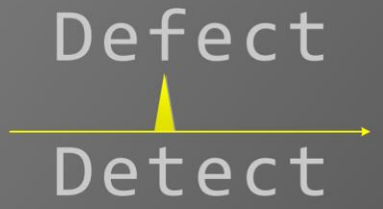

Windows Memory Dump Analysis Accelerated

Version 5.7

Part 1: Process User Space

Dmitry Vostokov
Software Diagnostics Services

Published by OpenTask, Republic of Ireland

OpenTask books and magazines are available through booksellers and distributors worldwide. For further information or comments, send requests to press@opentask.com.

A CIP catalog record for this book is available from the British Library.

ISBN-l3: 978-1-912636-97-6 (Paperback)

Revision 5.70 (September 2022)

Contents

Exercise P1: Analysis of a normal application process dump (64-bit notepad)

Goal: Learn how to see dump file type and version, get a stack trace, check its correctness, perform default analysis, list threads and modules, check module version information, dump module data, and check process environment.

Patterns: Manual Dump (Process); Stack Trace; Not My Version (Software); Environment Hint; Unknown Component.

1. Launch WinDbg Preview.

2. Open \AWMDA-Dumps\Process\x64\notepad.DMP.

3. We get the dump file loaded:

```
Microsoft (R) Windows Debugger Version 10.0.25136.1001 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.


Loading Dump File [C:\AWMDA-Dumps\Process\x64\notepad.DMP]
User Mini Dump File with Full Memory: Only application data is available


************* Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       srv*
Symbol search path is: srv*
Executable search path is:
Windows 10 Version 18362 MP (2 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
Edition build lab: 18362.1.amd64fre.19h1_release.190318-1202
Machine Name:
Debug session time: Thu Sep  5 07:37:05.000 2019 (UTC + 1:00)
System Uptime: 0 days 0:03:43.584
Process Uptime: 0 days 0:02:07.000
.....................................
For analysis of this file, run !analyze -v
win32u!NtUserGetMessage+0x14:
00007ffe`dad31164 ret
```

4. Open a log file to save all future output using **.logopen** command:

```
0:000> .logopen C:\AWMDA-Dumps\Process\x64\notepad.log
Opened log file 'C:\AWMDA-Dumps\Process\x64\notepad.log'
```

5. Type **k** command to verify the correctness of the stack trace:

```
0:000> k
# Child-SP          RetAddr               Call Site
00 000000a1`c2ccf988 00007ffe`dc19477d     win32u!NtUserGetMessage+0x14
01 000000a1`c2ccf990 00007ff7`437da3d3     user32!GetMessageW+0x2d
02 000000a1`c2ccf9f0 00007ff7`437f02b7     notepad!WinMain+0x293
03 000000a1`c2ccfac0 00007ffe`dc557bd4     notepad!__mainCRTStartup+0x19f
04 000000a1`c2ccfb80 00007ffe`ddc6cee1     kernel32!BaseThreadInitThunk+0x14
05 000000a1`c2ccfbb0 00000000`00000000     ntdll!RtlUserThreadStart+0x21
```

6. Type **version** command to get OS version, system and process uptimes, the dump file timestamp, and its type:

```
0:000> version
Windows 10 Version 18362 MP (2 procs) Free x64
Product: WinNt, suite: SingleUserTS Personal
18362.1.amd64fre.19h1_release.190318-1202
Machine Name:
Debug session time: Thu Sep  5 07:37:05.000 2019 (UTC + 1:00)
System Uptime: 0 days 0:03:43.584
Process Uptime: 0 days 0:02:07.000
  Kernel time: 0 days 0:00:00.000
  User time: 0 days 0:00:00.000
Full memory user mini dump: C:\AWMDA-Dumps\Process\x64\notepad.DMP

Microsoft (R) Windows Debugger Version 10.0.22549.1000 AMD64
Copyright (c) Microsoft Corporation. All rights reserved.

command line: '"C:\Program Files\WindowsApps\Microsoft.WinDbg_1.2202.7001.0_neutral__8wekyb3d8bbwe\amd64\EngHost.exe"
npipe:pipe=DbgX_cc00b95e98744c57b455e8713a3c8f05,password=f8fa2b555334 "C:\Program
Files\WindowsApps\Microsoft.WinDbg_1.2202.7001.0_neutral__8wekyb3d8bbwe\amd64" "C:\ProgramData\Dbg"'  Debugger Process 0x4D14
dbgeng:  image 10.0.22549.1000,
        [path: C:\Program Files\WindowsApps\Microsoft.WinDbg_1.2202.7001.0_neutral__8wekyb3d8bbwe\amd64\dbgeng.dll]
dbghelp: image 10.0.22549.1000,
        [path: C:\Program Files\WindowsApps\Microsoft.WinDbg_1.2202.7001.0_neutral__8wekyb3d8bbwe\amd64\dbghelp.dll]
        DIA version: 30795
Extension DLL search Path:
[...]
Extension DLL chain:
    dbghelp: image 10.0.22549.1000, API 10.0.6,
        [path: C:\Program Files\WindowsApps\Microsoft.WinDbg_1.2202.7001.0_neutral__8wekyb3d8bbwe\amd64\dbghelp.dll]
    exts: image 10.0.22549.1000, API 1.0.0,
        [path: C:\Program Files\WindowsApps\Microsoft.WinDbg_1.2202.7001.0_neutral__8wekyb3d8bbwe\amd64\WINXP\exts.dll]
    uext: image 10.0.22549.1000, API 1.0.0,
        [path: C:\Program Files\WindowsApps\Microsoft.WinDbg_1.2202.7001.0_neutral__8wekyb3d8bbwe\amd64\winext\uext.dll]
    ntsdexts: image 10.0.22550.1002, API 1.0.0,
        [path: C:\Program Files\WindowsApps\Microsoft.WinDbg_1.2202.7001.0_neutral__8wekyb3d8bbwe\amd64\WINXP\ntsdexts.dll]
```

Note: Debug session time is when the dump was generated. Although the dump is called a “mini dump,” it is a full memory user dump with all process memory included.

7. Type the default analysis command **!analyze -v**:

Note: This command may take some time initially as symbols may be downloaded from the symbol server:

```
0:000> !analyze -v
*******************************************************************************
*                                                                             *
*                        Exception Analysis                                   *
*                                                                             *
*******************************************************************************


KEY_VALUES_STRING: 1

    Key  : Analysis.CPU.mSec
    Value: 984

    Key  : Analysis.DebugAnalysisManager
    Value: Create

    Key  : Analysis.Elapsed.mSec
    Value: 1139

    Key  : Analysis.Init.CPU.mSec
    Value: 421

    Key  : Analysis.Init.Elapsed.mSec
    Value: 474153

    Key  : Analysis.Memory.CommitPeak.Mb
    Value: 205

    Key  : Timeline.OS.Boot.DeltaSec
    Value: 223

    Key  : Timeline.Process.Start.DeltaSec
    Value: 127

    Key  : WER.OS.Branch
    Value: 19h1_release

    Key  : WER.OS.Timestamp
    Value: 2019-03-18T12:02:00Z

    Key  : WER.OS.Version
```

```
    Value: 10.0.18362.1

    Key  : WER.Process.Version
    Value: 10.0.18362.1


FILE_IN_CAB:  notepad.DMP

NTGLOBALFLAG:  0

APPLICATION_VERIFIER_FLAGS:  0

EXCEPTION_RECORD:  (.exr -1)
ExceptionAddress: 0000000000000000
   ExceptionCode: 80000003 (Break instruction exception)
  ExceptionFlags: 00000000
NumberParameters: 0

FAULTING_THREAD:  00000750

PROCESS_NAME:  notepad.exe

ERROR_CODE: (NTSTATUS) 0x80000003 - {EXCEPTION}  Breakpoint  A breakpoint has been reached.

EXCEPTION_CODE_STR:  80000003

STACK_TEXT:
000000a1`c2ccf988 00007ffe`dc19477d     : 00000000`00007f48 00000000`0001e327 000019ee`00000000 00007ff7`00000001 : win32u!NtUserGetMessage+0x14
000000a1`c2ccf990 00007ff7`437da3d3     : 00007ff7`437d0000 00000000`0006038b 00000000`00000000 00000000`00000000 : user32!GetMessageW+0x2d
000000a1`c2ccf9f0 00007ff7`437f02b7     : 00000221`19ac3390 00000221`19ac3392 00000000`00000000 00000000`00000000 : notepad!WinMain+0x293
000000a1`c2ccfac0 00007ffe`dc557bd4     : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : notepad!__mainCRTStartup+0x19f
000000a1`c2ccfb80 00007ffe`ddc6cee1     : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : kernel32!BaseThreadInitThunk+0x14
000000a1`c2ccfbb0 00000000`00000000     : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : ntdll!RtlUserThreadStart+0x21


STACK_COMMAND:  ~0s; .ecxr ; kb

SYMBOL_NAME:  win32u!NtUserGetMessage+14

MODULE_NAME: win32u

IMAGE_NAME:  win32u.dll

FAILURE_BUCKET_ID:  BREAKPOINT_80000003_win32u.dll!NtUserGetMessage

OS_VERSION:  10.0.18362.1

BUILDLAB_STR:  19h1_release

OSPLATFORM_TYPE:  x64

OSNAME:  Windows 10

IMAGE_VERSION:  10.0.18362.329

FAILURE_ID_HASH:  {3112b5eb-303b-e877-0655-90bdfa336126}

Followup:     MachineOwner
---------
```

Note: "**Break instruction exception**" can be the sign of **Manual Dump** pattern, but often WinDbg Preview is not able to figure out an exception that may be on another thread or hidden. **STACK_COMMAND** shows the sequence commands that WinDbg Preview executed to get **STACK_TEXT**.

8. Now we check how many threads by using **~** command:

```
0:000> ~
.  0  Id: 740.750 Suspend: 0 Teb: 000000a1`c2f94000 Unfrozen
   1  Id: 740.770 Suspend: 0 Teb: 000000a1`c2f9a000 Unfrozen
   2  Id: 740.7e0 Suspend: 0 Teb: 000000a1`c2f9e000 Unfrozen
```

Note: **740** is Process ID (PID), and **750** is Thread ID (TID). **740.750** is called CID (Client ID).

9. Now we dump a stack trace using **kc** command (only modules and symbols):

```
0:000> kc
# Call Site
00 win32u!NtUserGetMessage
01 user32!GetMessageW
02 notepad!WinMain
03 notepad!__mainCRTStartup
04 kernel32!BaseThreadInitThunk
05 ntdll!RtlUserThreadStart
```

10. Now we dump the stack trace of the current thread using **k** command (with symbols, return addresses, and function offsets):

```
0:000> k
 # Child-SP          RetAddr           Call Site
00 000000a1`c2ccf988 00007ffe`dc19477d win32u!NtUserGetMessage+0x14
01 000000a1`c2ccf990 00007ff7`437da3d3 user32!GetMessageW+0x2d
02 000000a1`c2ccf9f0 00007ff7`437f02b7 notepad!WinMain+0x293
03 000000a1`c2ccfac0 00007ffe`dc557bd4 notepad!__mainCRTStartup+0x19f
04 000000a1`c2ccfb80 00007ffe`ddc6cee1 kernel32!BaseThreadInitThunk+0x14
05 000000a1`c2ccfbb0 00000000`00000000 ntdll!RtlUserThreadStart+0x21
```

Hint: How to check that the stack trace is correct. Use **ub** command (**u**nassemble **b**ackwards) to check if there is a *call* instruction. We check that *GetMessageW* function was called from *WinMain* function:

```
0:000> k
 # Child-SP          RetAddr           Call Site
00 000000a1`c2ccf988 00007ffe`dc19477d win32u!NtUserGetMessage+0x14
01 000000a1`c2ccf990 00007ff7`437da3d3 user32!GetMessageW+0x2d
02 000000a1`c2ccf9f0 00007ff7`437f02b7 notepad!WinMain+0x293
03 000000a1`c2ccfac0 00007ffe`dc557bd4 notepad!__mainCRTStartup+0x19f
04 000000a1`c2ccfb80 00007ffe`ddc6cee1 kernel32!BaseThreadInitThunk+0x14
05 000000a1`c2ccfbb0 00000000`00000000 ntdll!RtlUserThreadStart+0x21

0:000> ub 00007ff7`437da3d3
notepad!WinMain+0x271:
00007ff7`437da3b1 ff1589850100    call    qword ptr [notepad!_imp_TranslateMessage (00007ff7`437f2940)]
00007ff7`437da3b7 488d4de7        lea     rcx,[rbp-19h]
00007ff7`437da3bb ff1587850100    call    qword ptr [notepad!_imp_DispatchMessageW (00007ff7`437f2948)]
00007ff7`437da3c1 4533c9          xor     r9d,r9d
00007ff7`437da3c4 488d4de7        lea     rcx,[rbp-19h]
00007ff7`437da3c8 4533c0          xor     r8d,r8d
00007ff7`437da3cb 33d2            xor     edx,edx
00007ff7`437da3cd ff1555850100    call    qword ptr [notepad!_imp_GetMessageW (00007ff7`437f2928)]
```

Then we check that *NtUserGetMessage* function was called from *GetMessageW* function:

```
0:000> k
 # Child-SP          RetAddr           Call Site
00 000000a1`c2ccf988 00007ffe`dc19477d win32u!NtUserGetMessage+0x14
01 000000a1`c2ccf990 00007ff7`437da3d3 user32!GetMessageW+0x2d
02 000000a1`c2ccf9f0 00007ff7`437f02b7 notepad!WinMain+0x293
03 000000a1`c2ccfac0 00007ffe`dc557bd4 notepad!__mainCRTStartup+0x19f
04 000000a1`c2ccfb80 00007ffe`ddc6cee1 kernel32!BaseThreadInitThunk+0x14
05 000000a1`c2ccfbb0 00000000`00000000 ntdll!RtlUserThreadStart+0x21
```

```
0:000> ub 00007ffe`dc19477d
user32!GetMessageW+0x9:
00007ffe`dc194759 488bd9          mov     rbx,rcx
00007ffe`dc19475c 458bc8          mov     r9d,r8d
00007ffe`dc19475f 440bc8          or      r9d,eax
00007ffe`dc194762 41f7c10000feff  test    r9d,0FFFE0000h
00007ffe`dc194769 0f8531db0100    jne     user32!GetMessageW+0x1db50 (00007ffe`dc1b22a0)
00007ffe`dc19476f 448bc8          mov     r9d,eax
00007ffe`dc194772 48897c2460      mov     qword ptr [rsp+60h],rdi
00007ffe`dc194777 ff158b320600    call    qword ptr [user32!_imp_NtUserGetMessage (00007ffe`dc1f7a08)]
```

Note: Remember the functions call each other from bottom to top. The topmost function from the stack trace is the last one that was called. **ExceptionAddress** may point to the last one. We would come to this in the real exception process dumps later.

11. Now we check the list of loaded modules using **lm** command:

```
0:000> lm
start             end                 module name
00007ff7`437d0000 00007ff7`43802000   notepad    (pdb symbols)          C:\ProgramData\dbg\sym\notepad.pdb\48F76637AE64DAE8764C8F9F4B27AEA51\notepad.pdb
00007ffe`b0ae0000 00007ffe`b0bb7000   efswrt     (deferred)
00007ffe`c67e0000 00007ffe`c6a65000   comctl32   (deferred)
00007ffe`c9300000 00007ffe`c939e000   TextInputFramework   (deferred)
00007ffe`cbd70000 00007ffe`cbe82000   MrmCoreR   (deferred)
00007ffe`cdb70000 00007ffe`cdbd5000   oleacc     (deferred)
00007ffe`d2d80000 00007ffe`d2d9b000   mpr        (deferred)
00007ffe`d3280000 00007ffe`d3526000   iertutil   (deferred)
00007ffe`d5fb0000 00007ffe`d6103000   WinTypes   (deferred)
00007ffe`d6110000 00007ffe`d643a000   CoreUIComponents   (deferred)
00007ffe`d8590000 00007ffe`d8664000   CoreMessaging   (deferred)
00007ffe`d89f0000 00007ffe`d8a89000   uxtheme    (deferred)
00007ffe`d8ac0000 00007ffe`d8d1a000   twinapi_appcore   (pdb symbols)
C:\ProgramData\dbg\sym\twinapi.appcore.pdb\DA642C52541E6BAA4169D2DD7701DF701\twinapi.appcore.pdb
00007ffe`d8e10000 00007ffe`d8e39000   rmclient   (deferred)
00007ffe`d9b60000 00007ffe`d9b91000   ntmarta    (deferred)
00007ffe`daad0000 00007ffe`daae0000   umpdc      (deferred)
00007ffe`daae0000 00007ffe`dab2a000   powrprof   (deferred)
00007ffe`dab30000 00007ffe`dab41000   kernel_appcore   (deferred)
00007ffe`dab70000 00007ffe`dab8f000   profapi    (deferred)
00007ffe`dab90000 00007ffe`dad24000   gdi32full   (deferred)
00007ffe`dad30000 00007ffe`dad51000   win32u     (pdb symbols)          C:\ProgramData\dbg\sym\win32u.pdb\BC2E49ABE46D2E93B278B4DECFCA62A81\win32u.pdb
00007ffe`dae40000 00007ffe`dae8a000   cfgmgr32   (deferred)
00007ffe`dae90000 00007ffe`daf10000   bcryptPrimitives   (deferred)
00007ffe`daf10000 00007ffe`db00a000   ucrtbase   (deferred)
00007ffe`db010000 00007ffe`db2b3000   KERNELBASE   (export symbols)       KERNELBASE.dll
00007ffe`db2c0000 00007ffe`db35e000   msvcp_win   (deferred)
00007ffe`db360000 00007ffe`db377000   cryptsp    (deferred)
00007ffe`db3e0000 00007ffe`dbb5e000   windows_storage   (deferred)
00007ffe`dbe70000 00007ffe`dbec2000   shlwapi    (deferred)
00007ffe`dbf60000 00007ffe`dc080000   rpcrt4     (deferred)
00007ffe`dc170000 00007ffe`dc303000   user32     (pdb symbols)          C:\ProgramData\dbg\sym\user32.pdb\BC4CC7CC9A33B8A66AEA91BFB0D5FCAA1\user32.pdb
00007ffe`dc400000 00007ffe`dc536000   msctf      (deferred)
00007ffe`dc540000 00007ffe`dc5f2000   kernel32   (pdb symbols)          C:\ProgramData\dbg\sym\kernel32.pdb\5A77DE8CE8D58731F0EA38F1C92F48D81\kernel32.pdb
00007ffe`dca80000 00007ffe`dcb29000   SHCore     (deferred)
00007ffe`dcb40000 00007ffe`dcbe2000   clbcatq    (deferred)
00007ffe`dcbf0000 00007ffe`dcc16000   gdi32      (deferred)
00007ffe`dcc20000 00007ffe`dccc3000   advapi32   (deferred)
00007ffe`dce80000 00007ffe`dd565000   shell32    (deferred)
00007ffe`dd570000 00007ffe`dd634000   oleaut32   (deferred)
00007ffe`dd710000 00007ffe`dd7a7000   sechost    (deferred)
00007ffe`dd7b0000 00007ffe`ddae6000   combase    (private pdb symbols)  C:\ProgramData\dbg\sym\combase.pdb\4BD3112C741B7D1CC4A59CBD4F196A461\combase.pdb
00007ffe`ddaf0000 00007ffe`ddb8e000   msvcrt     (deferred)
00007ffe`ddb90000 00007ffe`ddbbe000   imm32      (deferred)
00007ffe`ddc00000 00007ffe`dddf0000   ntdll      (pdb symbols)          C:\ProgramData\dbg\sym\ntdll.pdb\27A66DD3103F6B2E03B27D315F1A8AF31\ntdll.pdb
```

Note: **start** and **end** addresses show where modules are loaded in process virtual memory. You can see the module contents by using **dc** command (**Unknown Component** pattern):

```
0:000> dc 00007ff7`437d0000 00007ff7`43802000
00007ff7`437d0000  00905a4d 00000003 00000004 0000ffff  MZ..............
00007ff7`437d0010  000000b8 00000000 00000040 00000000  ........@.......
00007ff7`437d0020  00000000 00000000 00000000 00000000  ................
00007ff7`437d0030  00000000 00000000 00000000 000000f8  ................
00007ff7`437d0040  0eba1f0e cd09b400 4c01b821 685421cd  ........!..L.!Th
00007ff7`437d0050  70207369 72676f72 63206d61 6f6e6e61  is program canno
00007ff7`437d0060  65622074 6e757220 206e6920 20534f44  t be run in DOS
00007ff7`437d0070  65646f6d 0a0d0d2e 00000024 00000000  mode....$.......
00007ff7`437d0080  29f17b95 7a9f1ad1 7a9f1ad1 7a9f1ad1  .{.)...z...z...z
00007ff7`437d0090  7a0c62d8 7a9f1a83 7b9c728a 7a9f1ad2  .b.z...z.r.{...z
00007ff7`437d00a0  7b9b728a 7a9f1ac4 7b9a728a 7a9f1ad7  .r.{...z.r.{...z
00007ff7`437d00b0  7b9e728a 7a9f1ad8 7a9e1ad1 7a9f1bf6  .r.{...z...z...z
00007ff7`437d00c0  7b96728a 7a9f1ac8 7a60728a 7a9f1ad0  .r.{...z.r`z...z
00007ff7`437d00d0  7b9d728a 7a9f1ad0 68636952 7a9f1ad1  .r.{...zRich...z
00007ff7`437d00e0  00000000 00000000 00000000 00000000  ................
00007ff7`437d00f0  00000000 00000000 00004550 00078664  ........PE..d...
00007ff7`437d0100  9e7797dd 00000000 00000000 002200f0  ..w..........".
00007ff7`437d0110  0f0e020b 00020600 0000d400 00000000  ................
[...]
00007ff7`43801f90  00000000 00000000 00000000 00000000  ................
00007ff7`43801fa0  00000000 00000000 00000000 00000000  ................
00007ff7`43801fb0  00000000 00000000 00000000 00000000  ................
00007ff7`43801fc0  00000000 00000000 00000000 00000000  ................
00007ff7`43801fd0  00000000 00000000 00000000 00000000  ................
00007ff7`43801fe0  00000000 00000000 00000000 00000000  ................
00007ff7`43801ff0  00000000 00000000 00000000 00000000  ................
00007ff7`43802000  ????????                              ????
```

12. We can check verbose module information using **lmv** command or use **lmv m** *<module name>* to check an individual module (**Not My Version** pattern):

```
0:000> lmv m notepad
Browse full module list
start             end                 module name
00007ff7`437d0000 00007ff7`43802000   notepad    (pdb symbols)
C:\ProgramData\Dbg\sym\notepad.pdb\48F76637AE64DAE8764C8F9F4B27AEA51\notepad.pdb
    Loaded symbol image file: notepad.exe
    Image path: C:\Windows\System32\notepad.exe
    Image name: notepad.exe
    Browse all global symbols  functions  data
    Image was built with /Brepro flag.
    Timestamp:        9E7797DD (This is a reproducible build file hash, not a timestamp)
    CheckSum:         00034590
    ImageSize:        00032000
    File version:     10.0.18362.1
    Product version:  10.0.18362.1
    File flags:       0 (Mask 3F)
    File OS:          40004 NT Win32
    File type:        1.0 App
    File date:        00000000.00000000
    Translations:     0409.04b0
    Information from resource tables:
        CompanyName:      Microsoft Corporation
        ProductName:      Microsoft® Windows® Operating System
        InternalName:     Notepad
        OriginalFilename: NOTEPAD.EXE
        ProductVersion:   10.0.18362.1
        FileVersion:      10.0.18362.1 (WinBuild.160101.0800)
```

```
        FileDescription:  Notepad
        LegalCopyright:   © Microsoft Corporation. All rights reserved.
```

13. Sometimes **lmv** command doesn't show much and **!lmi** command might give extra information:

```
0:000> !lmi notepad
Loaded Module Info: [notepad]
         Module: notepad
   Base Address: 00007ff7437d0000
     Image Name: notepad.exe
   Machine Type: 34404 (X64)
     Time Stamp: 9e7797dd (This is a reproducible build file hash, not a true timestamp)
           Size: 32000
       CheckSum: 34590
Characteristics: 22
Debug Data Dirs: Type  Size     VA  Pointer
             CODEVIEW    24, 26ec8,   258c8 RSDS - GUID: {48F76637-AE64-DAE8-764C-8F9F4B27AEA5}
               Age: 1, Pdb: notepad.pdb
                 POGO   3ac, 26eec,   258ec [Data not mapped]
                REPRO    24, 27298,   25c98 Reproducible build
     Image Type: MEMORY   - Image read successfully from loaded memory.
    Symbol Type: PDB      - Symbols loaded successfully from image header.

C:\ProgramData\Dbg\sym\notepad.pdb\48F76637AE64DAE8764C8F9F4B27AEA51\notepad.pdb
    Load Report: public symbols , not source indexed

C:\ProgramData\Dbg\sym\notepad.pdb\48F76637AE64DAE8764C8F9F4B27AEA51\notepad.pdb
```

Note: We can also use **lmt** command variant if we are interested in timestamps only.

14. Sometimes **Environment Hint** pattern can give troubleshooting suggestions related to environment variables and DLL paths. **!peb** command (**P**rocess **E**nvironment **B**lock):

```
0:000> !peb
PEB at 000000a1c2f93000
    InheritedAddressSpace:    No
    ReadImageFileExecOptions: No
    BeingDebugged:            No
    ImageBaseAddress:         00007ff7437d0000
    NtGlobalFlag:             0
    NtGlobalFlag2:            0
    Ldr                       00007ffeddd653c0
    Ldr.Initialized:          Yes
    Ldr.InInitializationOrderModuleList: 0000022119ac2420 . 0000022119af9fa0
    Ldr.InLoadOrderModuleList:           0000022119ac2590 . 0000022119af9f80
    Ldr.InMemoryOrderModuleList:         0000022119ac25a0 . 0000022119af9f90
                    Base TimeStamp                     Module
            7ff7437d0000 9e7797dd Apr 01 07:29:49 2054 C:\WINDOWS\system32\notepad.exe
            7ffeddc00000 5c516d44 Jan 30 09:24:20 2019 C:\WINDOWS\SYSTEM32\ntdll.dll
            7ffedc540000 d0cecc10 Jan 04 10:53:04 2081 C:\WINDOWS\System32\KERNEL32.DLL
            7ffedb010000 7c860e56 Mar 15 02:17:58 2036 C:\WINDOWS\System32\KERNELBASE.dll
            7ffedcbf0000 90b22122 Dec 05 03:23:14 2046 C:\WINDOWS\System32\GDI32.dll
            7ffedad30000 5343f4fb Apr 08 14:09:15 2014 C:\WINDOWS\System32\win32u.dll
            7ffedab90000 ce634582 Sep 22 13:45:54 2079 C:\WINDOWS\System32\gdi32full.dll
            7ffedb2c0000 2085286c Apr 16 19:52:28 1987 C:\WINDOWS\System32\msvcp_win.dll
            7ffedaf10000 080a13f7 Apr 11 03:09:59 1974 C:\WINDOWS\System32\ucrtbase.dll
            7ffedc170000 aaa5ecca Sep 21 14:59:38 2060 C:\WINDOWS\System32\USER32.dll
            7ffeddaf0000 f5bdefd7 Aug 25 09:27:03 2100 C:\WINDOWS\System32\msvcrt.dll
            7ffedd7b0000 a927eb79 Dec 06 19:48:09 2059 C:\WINDOWS\System32\combase.dll
            7ffedbf60000 0530c620 Oct 04 22:35:28 1972 C:\WINDOWS\System32\RPCRT4.dll
            7ffedae90000 5c8eaa57 Mar 17 20:13:11 2019 C:\WINDOWS\System32\bcryptPrimitives.dll
            7ffedca80000 48cfe63c Sep 16 18:00:44 2008 C:\WINDOWS\System32\shcore.dll
            7ffedcc20000 0ba7a4cd Mar 13 03:34:37 1976 C:\WINDOWS\System32\advapi32.dll
```

```
            7ffedd710000 1c757ba0 Feb 17 04:10:08 1985 C:\WINDOWS\System32\sechost.dll
            7ffec67e0000 cd8762a1 Apr 08 18:51:29 2079 C:\WINDOWS\WinSxS\amd64_microsoft.windows.common-
controls_6595b64144ccf1df_6.0.18362.329_none_e6c7b0c7130c72de\COMCTL32.dll
            7ffeddb90000 9ca2b089 Apr 10 15:22:01 2053 C:\WINDOWS\System32\IMM32.DLL
            7ffedab30000 05bef372 Jan 20 17:50:42 1973 C:\WINDOWS\System32\kernel.appcore.dll
            7ffed89f0000 fdddb636 Dec 19 21:35:50 2104 C:\WINDOWS\system32\uxtheme.dll
            7ffedcb40000 9506208f Mar 24 13:52:15 2049 C:\WINDOWS\System32\clbcatq.dll
            7ffecbd70000 1cab1146 Mar 29 19:38:46 1985 C:\Windows\System32\MrmCoreR.dll
            7ffedc400000 a4e84978 Sep 02 21:03:04 2057 C:\WINDOWS\System32\MSCTF.dll
            7ffedd570000 8fcb7820 Jun 13 05:20:48 2046 C:\WINDOWS\System32\OLEAUT32.dll
            7ffedb3e0000 d508d6ae Apr 05 05:50:54 2083 C:\WINDOWS\System32\windows.storage.dll
            7ffedab70000 facae6c0 May 03 03:09:04 2103 C:\WINDOWS\System32\profapi.dll
            7ffedaae0000 fdc4588a Nov 30 15:49:30 2104 C:\WINDOWS\System32\powrprof.dll
            7ffedaad0000 a2ccd413 Jul 20 16:30:27 2056 C:\WINDOWS\System32\UMPDC.dll
            7ffedbe70000 f8807ba1 Feb 12 06:43:45 2102 C:\WINDOWS\System32\shlwapi.dll
            7ffeb0ae0000 993fa218 Jun 22 23:05:12 2051 C:\Windows\System32\efswrt.dll
            7ffed2d80000 57b24d2b Aug 16 00:15:55 2016 C:\Windows\System32\MPR.dll
            7ffed5fb0000 9d34108f Jul 29 21:50:23 2053 C:\WINDOWS\SYSTEM32\wintypes.dll
            7ffed8ac0000 42f071ca Aug 03 08:27:06 2005 C:\Windows\System32\twinapi.appcore.dll
            7ffed8e10000 5be0eb17 Nov 06 01:15:03 2018 C:\Windows\System32\RMCLIENT.dll
            7ffedce80000 0f63808d Mar 08 04:26:53 1978 C:\WINDOWS\System32\SHELL32.dll
            7ffedae40000 afaaabaa May 24 03:04:26 2063 C:\WINDOWS\System32\cfgmgr32.dll
            7ffedb360000 a51023f1 Oct 03 02:33:37 2057 C:\WINDOWS\System32\cryptsp.dll
            7ffecdb70000 20690459 Mar 26 10:35:05 1987 C:\Windows\System32\oleacc.dll
            7ffec9300000 a5b3d27e Feb 04 05:17:50 2058 C:\WINDOWS\System32\TextInputFramework.dll
            7ffed6110000 d02b205d Sep 02 08:21:01 2080 C:\WINDOWS\System32\CoreUIComponents.dll
            7ffed8590000 101a45b8 Jul 24 20:40:40 1978 C:\WINDOWS\System32\CoreMessaging.dll
            7ffed9b60000 d95e6299 Jul 24 20:31:37 2085 C:\WINDOWS\SYSTEM32\ntmarta.dll
            7ffed3280000 ceda2ec9 Dec 21 17:28:41 2079 C:\WINDOWS\System32\iertutil.dll
    SubSystemData:     00007ffed8cee120
    ProcessHeap:       0000022119ac0000
    ProcessParameters: 0000022119ac1b40
    CurrentDirectory:  'C:\Users\dumpa\'
    WindowTitle:  'C:\Users\dumpa\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Accessories\Notepad.lnk'
    ImageFile:    'C:\WINDOWS\system32\notepad.exe'
    CommandLine:  '"C:\WINDOWS\system32\notepad.exe" '
    DllPath:      '< Name not readable >'
    Environment:  0000022119ac0fe0
        =::=::\
        ALLUSERSPROFILE=C:\ProgramData
        APPDATA=C:\Users\dumpa\AppData\Roaming
        CommonProgramFiles=C:\Program Files\Common Files
        CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
        CommonProgramW6432=C:\Program Files\Common Files
        COMPUTERNAME=DESKTOP-OGPC0LO
        ComSpec=C:\WINDOWS\system32\cmd.exe
        DriverData=C:\Windows\System32\Drivers\DriverData
        HOMEDRIVE=C:
        HOMEPATH=\Users\dumpa
        LOCALAPPDATA=C:\Users\dumpa\AppData\Local
        LOGONSERVER=\\DESKTOP-OGPC0LO
        NUMBER_OF_PROCESSORS=2
        OneDrive=C:\Users\dumpa\OneDrive
        OS=Windows_NT

Path=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\Sy
stem32\OpenSSH\;C:\Users\dumpa\AppData\Local\Microsoft\WindowsApps;
        PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
        PROCESSOR_ARCHITECTURE=AMD64
        PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 142 Stepping 10, GenuineIntel
        PROCESSOR_LEVEL=6
        PROCESSOR_REVISION=8e0a
        ProgramData=C:\ProgramData
        ProgramFiles=C:\Program Files
        ProgramFiles(x86)=C:\Program Files (x86)
        ProgramW6432=C:\Program Files
        PSModulePath=C:\Program Files\WindowsPowerShell\Modules;C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules
        PUBLIC=C:\Users\Public
        SESSIONNAME=Console
        SystemDrive=C:
        SystemRoot=C:\WINDOWS
        TEMP=C:\Users\dumpa\AppData\Local\Temp
        TMP=C:\Users\dumpa\AppData\Local\Temp
        USERDOMAIN=DESKTOP-OGPC0LO
        USERDOMAIN_ROAMINGPROFILE=DESKTOP-OGPC0LO
        USERNAME=Training
```

```
        USERPROFILE=C:\Users\dumpa
        windir=C:\WINDOWS
```

15. To launch classic help from WinDbg Preview, type **.hh** command.

16. We close logging before exiting WinDbg:

```
0:000> .logclose
Closing open log file C:\AWMDA-Dumps\Process\x64\notepad.log
```