virtualtreegridcon	55	0x000015b0	0x000015ef	0x000a23b8	1	0x
Segments Image: Construction of the section of the		🖽 🕨 10	CGridCell::Reset	e:\om\c\3rd_src\virtualtreegridcon	71	0x000015f0	0x00001662	0x0009ec44	1	0x
Big Data Image: Display the second secon	ata Bounds	🖽 🕨 11	CGridCell::SetFont	e:\om\c\3rd_src\virtualtreegridcon	91	0x00001670	0x00001719	0x0009c24c	1	0x
664 Exception Handling Image: Second Se	PO Data	⊞ ▶ 12	CGridCell::GetFont	e:\om\c\3rd src\virtualtreegridcon	121	0x00001720	0x00001751	0x0009c414	1	0x
ersion Information anifest anifest b 14 CGridCell::GetFormat er\om\c\3rd_src\virtualtreegridcon 175 0x000017e0 0x00001810 0x0009c414 1 b 15 CGridCell::GetMargin er\om\c\3rd_src\virtualtreegridcon 188 0x00001810 0x00001840 0x0009c414 1	64 Exception Handling	⊞ ▶ 13	CGridCell::GetFont	e:\om\c\3rd_src\virtualtreegridcon	135	0×00001760	0x000017a9	0x000a23b8	1	0
allitest gital Signatures igital Signatures igital Signatures CGridCell::GetMargin e\om\c\3rd_src\virtualtreegridcon 188 0x0001810 0x00001840 0x00002414 1	ersion Information		CGridCelluGetEormat	e\om\c\?rd_src\virtualtreegrid.con	175	0×000017=0	0~00001810	0x0000c414	1	0
Stand up to the stand up to t	nital Signatures		condem.oet offiat	e. (on (c (ord_sic (virtualitieegnacon	175	000001760	0,00001010	0,00050414		
	hread Local Storage	□ □ ▶ 15	CGridCell::GetMargin	e:\om\c\3rd_src\virtualtreegridcon	188	0x00001810	0x00001840	0x0009c414	1	0x

Each line of the grid displays the following information. If you expand the entry you can see the codes used for that exception handler.

#

The index of the RUNTIME_FUNCTION in the PE file.

Function

The name of the function this RUNTIME_FUNCTION relates to This value is only populated if debug information can be read for this PE file.

Filename

The filename of the function this RUNTIME_FUNCTION relates to *This value is only populated if debug information can be read for this PE file.*

Line

The line number of the function this RUNTIME_FUNCTION relates to *This value is only populated if debug information can be read for this PE file.*

Begin

The relative virtual address of the start of the area affected by the exception handler.

End

The relative virtual address of the end of the area affected by the exception handler.

Unwind

The relative virtual address of the unwind data for the exception handler.

Version

The version format of the exception handling data

Flags

Flags that indicate how what this exception handler is. Valid values are:

- Exception Handler
- Termination Handler
- Chained unwind info

Prolog

Size of the function prologue.

Frame Reg

The name of frame register.

Frame Offset

The offset for the frame.

Codes

The number of codes used by this exception handler.

Except RVA

The relative virtual address of the exception handler.

4.1.17 Version Information

The Version Information display shows the all the version information strings in the PE file.

File Software Updates Help Header Amme Value Product name Product name PE File Browser Optional Header Company name Software Verify Limited - www.softwareverify.com Additional Information Legal convright Convright © 2003-2023 Software Verify Limited
Header Amme Value - Optional Header Product name PE File Browser - Optional Header Company name Software Verify Limited - www.softwareverify.com - Additional Information Legal convright Convright © 2003-2023 Software Verify Limited
File Header Product name PF File Browser Optional Header Optional Header Company name Software Verify Limited - www.softwareverify.com Additional Information Iterat Convright © 2003-2023 Software Verify Limited
Optional Header Directories Company name Software Verify Limited - www.softwareverify.com Additional Information List 25 comparison List 25 comparison List 25 comparison
Additional Information
- Net Header File description PE File Browser
Modules File version 1, 95, 0, 0
Participated Modules and Functions Product version 1, 95, 0, 0
All Dependent Modules Original filename peFileBrowser.exe
Functions Internal name peFileBrowser.exe
Experted Functions
Data
- Data Bounds
- FPO Data
- x64 Exception Handling
Manifest
- Digital Signatures
- Thread Local Storage
Resources
Symbols V

The following version strings are queried to see if they are present in the PE file. Only values that are present are displayed.

- Product name
- Company name
- Legal copyright
- Legal Trademarks
- File description
- File version
- Product version
- Original filename
- Internal name
- Private build
- Special build
- Comments

4.1.18 Manifest

The Manifest display shows the manifest that is used by the PE file.



The manifest will be reformatted to make it more readable if required. Embedded manifests have no formatting and are thus quite hard to read.

4.1.19 Digital Signatures

The Digital Signatures display shows the digital signature information in the PE file.

🐺 PE File Browser x64 [C:\Program I	Files (x	86)\Software Verify\Memo	y Validator x64 Evaluation\memoryValidator_x64.exe]	-		×
File Software Updates Help						
Header	^	Name	Value		_	^
Optional Header		Display Name	Sectigo RSA Code Signing CA			
Optional Header Directories		Organisation Unit				
.Net Information		Organisation	Sectigo Limited			
		Street				
Modules		State/Province	Greater Manchester			
- Imported Modules and Functions		Locality	Salford			
All Dependent Modules		Country	GB			
Functions Imported Eurotions		Email				
Exported Functions						
Data Sections / Segments		Display Name	Software Verify Limited			
Data Bounds		Organisation Unit				
FPO Data x64 Exception Handling		Organisation	Software Verify Limited			
Version Information		Street	Eldo House			
Digital Signatures		State/Province	Suffolk			
Thread Local Storage		Locality	Runz Saint Edmunds			~
Symbols	~	<			>	

The digital signature is queried for the following information. Any queries that provide a result are displayed here.

- Display Name
- Organisation Unit
- Organisation
- Street
- State/Province
- Locality
- Country
- Email

4.1.20 Thread Local Storage

The Thread Local Storage display shows the thread local storage in the PE file.

der 🔨	Name	Value	Comment		
File Header Ontional Header	StartAddressOfRawData	0x00007ff667545000	.tls		
Optional Header Directories	EndAddressOfRawData	0x00007ff667545001	.tls		
Additional Information	SizeOfZeroFill	0x00000000			
Net Information	Characteristics	0x00000000			
.Net MetaData	AddressOfIndex	0x00007ff667543020	.data		
ules	AddressOfCallBacks	0x00007ff6675421a0	.rdata		
Imported Modules and Functions	Callback 1	0x00007ff667541000	.text		
All Dependent Modules	Callback 2	0x00007ff667541000	.text		
ctions	Callback 3	0×00000000000000000			
Imported Functions					
Sections / Segments					
Data Bounds					
rPO Data					
Version Information					
Manifest					
Digital Signatures					
Thread Local Storage					

Copyright © 2015-2025 Software Verify Limited

The thread local storage is queried for the following information. Any queries that provide a result are displayed here.

- StartAddressOfRawData
- EndAddressOfRawData
- SizeOfZeroFill
- Characteristics
- AddressOfIndex
- AddressOfCallBacks
- Callback 1
- Callback 2
- Callback 3

4.1.21 Resources

The Resources display shows the resources section in the PE file.

💐 PE File Browser x64 [E:\om\c\memory	2\tabserv\release_x64\memoryValidate	pr_x64.exe]	-	×
File Software Updates Help				
PE File Browser x64 [E\com\c\memory: File Software Updates Help Header ^	I2\tabser\release_x64\memoryValidato	Value 0x0000000 0x0000000 4 0 1 11		×
- FPO Data - x64 Exception Handling - Version Information - Manifest - Digital Signatures - Thread Local Storage - Resources				
< >>				

The resources header is queried for the following information. Any queries that provide a result are displayed here.

- Characteristics
- TimeStamp
- MajorVersion
- MinorVersion
- NumberOfNamedEntries
- NumberOfldEntries

4.1.22 Debug Information

The Debug Information display shows the CV record for the module plus the symbol server id and local symbol store id.

👯 PE File Browser x64 [E:\om\c\p	eFileBro	owser\Release\x64\peFileE	3rowser_x64.exe]	-	×
File Software Updates Help					
	^	Name	Value		
····· .Net MetaData Modules		GUID	24783792-d2da-45ce-a65e-b15b4054f626		ĺ
Imported Modules and Functions		Age	3		
All Dependent Modules		FileName	e:\om\c\peFileBrowser\Release\x64\peFileBrowser_x64.pdb		
Functions Imported Eurotions		Symbol Server ID	24783792D2DA45CEA65EB15B4054F6263		
Exported Functions		Symbol Store ID	64A6D80714A000		
Sections / Segments Data Bounds FPO Data vs64 Exception Handling Version Information Manifest Digital Signatures Thread Local Storage Resources Symbols Debug Information PDB Symbols COFF Symbols COFF Symbols COFF Symbols STABS Symbols Misc Debug Data	×				
1	>				

If the module contains a CV record the data in that record is shown here.

GUID	The globally unique identifier that identifies the module.
Age	The age of the module - this is the number of times the module has been linked after a clean or full rebuild.
Filename	The name of the PDB file at the time the module was built.
Symbol server id	The id that identifies a PDB file.
Symbol store id	The id that identifies a module.

4.1.23 PDB Symbols

The PDB Symbols display shows the all the PDB format symbols for the PE file.

.Net Header	Filter:	Filter		E	xport
lules	#	Name	🛆 Call Conv	Address	Size
Imported Modules and Functions	0	\$S1		0x00007FF6BECA8778	4 (0)
All Dependent Modules	1	\$S1		0x00007FF6BECA8FD0	4 (0)
ctions	2	Sxdatasym		0x00007FF6BEC8ABE8	1 (0)
Imported Functions	3	Sxdatasym		0x00007FF6BEC8AC88	1 (0)
a	4	AFX EXCEPTION LINK::AFX EXCEPTION LINK		0x00007FF6BEC54912	0 (0
Sections / Segments	5	AFX EXCEPTION LINK::~AFX EXCEPTION LINK	CV CALL NEAR C	0x00007FF6BEBF5080	7 (0
Data Bounds	6	ATI :: AtiThrowing	CV CALL NEAR C	0x00007EE6BEC04C70	26 (
x64 Exception Handling	7	ATI "Clmage 'RTI Type Descriptor'		0x00007EE6BECA50C0	0.00
Version Information	8	ATL Clmage: AlphaBlend	CV CALL NEAR C	0x00007EE6BEC0D870	314
Manifest	0	ATLuCimageuCimage		0x00007EE68EC0DD20	101
Digital Signatures	9	ATERCIMAGESCIMAGE	CV_CALL_NEAR_C	0x00007FF0BEC0DD30	101
Resources	10	ATL::CImage::CInit@DIPlus::Init	CV_CALL_NEAR_C	0x0000/FF6BECTICF0	137
bols	11	ATL::CImage::CreateEx	CV_CALL_NEAR_C	0x00007FF6BEC0DB00	339
Debug Information	12	ATL::CImage::CreateFromGdiplusBitmap	CV_CALL_NEAR_C	0x00007FF6BEC11F80	105
PDB Symbols	13	ATL::CImage::Load	CV_CALL_NEAR_C	0x00007FF6BEC11C50	147
COFF Symbols	14	ATL::CImage::SetColorTable	CV CALL NEAR C	0x00007FF6BEC11E10	199
TADO Cumbala					

Columns

#

The ordinal of the symbol in the symbol data.

Name

The name of the symbol.

Call Conv

The calling convention for the symbol.

Address

The symbol address.

Size

The symbol size.

Туре

The type of debug data. Valid values are:

- SymTagNull
- SymTagExe
- SymTagCompiland
- SymTagCompilandDetails
- SymTagCompilandEnv
- SymTagFunction
- SymTagBlock
- SymTagData
- SymTagAnnotation
- SymTagLabel
- SymTagPublicSymbol
- SymTagUDT
- SymTagEnum
- SymTagFunctionType
- SymTagPointerType
- SymTagArrayType
- SymTagBaseType
- SymTagTypedef
- SymTagBaseClass
- SymTagFriend
- SymTagFunctionArgType
- SymTagFuncDebugStart
- SymTagFuncDebugEnd
- SymTagUsingNamespace
- SymTagVTableShape
- SymTagVTable
- SymTagCustom
- SymTagThunk
- SymTagCustomType
- SymTagManagedType
- SymTagDimension

Flags

Any flags that related to this debug information

Line Number

The line number for the symbol, if any.

Filename

The filename for the symbol, if any.

Sorting

The data can be sorted by any column. Click the column header to select sorting by that column. Click the same header to reverse the sort direction.

Filtering

To restrict the display to just symbols that match the filter, or symbols with a filename that matches the filter, enter your filter specification into the filter field then click **Filter**.

The filter can include the * wildcard to match any sequence of characters.

Example filters:

Get* *test* *.cpp

To reset the filter, delete all text from the filter field then click Filter.

Exporting Data

To export the contents of the grid to a comma separated text file, click the **Export...** button then enter the name of the file you would like the data saved in.

4.1.24 COFF Symbols

The COFF Symbols display shows the all the COFF format symbols for the PE file.

💐 PE File Browser x86 [E:\om\bugs	\a_deb	ug_formats\COFF\Ben\	/anRens\memoryValidatorTest\a-gco	off.exe]					
File Software Updates Help									
Net Information Net Header	^	Filter:	Filter						Export
.Net MetaData		Name		Section	Туре	Storage	Size	Lines	Filen ^
Modules		.bb	0x000002fa	Section 1	void	Block	0 (0x0000)		main.c
- Delay Loaded Modules		.bb	0x00000336	Section 1	void	Block	0 (0x0000)		functio
All Dependent Modules		.bf	0x000002d0	Section 1	void	Function	0 (0x0000)		main.«
Functions		.bf	0x00000330	Section 1	void	Function	0 (0x0000)		functio
Exported Functions		.bss	0x0000008	Section 4	int	Static	0 (0x0000)		new_o
Data		.bss	0x0000008	Section 4	int	Static	0 (0x0000)		eh_pei
- Sections / Segments		.bss	0x000040a8	Section 4	int	Static	0 (0x0000)		crtstuf
- FPO Data		.bss	0x000040a8	Section 4	int	Static	0 (0x0000)		pseud
Version Information		.bss	0x000040a8	Section 4	int	Static	0 (0x0000)		crtstuf
- Digital Signatures		.bss	0x000040a8	Section 4	int	Static	0 (0x0000)		fake
Thread Local Storage		.bss	0x000040a8	Section 4	int	Static	0 (0x0000)		fake
Resources		.bss	0x000040a8	Section 4	int	Static	0 (0x0000)		crtstuf
- Debug Information		.bss	0x000040a8	Section 4	int	Static	0 (0x0000)		pseud
PDB Symbols		.bss	0x00000000	Section 4	int	Static	0 (0x0000)		crt1.c
COFF Symbols		bss	0x000040a8	Section 4	int	Static	0 (0x0000)		pseud
STABS Symbols		bss	0x000040a8	Section 4	int	Static	0 (0x0000)		fake 🗸
Misc Debug Data	*	<					(>
< 2	•								

Columns

Copyright © 2015-2025 Software Verify Limited

Name

The name of the symbol.

Address

The relative virtual address of the symbol.

Section

The symbol section number.

Type The datatype for the symbol.

Storage The storage class for the symbol.

Size The size of the symbol.

Lines Number of lines for this function (blank if not available)

Filename

The symbol filename.

Sorting

The data can be sorted by any column. Click the column header to select sorting by that column. Click the same header to reverse the sort direction.

Filtering

To restrict the display to just symbols that match the filter, or symbols with a filename that matches the filter, enter your filter specification into the filter field then click **Filter**.

The filter can include the * wildcard to match any sequence of characters.

Example filters:

Get* *test* *.cpp

To reset the filter, delete all text from the filter field then click Filter.

Exporting Data

To export the contents of the grid to a comma separated text file, click the **Export...** button then enter the name of the file you would like the data saved in.

4.1.25 CodeView Symbols

The CodeView Symbols display shows the all the CodeView format symbols for the PE file.

ile Software Updates Help							
	Filter:	Filter					Expor
.Net Header				011	1. N. I		
Modules	Name	Пуре	Address	Offset	Line Number	Filename	
		Line Offset	0x0000000	0x000077a3	279	file_io.c	
- Delay Loaded Modules		Data	0xfd5a05b9	0x0000000	0	math_x87.obj	
- All Dependent Modules		Data	0x65c1ef1f	0x0000000	0	math_x87.obj	
Functions		Data	0x65c1ef1f	0x00000000	0	math x87.obj	
Exported Functions		Data	0x65c1ef1f	0×0000000	0	locale.obi	
Data		Data	0v65c1cf1f	0~0000000	0	stateld obj	
- Sections / Segments		Data		000000000	0	sittoid.obj	
Data Bounds		Data	0x65c1ef1f	0x0000000	0	math_x87.obj	
FPO Data		Data	0x7ce36145	0x0000000	0	wcstold.obj	
Version Information		Data	0x7ce36145	0x00000000	0	strtold.obj	
- Digital Signatures		Data	0x628bbaf4	0x00000000	0	strtold.obj	
Thread Local Storage		Data	0x628bbaf4	0x00000000	0	ThreadLocalData.obi	
Resources		Data	0x99573ab6	0×0000000	0	locale obi	
Symbols		Data	0.145011.40	0.00000000	0	lo cale obj	
DDR Symbole		Data	Uxbf30bbd0	0x0000000	U	locale.obj	
COFE Symbols		Data	0xc8e726b7	0x0000000	0	locale.obj	
CodeView Symbols		Data	0x9a5be5b7	0x0000000	0	locale.obj	
- STABS Symbols		Data	0x9b9a45b7	0x0000000	0	locale.obi	

Columns

Name

The name of the symbol.

Туре

The type of the symbol.

Address

The relative virtual address of the symbol.

Offset

The symbol offset.

Line Number

The line number, if any, of the symbol.

Filename

The filename of the symbol.

Sorting

The data can be sorted by any column. Click the column header to select sorting by that column. Click the same header to reverse the sort direction.

Filtering

To restrict the display to just symbols that match the filter, or symbols with a filename that matches the filter, enter your filter specification into the filter field then click **Filter**.

The filter can include the * wildcard to match any sequence of characters.

Example filters: Get* *test* *.cpp

To reset the filter, delete all text from the filter field then click Filter.

Exporting Data

To export the contents of the grid to a comma separated text file, click the **Export...** button then enter the name of the file you would like the data saved in.

4.1.26 STABS Symbols

The STABS Symbols display shows the all the STABS format symbols for the PE file.

👯 PE File Browser x86 [E:\om\bugs	s\BenR	endel\genie\genie.exe]						- 0
ile Software Updates Help								
Net Information 	^	Filter:	Filter					Export.
		Name		Section	Туре	Storage	Size	Filename ^
Modules		, BinaryToGrayn	0x000468f0	Section 8	struct	External	12 (0x000c)	d:\ANTLR\VER-MAIN
- Delay Loaded Modules		, _ioinitc	0x0031f0d4	Section 8	struct	External	0 (0x0000)	d:\ANTLR\VER-MAIN-
All Dependent Modules		, _ioinite	0x0031f5f0	Section 8	struct	External	0 (0x0000)	
unctions		,_ioinite	0x0031f1a8	Section 8	struct	External	0 (0x0000)	
Exported Functions		,_ioinite	0x0031f020	Section 8	struct	External	0 (0x0000)	
ata		, _ioinite	0x0031f328	Section 8	struct	External	0 (0x0000)	
Sections / Segments		, _ioinite	0x0031f460	Section 8	struct	External	0 (0x0000)	
- FPO Data		, ioinite	0x0031f4c9	Section 8	struct	External	0 (0x0000)	
Version Information		, ioinite	0x0031f06c	Section 8	struct	External	0 (0x0000)	
Manifest Digital Signatures		ioinite	0x0031f494	Section 8	struct	External	0 (0x0000)	
- Thread Local Storage		ioinite	0x0031f42c	Section 8	struct	External	0 (0x0000)	
Resources		ioinite	0x0031f4fd	Section 8	struct	External	0 (0x0000)	
ymbols Debug Information		ioinite	0x0031f30c	Section 8	struct	External	0 (0×0000)	
- PDB Symbols		, _ioinite	0.00316565	Section 0	struct	External	0 (0,0000)	
COFF Symbols		,_ionite	0x00311505	Section 8	struct	External	0 (0x0000)	
- CodeView Symbols		, _loinite	UXUU311318	Section 8	struct	External	U (UXUUUU)	~
Misc Debug Data		<						>
<	>							

Columns

Name

The name of the symbol.

Address

The relative virtual address of the symbol.

Section

The symbol section number.

Туре

The datatype for the symbol.

Storage

The storage class for the symbol.

Size

The size of the symbol.

Filename

The symbol filename.

Sorting

The data can be sorted by any column. Click the column header to select sorting by that column. Click the same header to reverse the sort direction.

Filtering

To restrict the display to just symbols that match the filter, or symbols with a filename that matches the filter, enter your filter specification into the filter field then click **Filter**.

The filter can include the * wildcard to match any sequence of characters.

Example filters:

Get* *test* *.cpp

To reset the filter, delete all text from the filter field then click Filter.

Exporting Data

To export the contents of the grid to a comma separated text file, click the **Export...** button then enter the name of the file you would like the data saved in.

4.1.27 Misc Debug Data

The Misc Debug Data display shows the all the Miscellaneous debug symbols for the PE file.

Γ	👯 PE File Browser x86 [C:\Program I	Files (x	x86)\Microsoft Visual Studio 8\VC\ce\bin\x86_arm\midl.exe]		×
	File Software Updates Help				
	File Software Updates Help - Net Information Net Meder Imported Modules and Functions - Delay Loaded Modules - All Dependent Modules Functions - Exported Functions - Exported Functions Data - Deta Bounds - Po Data - Version Information - Manifest - Digital Signatures - Thread Local Storage - Resources	^	Name exe\midl.dbg		
	Symbols Debug Information				
		~	٢	>	
	< >>				

4.2 View Memory Dialog

The Inspect Memory dialog is shown below.

47 DbgHelp Browser Help



Offset > an offset inside the DLL to start viewing memory

Size > how many bytes to view

Specify an offset inside the DLL and the number of bytes to view.

If the memory is readable, the memory will be displayed.

If the memory is executable, a disassembly view is also availble.

Memory View, 0x00007FF9D74B7000 41523 bytes							
BYTE WORD	DWORD QWORD Disassembly						
0x00007FF9D74B7000 58 a1 0x00007FF9D74B7010 3a 9e 0x00007FF9D74B7020 64 9e 0x00007FF9D74B7030 86 9e 0x00007FF9D74B7040 a8 9e 0x00007FF9D74B7050 c2 9e 0x00007FF9D74B7060 f0 9e 0x00007FF9D74B7070 18 9f 0x00007FF9D74B7090 16 a1 0x00007FF9D74B7090 16 a1 0x00007FF9D74B7090 16 a1 0x00007FF9D74B7080 88 a1 0x00007FF9D74B7000 a0 a0 0x00007FF9D74B70D0 88 a0 0x00007FF9D74B70D0 88 a0 0x00007FF9D74B70D0 88 a0 0x00007FF9D74B70D0 88 a0	00 00 <td< td=""><td>^</td></td<>	^					
		Close					

4.3 Search Memory Dialog

The Search Memory dialog is shown below.

Search Mem	iory		? ×				
Search Search fo	Clear or a text string:		Close]			
EurekaLog Match case Unicode Search for bytes:							
ID (84)	Address	Section	Description 1				
66	0x00473EC2	CODE	EurekaLog 6.0.23 trial				
67	0x00473F2E	CODE	EurekaLog full version				
68	0x00473F57	CODE	eurekalog.com				
69	0x004B80DF	.edata	EurekaLog_AttachedF				
70	0x004B8103	.edata	EurekaLog_CallCreate				
71	0x004B811E	.edata	EurekaLog_CallExcept				
۲٦ <	0-00400120		Frankel and Collecter >	1			

You can search for text strings or you can search for byte sequences.

Search for a text string > type the string you wish to search for into the text box

Match case > select the check box if the string match should be case sensitive

Unicode > select the check box if the string match should be Unicode. If the check box is not selected the string match is ANSI

Search for bytes > type the bytes you wish to search for into the text box. A byte should be specified as a two digit hex value. Separate bytes with spaces

Search > perform the search. The progress of the search is shown on the progress bar, any matching search results are shown in the list.

Clear > clear the search results

A context menu on the search results provides a single option:

View data...

Clicking **View data...** opens a memory inspection dialog, allowing you to view the search results memory as BYTEs, WORDs, DWORDs or QWORDs.

Search Memory, 0x00473F57 9 bytes	?	×
BYTE WORD DWORD QWORD Disassembly 0x00473F10 74 73 20 63 6f 6d 70 69 6c 61 74 69 6f 6e 2e 0d ts compilation. 0x00473F20 0a 0d 0a 54 6f 20 62 75 79 20 74 68 65 20 45 75 To buy the Eu 0x00473F40 73 69 6f 62 06 75 62 66 75 66 67 compilation. To buy the Eu 0x00473F50 3a 2f 2f 77 77 2e 55 75 72 26 66 67 compilation. To buy the Eu 0x00473F50 2f ff 77 77 2e 55 75 72 26 67 67 compilation. RaiseExc 0x00473	?	×
	Clos	e



5 Command Line Interface

PE File Browser can be used from the command line as well as with the GUI.

The command line options allow you to launch PE File Browser from another an application so that you can display PE file internals for a specific DLL.

5.1 Alphabetic Reference

/dll

Specifies the PE File that will be displayed

/dll path-to-executable

Example: /dll e:\om\c\test\release\test.dll

/exit

Causes PE File Browser to exit after exporting modules

/exportModules

Specifies the filename to export the list of dependent modules to

/exportModules path-to-export-filename

Example: /exportModules e:\om\export\dependentModules.txt

/exportModulesAndFunctions

Specifies the filename to export the list of dependent modules and functions to

/exportModulesAndFunctions path-to-export-filename

Example: /exportModulesAndFunctions e:\om\export\dependentModules.txt

/exportModulesTopLevel

Specifies the filename to export the list of dependent modules to. Only the immediate dependent modules are exported. No descendant dependencies are exported.

/exportModulesTopLevel path-to-export-filename

Example: /exportModulesTopLevel e:\om\export\dependentModules.txt

5.2 Usage Reference

Displaying a DLL

/dll

Specifies the PE File that will be displayed

/dll path-to-executable

Example: /dll e:\om\c\test\release\test.dll

Export the modules that are dependent on the DLL

Specifies the filename to export the list of dependent modules to

/exportModulesTopLevel path-to-export-filename

Example: /exportModulesTopLevel e:\om\export\dependentModules.txt

Export the modules (and their dependent modules) that are dependent on the DLL

Specifies the filename to export the list of dependent modules to

/exportModules path-to-export-filename

Example: /exportModules e:\om\export\dependentModules.txt

Export the modules (and their dependent modules) and functions that are dependent on the DLL

Specifies the filename to export the list of dependent modules and functions to

/exportModulesAndFunctions path-to-export-filename

Example: /exportModulesAndFunctions e:\om\export\dependentModules.txt

Close PE File Browser after exporting

/exit

