

AppCodeTrace

AppCodeTrace enables the profiling of a managed .NET target application. It is a command line tool that traces the execution of .NET compiled assemblies (.DLL or .EXE binaries) by investigating classes that represent the assembly and its modules, types, methods, properties and fields, and the behavior of function calls and the order in which functions are invoked from the main function.

To display the filename, line number and column number of functions, AppCodeTrace must have access to the symbol (.PDB) files for the target application that is to be analyzed. A program database (PDB) file holds debugging and project state information, such as line numbers, source code file information and sequence points.

AppCodeTrace helps you understand the behavior of functions invoked within an assembly, even when .PDB files are not available, by retrieving the Intermediate Language (IL) code and instrumenting it. A backup of the target application is created in the backup directory prior to profiling the target application.

Installation

To install AppCodeTrace, run the `AppCodeTraceCmdSetup.msi` file. This installs AppCodeTrace in the "C:\Program Files" directory by default. Upon successful installation, three folders are created in the install path:

- Backup
- Logs
- Silverlight

AppCodeTrace instruments the target application and *injects* the instrumented byte/IL-code into the target application automatically. We refer to this process as *patching* the binary. A copy of the executable is stored in the `backup` directory with the file extension `.bak` appended to the name of the executable, before being *patched*. Results of this patched .DLL or .EXE file are stored in log files in the `logs` directory. A file with the extension `.log` appended to the name of the executable is created to record a list of function calls in the instrumented IL code.

For example, if `SampleLog.exe` is provided to `AppCodeTraceCmd.exe` as the file to be patched, the file `SampleLog.exe` is first copied to the backup directory and renamed to `SampleLog.exe.bak`, prior to being patched.

AppCodeTrace also instruments Silverlight applications. Silverlight, a free plug-in, powered by the .NET framework and compatible with multiple browsers, devices and operating systems, is a powerful development tool for creating rich, interactive user experiences for Web and mobile applications. A Silverlight application is bundled into a .XAP file and contains the client DLL for the Silverlight application.

The .XAP file is a compressed file that uses the standard .zip compression algorithm to minimize client download size. Rename the .XAP file extension to a .ZIP file extension, to view and extract the contents.

Usage

AppCodeTrace does not change the original IL code; it inserts calls to library routines for each function. When AppCodeTrace runs, it looks in the current directory of the loaded assembly for a matching debug (.PDB) file. If a matching debug file does not exist, AppCodeTrace profiles the target application but does not display the filename, line number and column number of functions.

```
AppCodeTraceCmd -input <input file> [-config <config file>]
```

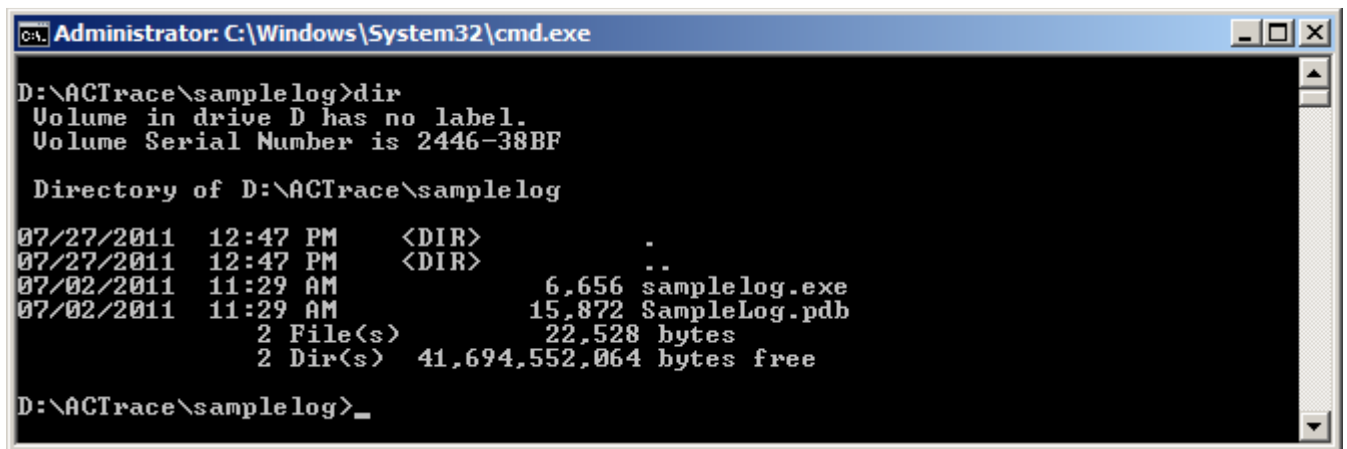
input – .NET DLL/EXE binaries to be instrumented, with absolute path specified, i.e. "D:\AppCodeTrace\test\SampleLog.exe"
config: configuration file to be read. The **default** configuration file is AppCodeTrace.config.

NOTE: Ensure that you have the correct .PDB files for the binaries to be instrumented.

1. Instrumenting .NET binary

To instrument samplelog.exe with a corresponding SampleLog.pdb file, the command is:

```
C:\Program Files\AppCodeTrace>AppCodeTraceCmd -input  
"D:\ACTrace\samplelog\samplelog.exe"
```



```
Administrator: C:\Windows\System32\cmd.exe  
D:\ACTrace\samplelog>dir  
Volume in drive D has no label.  
Volume Serial Number is 2446-38BF  
  
Directory of D:\ACTrace\samplelog  
07/27/2011 12:47 PM <DIR> .  
07/27/2011 12:47 PM <DIR> ..  
07/02/2011 11:29 AM 6,656 samplelog.exe  
07/02/2011 11:29 AM 15,872 SampleLog.pdb  
2 File(s) 22,528 bytes  
2 Dir(s) 41,694,552,064 bytes free  
D:\ACTrace\samplelog>_
```

Figure 1.1 Directory Listing showing the target application samplelog.exe and the corresponding SampleLog.PDB files

```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\AppCodeTrace>AppCodeTraceCmd -input "D:\ACTrace\samplelog\samplelog.exe"
Input file is D:\ACTrace\samplelog\samplelog.exe
Patching started on D:\ACTrace\samplelog\samplelog.exe
skip SampleLog.Validate.get_WhiteList , special method found
skip SampleLog.Validate.set_WhiteList , special method found
D:\ACTrace\samplelog\samplelog.exe patched successfully

C:\Program Files\AppCodeTrace>
```

Figure 1.2 Running AppCodeTrace to instrument samplelog.exe

```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\AppCodeTrace>cd logs
C:\Program Files\AppCodeTrace\logs>dir
Volume in drive C has no label.
Volume Serial Number is AC12-03AB

Directory of C:\Program Files\AppCodeTrace\logs
07/26/2011 06:50 PM <DIR> .
07/26/2011 06:50 PM <DIR> ..
07/13/2010 06:50 AM 0 logs.txt
07/26/2011 06:50 PM 606 samplelog.exe.log
                2 File(s)          606 bytes
                2 Dir(s)  59,377,065,984 bytes free

C:\Program Files\AppCodeTrace\logs>
```

Figure 1.3 samplelog.exe.log file created after patching

```
C:\Windows\System32\cmd.exe
>samplelog.exe ' OR 1=1;--
Input 1: '
Input doesn't contain any invalid characters.

>samplelog.exe "" OR 1=1;--
Input 1: ' OR 1=1;--
Input doesn't contain any invalid characters.

>
```

Figure 1.4 Running the *patched* binary with arguments

Output

The log file shows the functions called along with any input passed from the command line. In this example,

```
samplelog.exe.log
1 Entered SampleLog.Program.Main(System.String[] args) F:\VSSDB\SecurityProjects\.NET Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 13:7
2 Entered SampleLog.Validate.ctor() [] F:\VSSDB\SecurityProjects\.NET Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 55:7
3 Returned SampleLog.Validate.ctor()
4 Entered SampleLog.Validate.CheckSQLInjection(System.String input) [ABC] F:\VSSDB\SecurityProjects\.NET
Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 61:7
5 Returned SampleLog.Validate.CheckSQLInjection(System.String input)
6 Returned SampleLog.Program.Main(System.String[] args)
7 Entered SampleLog.Program.Main(System.String[] args) F:\VSSDB\SecurityProjects\.NET Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 13:7
8 Entered SampleLog.Validate.ctor() [] F:\VSSDB\SecurityProjects\.NET Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 55:7
9 Returned SampleLog.Validate.ctor()
10 Entered SampleLog.Validate.CheckSQLInjection(System.String input) ['] F:\VSSDB\SecurityProjects\.NET
Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 61:7
11 Returned SampleLog.Validate.CheckSQLInjection(System.String input)
12 Returned SampleLog.Program.Main(System.String[] args)
13 Entered SampleLog.Program.Main(System.String[] args) F:\VSSDB\SecurityProjects\.NET Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 13:7
14 Entered SampleLog.Validate.ctor() [] F:\VSSDB\SecurityProjects\.NET Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 55:7
15 Returned SampleLog.Validate.ctor()
16 Entered SampleLog.Validate.CheckSQLInjection(System.String input) [' OR 1=1;--] F:\VSSDB\SecurityProjects\.NET
Instrumentor\Samples\SampleLog\SampleLog\Program.cs - 61:7
17 Returned SampleLog.Validate.CheckSQLInjection(System.String input)
18 Returned SampleLog.Program.Main(System.String[] args)
19
```

Figure 1.5 Contents of the log file samplelog.exe.log

2. Instrumenting Silverlight applications

To instrument Silverlight applications, use the `-silverlight` option. AppCodeTrace will automatically read the contents from the configuration file stored in the silverlight directory.

To instrument the Silverlight application bundled in the file test.xap,

Step 1: Rename `test.xap` to `test.zip`. Extract the contents to a local folder.

Step 2: Pass the `test.dll` file as input to AppCodeTrace. Make sure to use the `-silverlight` option. The configuration file is read from the Silverlight folder under "C:\Program Files\AppCodeTrace"

```
C:\Program Files\AppCodeTrace>AppCodeTraceCmd -input "D:\ACTrace\test\test.dll"  
-silverlight
```

```

Administrator: C:\Windows\System32\cmd.exe
D:\ACTrace\test>dir
Volume in drive D has no label.
Volume Serial Number is 2446-38BF

Directory of D:\ACTrace\test

07/27/2011  03:08 PM    <DIR>          .
07/27/2011  03:08 PM    <DIR>          ..
03/31/2011  04:45 PM                487 AppManifest.xaml
07/25/2011  02:23 PM                650 ServiceReferences.ClientConfig
08/17/2009  10:35 PM            386,912 System.Windows.Controls.dll
03/31/2011  04:44 PM            574,976 test.dll
03/28/2011  11:58 AM            197,632 WeborbClient.dll
           5 File(s)          1,160,657 bytes
           2 Dir(s)         41,692,889,088 bytes free

D:\ACTrace\test>c:
C:\Program Files\AppCodeTrace>AppCodeTraceCmd.exe -input "D:\ACTrace\test\test.d
ll" -silverlight
Input file is D:\ACTrace\test\test.dll
Patching started on D:\ACTrace\test\test.dll
Failed to read pdb file. PDB file does not exist or invalid. Error: ExceptionOcc
urred
skip test.App.get_WeborbURL , special method found
skip test.App.set_WeborbURL , special method found
skip test.dvds4less.Products.get_ProductId , special method found
skip test.dvds4less.Products.set_ProductId , special method found
skip test.dvds4less.Products.get_ProductName , special method found
skip test.dvds4less.Products.set_ProductName , special method found
skip test.dvds4less.Products.get_Product_desc_summary , special method found
skip test.dvds4less.Products.set_Product_desc_summary , special method found
skip test.dvds4less.Products.get_Product_desc , special method found
skip test.dvds4less.Products.set_Product_desc , special method found
skip test.dvds4less.Products.get_Product_price , special method found
skip test.dvds4less.Products.set_Product_price , special method found
skip test.dvds4less.Products.get_Image_path , special method found
skip test.dvds4less.Products.set_Image_path , special method found
skip test.dvds4less.Products.get_Rebates_file , special method found
skip test.dvds4less.Products.set_Rebates_file , special method found
skip test.dvds4less.Products.add_PropertyChanged , special method found
skip test.dvds4less.Products.remove_PropertyChanged , special method found
skip test.dvds4less.IntroCompletedEventArgs.get_Result , special method found
skip test.dvds4less.getProductInfoCompletedEventArgs.get_Result , special method
found
skip test.dvds4less.getProductInfoObjectCompletedEventArgs.get_Result , special
method found
skip test.dvds4less.getProductInfoXMLCompletedEventArgs.get_Result , special met
hod found
skip test.dvds4less.getRebatesInfoCompletedEventArgs.get_Result , special method
found
skip test.dvds4less.getSecurityTokenCompletedEventArgs.get_Result , special meth
od found
skip test.dvds4less.dvds4lessSoapClient.get_CookieContainer , special method fou
nd
skip test.dvds4less.dvds4lessSoapClient.set_CookieContainer , special method fou
nd
skip test.dvds4less.dvds4lessSoapClient.add_IntroCompleted , special method foun

```

Figure 2.1 Instrumenting test.xap

```
Administrator: C:\Windows\System32\cmd.exe
skip test.dvds4less.Products.get_Product_price , special method found
skip test.dvds4less.Products.set_Product_price , special method found
skip test.dvds4less.Products.get_Image_path , special method found
skip test.dvds4less.Products.set_Image_path , special method found
skip test.dvds4less.Products.get_Rebates_file , special method found
skip test.dvds4less.Products.set_Rebates_file , special method found
skip test.dvds4less.Products.add_PropertyChanged , special method found
skip test.dvds4less.Products.remove_PropertyChanged , special method found
skip test.dvds4less.IntroCompletedEventArgs.get_Result , special method found
skip test.dvds4less.getProductInfoCompletedEventArgs.get_Result , special method found
skip test.dvds4less.getProductInfoObjectCompletedEventArgs.get_Result , special method found
skip test.dvds4less.getProductInfoXMLCompletedEventArgs.get_Result , special method found
skip test.dvds4less.getRebatesInfoCompletedEventArgs.get_Result , special method found
skip test.dvds4less.getSecurityTokenCompletedEventArgs.get_Result , special method found
skip test.dvds4less.dvds4lessSoapClient.get_CookieContainer , special method found
skip test.dvds4less.dvds4lessSoapClient.set_CookieContainer , special method found
skip test.dvds4less.dvds4lessSoapClient.add_IntroCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.remove_IntroCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.add_getProductInfoCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.remove_getProductInfoCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.add_getProductInfoObjectCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.remove_getProductInfoObjectCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.add_getProductInfoXMLCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.remove_getProductInfoXMLCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.add_getRebatesInfoCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.remove_getRebatesInfoCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.add_getSecurityTokenCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.remove_getSecurityTokenCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.add_OpenCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.remove_OpenCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.add_CloseCompleted , special method found
skip test.dvds4less.dvds4lessSoapClient.remove_CloseCompleted , special method found
D:\ACTrace\test\test.dll patched successfully
C:\Program Files\AppCodeTrace>_
```

Figure 2.2 Instrumenting test.xap

Step 3: The file `test.dll` is now patched by AppCodeTrace. Add this patched compiled binary `test.dll` along with the other files that were extracted in step 1 to the archive `test.zip`. Rename the archive to `test.xap`. View this XAP file in a browser.

Step 4: Open DebugView.exe (runas Administrator). DebugView.exe may be downloaded from www.sysinternals.com

Step 5: Click anywhere on the Silverlight test application. All function calls captured by AppCodeTrace can be viewed in DebugView. See Figure 2.0

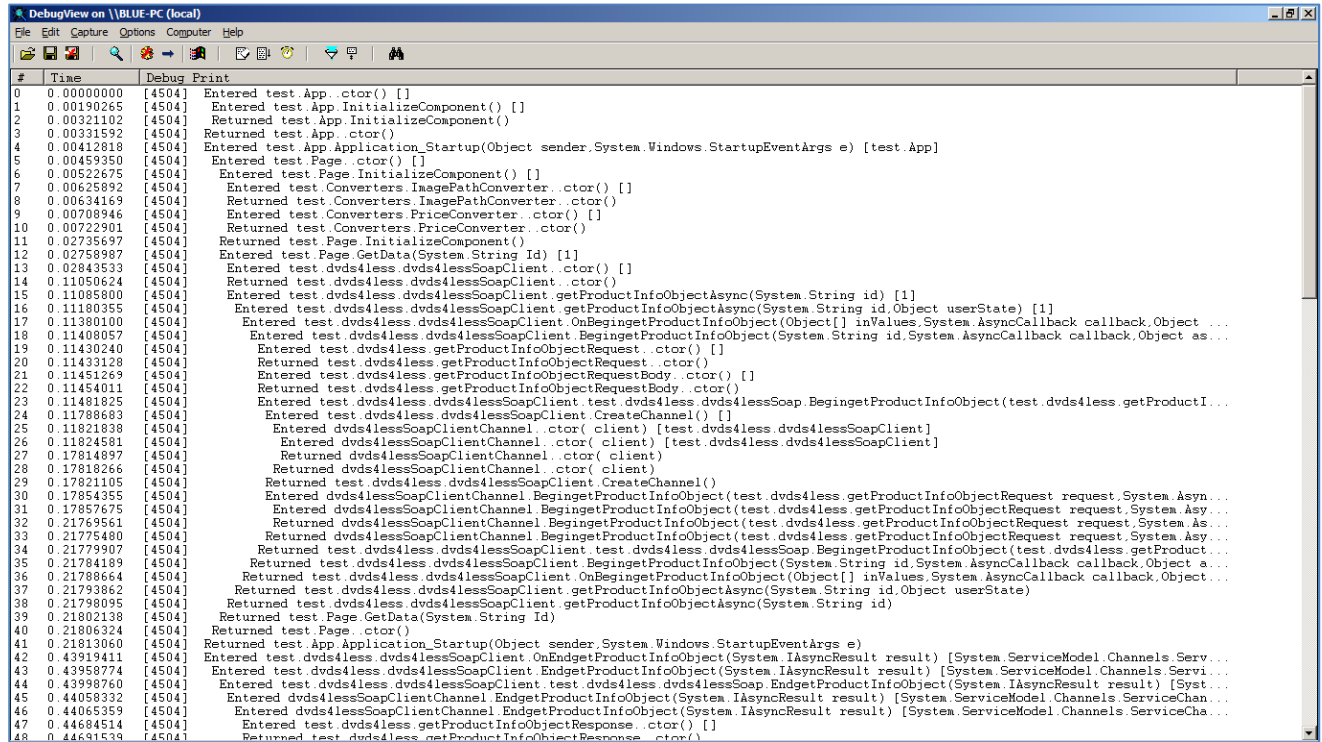


Figure 2.3 Capturing all Silverlight application calls

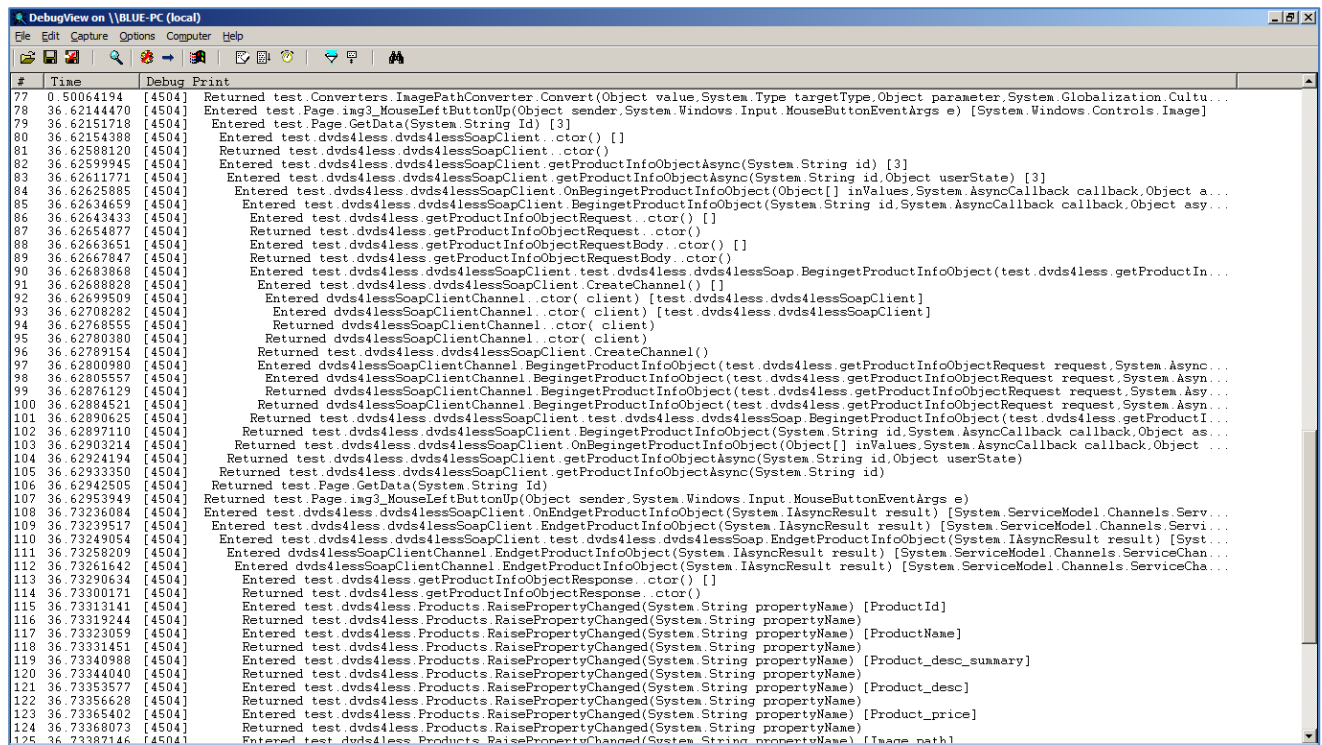


Figure 1.2 Clicking on any one of the videos from those on offer is captured in the logs, identified by the ID value

AppCodeTrace Exclusions

Function calls that are not required to be profiled may be excluded by using a regular expression pattern. For example, to exclude all function calls that begin with "_", the pattern will be written as follows:

```
<excludeCalls>
```

```
<pattern><![CDATA[^\_]]></pattern>
```

```
</excludeCalls>
```

AppCodeTrace.config explained

The configuration file is written in XML. Shown below is a sample file.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <backupPath>$executiondirectory$\backup</backupPath>
  <logPath>$executiondirectory$\logs</logPath>
  <traceLogDllPath>$executiondirectory$\TraceLogger.dll</traceLogDllPath>
  <traceSysCalls>0</traceSysCalls>
  <unidentifyFunctions>1</unidentifyFunctions>
  <traceFilter>1</traceFilter>
  <excludeCalls>
    <pattern><![CDATA[^\_]]></pattern>
    <pattern><![CDATA[^\<PrivateImplementationDetails>]]></pattern>
  </excludeCalls>
  <monitorCalls></monitorCalls>
</configuration>
```

XML Tag	Explanation
<backupPath>	location (absolute path) where the .NET Assemblies to be profiled, are copied
<logPath>	location of the log files that are created when AppCodeTraceCmd is run
<traceLogDllPath>	location of the Tracelogger DLL
<traceSysCalls>	reserved for future versions
<unidentifyFunctions>	A <i>boolean</i> value that takes either the value 1 (default) or the value 0: 1 indicates a .PDB file is not required for instrumentation 0 indicates a .PDB file is required for instrumentation (trace with symbols)
<traceFilter>	reserved for future versions and not used in the command-line version
<excludeCalls>	One or more function calls to be excluded from being profiled by AppCodeTrace.
<monitorCalls>	reserved for future versions