# Evaluierung von Tools zum Auffinden von Undefined Behavior

**T3000**

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

**Christoph Böhringer**

21.06.2021


| | |
|---|---|
| **Matrikelnummer, Kurs:** | 3275565, TINF18-IN |
| **Ausbildungsfirma:** | Mitutoyo CTL Germany GmbH |
| **Betreuer:** | Dipl.-Inform. (FH) Thomas Weller |

# Erklärung

Ich versichere hiermit, dass ich meine Thesis mit dem Thema „Entwicklung eines softwaregestützten kontextabhängigen Kommunikationsassistenten “ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort, Datum

_____

Name

# Abstract

In einem Softwareprodukt trat bei der Umstellung von 32 Bit auf 64 Bit ein Fehler auf, dessen Ursache sich vermutlich auf Undefined Behavior von C++ zurückführen lässt. Diese Arbeit soll den Nachweis erbringen oder widerlegen, dass Undefined Behavior die Ursache für den Fehler war. Optional kann eine mögliche Erklärung gesucht werden, warum der betroffene Code in der 32 Bit Version keinen Fehler verursacht hat.

Das betroffene Projekt wurde vor der Umstellung mehrere Jahre lang nicht verändert. Es muss daher davon ausgegangen werden, dass sich ähnliche Fehler damals auch an anderen Stellen eingeschlichen haben, jedoch noch nicht bemerkt wurden. Die Arbeit soll untersuchen, ob weitere Fehler vom Typ Undefined Behavior in diesem Projekt vorliegen.

Um Fehler dieser Art auch in anderen Projekten auszuschließen, sollen Tools gesucht und evaluiert werden, mit denen die Fehler automatisiert gefunden und berichtet werden können. Falls kein passendes Tool existiert, soll aufgezeigt werden, mit welchem manuellen Vorgehen solche Stellen erkannt werden können.

Die Tools sollen möglichst mit den bestehenden Entwicklungsumgebungen verwendet werden können und idealerweise alle Arten von Undefined Behavior erkennen.

Hinweis: Da in der Arbeit vermutlich Source Code der Mitutoyo CTL Germany GmbH offengelegt wird, ist die Arbeit unter Verschluss zu halten.

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Kapitel 1

# Anhang

## 1.1 WinDbg Crash Analyse

Vom betroffenen Absturz, der womöglich auf Undefined Behavior zurückzuführen ist, wurde ein Crash Dump erstellt. Dieser Anhang beschreibt anhand einer aufgezeichneten Logdatei, welche Befehle genutzt wurden, um den Crash Dump zu analysieren. Bei der analysierten Datei handelt es sich um einen Crash Dump mit vollständig enthaltenem Speicherinhalt. Die Datei ist ca. 1 GB groß.

### 1.1.1 Grundlegende Prüfungen

```
0:000> ||
.  0 Full memory user mini dump: D:\MINIDUMP-20201217-150359.DMP
```

Dem Dateinamen nach wurde der Crash Dump am 17.12.2020 um 15:03:59 Uhr UTC geschrieben. Die Zeitangabe innerhalb des Crash Dumps bestätigt dies.

```
0:000> .time
Debug session time: Thu Dec 17 16:04:00.000 2020 (UTC + 1:00)
System Uptime: 0 days 7:25:08.968
Process Uptime: 0 days 0:01:31.000
```

```
  Kernel time: 0 days 0:00:32.000
  User time: 0 days 0:00:48.000
```

Der Bug wurde am 27.11.2020 berichtet und am 11.12.2020 erstmals bestätigt. Beim vorliegenden Crash Dump handelt es sich also um die Reproduktion des Fehlers.

```
0:000> |
.  0  id: 2b7c examine name: C:\MCOSMOSx64\EXE\GEOPAK64.exe
```

Das ausgeführte Programm stimmt mit dem des Fehlerberichts überein.

```
0:000> .exr -1
*** WARNING: Unable to verify checksum for MnGeom364.dll
ExceptionAddress: 00007ff995c1d48e (MnGeom364!tg_ajc_lin_int+0x000000000000014e)
   ExceptionCode: c0000005 (Access violation)
  ExceptionFlags: 00000000
NumberParameters: 2
   Parameter[0]: 0000000000000000
   Parameter[1]: 0000000000000000
Attempt to read from address 0000000000000000
```

Der Fehlercode und das fehlerhafte Modul stimmen ebenfalls mit dem des Fehlerberichts überein. Es kann also eine detailliertere Analyse durchgeführt werden.

### 1.1.2 Symbole

Für eine weitere Analyse ist das Vorhandensein von Symbolen erforderlich. Manche Symbole wurden zusammen mit dem Crash Dump abgelegt. Diese Symbole sollten zuerst berücksichtigt werden und müssen zuerst in den Symbolpath aufgenommen werden.

```
0:000> .sympath "D:\temp\Bug 32147"
Symbol search path is: D:\temp\Bug 32147
Expanded Symbol search path is: d:\temp\bug 32147
```

```
************* Path validation summary **************
Response                         Time (ms)    Location
OK                                            D:\temp\Bug 32147
```

Leider stimmen bei den Symbolen die Zeitstempel nicht überein, so dass die Symbole nicht geladen werden können. Die Prüfung des Zeitstempels kann jedoch ausgeschaltet werden.

```
0:000> .symopt+ 0x40
Symbol options are 0x30377:
  0x00000001 - SYMOPT_CASE_INSENSITIVE
  0x00000002 - SYMOPT_UNDNAME
  0x00000004 - SYMOPT_DEFERRED_LOADS
  0x00000010 - SYMOPT_LOAD_LINES
  0x00000020 - SYMOPT_OMAP_FIND_NEAREST
  0x00000040 - SYMOPT_LOAD_ANYTHING
  0x00000100 - SYMOPT_NO_UNQUALIFIED_LOADS
  0x00000200 - SYMOPT_FAIL_CRITICAL_ERRORS
  0x00010000 - SYMOPT_AUTO_PUBLICS
  0x00020000 - SYMOPT_NO_IMAGE_SEARCH
```

Weitere Symbole können vom Azure DevOps Server geladen werden

```
0:000> .sympath+ srv*d:\debug\symbols*\\tfs-build-2014\SymbolStore\
    ↪ Mitutoyo.MCOSMOS.BasicLibs_Release_NuGet
Symbol search path is: D:\temp\Bug 32147;srv*d:\debug\symbols*\\tfs-build-2014\
    ↪ SymbolStore\Mitutoyo.MCOSMOS.BasicLibs_Release_NuGet
Expanded Symbol search path is: d:\temp\bug 32147;srv*d:\debug\symbols*\\tfs-
    ↪ build-2014\symbolstore\mitutoyo.mcosmos.basiclibs_release_nuget

************* Path validation summary **************
Response                         Time (ms)    Location
OK                                            D:\temp\Bug 32147
Deferred                                      srv*d:\debug\symbols*\\tfs-build-2014\
    ↪ SymbolStore\Mitutoyo.MCOSMOS.BasicLibs_Release_NuGet
```

Und letztlich muss der Microsoft Server abgefragt werden, damit die Funktionen des Betriebssystems korrekt aufgelöst werden können.

```
0:000> .symfix+ d:\debug\symbols
```

Nach dem Ändern der möglichen Quellen für Symbole muss dem Debugger mitgeteilt

werden, dass er seine Informationen aktualisiert.

```
0:000> .reload /f
.*** WARNING: Unable to verify checksum for GEOPAK64.exe
.......*** WARNING: Unable to verify checksum for GeoWinBinToAsc64.dll
.....*** WARNING: Unable to verify checksum for Mafis_3DCmp64.dll
.......*** WARNING: Unable to verify checksum for MnGeoWnListCtrl64.dll
.*** WARNING: Unable to verify checksum for MnRecordPoints64.dll
.*** WARNING: Unable to verify checksum for UncertaintyCalculator64.dll
..

Press ctrl-c (cdb, kd, ntsd) or ctrl-break (windbg) to abort symbol loads that
    ↪ take too long.
Run !sym noisy before .reload to track down problems loading symbols.

[...]
Loading unloaded module list
.........................

************* Symbol Loading Error Summary **************
Module name          Error
WkWin64              The system cannot find the file specified
GeoWinBinToAsc64     The system cannot find the file specified
[...]

You can troubleshoot most symbol related issues by turning on symbol loading
    ↪ diagnostics (!sym noisy) and repeating the command that caused symbols to
    ↪  be loaded.
You should also verify that your symbol search path (.sympath) is correct.
```

Anhand der beiden bekannten und für die Fehleranalyse notwendigen Module Geopak und MnGeom3 kann überprüft werden, ob die Symbole korrekt geladen wurden.

```
0:000> lm m geopak*
Browse full module list
start            end                module name
00007ff7‘fa2d0000 00007ff7‘fc142000 GEOPAK64 C (private pdb symbols) d:\temp\bug
    ↪  32147\GEOPAK64.pdb

0:000> lm m mngeom364
Browse full module list
start            end                module name
00007ff9‘95c00000 00007ff9‘95c2c000 MnGeom364 C (private pdb symbols) d:\temp\
    ↪ bug 32147\MnGeom364.pdb
```

Für beide Module sind Symbole mit Informationen zu privaten Methoden etc. vorhanden.

### 1.1.3 Exception Analyse

Wie bereits bei den grundlegenden Prüfungen gesehen, handelt es sich beim Absturz um eine Access Violation, also eine Art NullPointerException. Nach dem Einstellen der Symbole verschwindet allerdings die Warnung.

```
0:000> .exr -1
ExceptionAddress: 00007ff995c1d48e (MnGeom364!tg_ajc_lin_int+0x000000000000014e)
   ExceptionCode: c0000005 (Access violation)
  ExceptionFlags: 00000000
NumberParameters: 2
   Parameter[0]: 0000000000000000
   Parameter[1]: 0000000000000000
Attempt to read from address 0000000000000000
```

Der Callstack liefert nicht die richtigen Angaben

```
0:000> k
 # Child-SP          RetAddr               Call Site
00 00000056‘12337058 00000193‘813b1bec ntdll!NtGetContextThread+0x14
01 00000056‘12337060 0000001f‘00000002 0x00000193‘813b1bec
02 00000056‘12337068 0000003e‘0000003e 0x0000001f‘00000002
03 00000056‘12337070 00009c67‘02cb62cf 0x0000003e‘0000003e
04 00000056‘12337078 00009c67‘02cb7d3f 0x00009c67‘02cb62cf
05 00000056‘12337080 00000000‘00000000 0x00009c67‘02cb7d3f
```

Dies bedeutet, dass der Kontext noch nicht auf die Exception gesetzt ist.

```
0:000> .ecxr
rax=0000000000000000 rbx=000000561233b3f8 rcx=000000561233af18
rdx=fffffffffffffe0 rsi=000000561233b490 rdi=000000561233b470
rip=00007ff995c1d48e rsp=000000561233afe0 rbp=000000561233b0e0
 r8=00000193a1a21c90 r9=0000000000000014 r10=000000000000003c
```

```
r11=000000561233afd0 r12=00000193a1a21c90 r13=000000561233b608
r14=0000000000000014 r15=0000000000000001
iopl=0         nv up ei pl nz na pe nc
cs=0033 ss=002b ds=002b es=002b fs=0053 gs=002b             efl=00010202
MnGeom364!tg_ajc_lin_int+0x14e:
00007ff9‘95c1d48e 0f1000      movups xmm0,xmmword ptr [rax] ds:00000000‘
    ↪ 00000000=????????????????????????????????
```

Nach dem Setzen des Kontexts wird die Methode `tg_ajc_lin_int` auf dem Stack erkannt.

```
0:000> k Lb
  *** Stack trace for last set context - .thread/.cxr resets it
 # Child-SP        RetAddr             Call Site
00 00000056‘1233afe0 00007ff9‘95c1d8b8 MnGeom364!tg_ajc_lin_int+0x14e [d:\
    ↪ gitrepos\mitutoyo.mcosmos.basicslibs\source\mngeom3\tg_geom.c @ 955]
01 00000056‘1233b350 00007ff9‘95c1fefb MnGeom364!tg_ajc_lin_rl+0xd8 [d:\gitrepos
    ↪ \mitutoyo.mcosmos.basicslibs\source\mngeom3\tg_geom.c @ 993]
02 00000056‘1233b560 00007ff7‘fae310df MnGeom364!tg_inn_ajc_lin+0x2b [d:\
    ↪ gitrepos\mitutoyo.mcosmos.basicslibs\source\mngeom3\tg_geom.c @ 1026]
03 00000056‘1233b6b0 00007ff7‘fae4a291 GEOPAK64!pel_line+0x77f [g:\git\mcosmos50
    ↪ \geopak\source\geopak\pelmadcp.cpp @ 5498]
04 00000056‘1233bf00 00007ff7‘fae49687 GEOPAK64!pelm_comp_elem_intern+0xc01 [g:\
    ↪ git\mcosmos50\geopak\source\geopak\pelmadcp.cpp @ 8255]
05 00000056‘1233e730 00007ff7‘fa7f5a88 GEOPAK64!pelm_comp_elem+0x77 [g:\git\
    ↪ mcosmos50\geopak\source\geopak\pelmadcp.cpp @ 8651]
06 00000056‘1233e790 00007ff7‘fa920b77 GEOPAK64!fctctr_cpnt_end+0x748 [g:\git\
    ↪ mcosmos50\geopak\source\geopak\fctctrmngr.cpp @ 1376]
07 00000056‘1233efe0 00007ff7‘fa91d26c GEOPAK64!fctmngr_wrk_fct+0x1177 [g:\git\
    ↪ mcosmos50\geopak\source\geopak\fct_mngr.cpp @ 1314]
08 00000056‘1233f070 00007ff7‘faaf8df3 GEOPAK64!fctmngr_wrk+0x68c [g:\git\
    ↪ mcosmos50\geopak\source\geopak\fct_mngr.cpp @ 1924]
09 00000056‘1233f120 00007ff7‘fabd9301 GEOPAK64!CGeopakDoc::PartProgCmdWrk+0x13
    ↪ [g:\git\mcosmos50\geopak\source\geopak\geopakdoc.cpp @ 3711]
0a 00000056‘1233f150 00007ff9‘ab5287f9
    ↪ GEOPAK64!CMainFrame::OnMsgNvUserEvent+0x281 [g:\git\mcosmos50\geopak\
    ↪ source\geopak\mainfrm.cpp @ 2215]
```

Ausgehend von einer Nutzerinteraktion (`OnMsgNvUserEvent`) werden mehrere Methoden aufgerufen, die dann zum Absturz führen.