

MS Visual Studio 2005 Team Suite - Performance Tool

przygotował: Krzysztof Jurczuk
Politechnika Białostocka
Wydział Informatyki Katedra Oprogramowania
ul. Wiejska 45A
15-351 Białystok

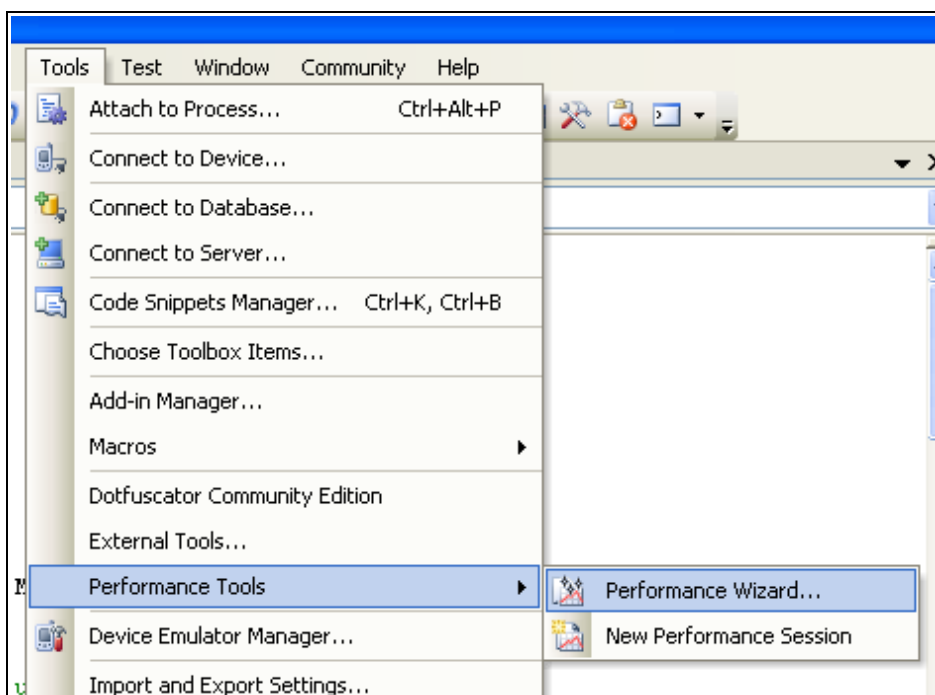
Streszczenie: Dokument zawiera podstawowe informacje na temat narzędzia do profilowania aplikacji, które jest składnikiem pakietu MS Visual Studio 2005 Team Suite.

1. Wprowadzanie

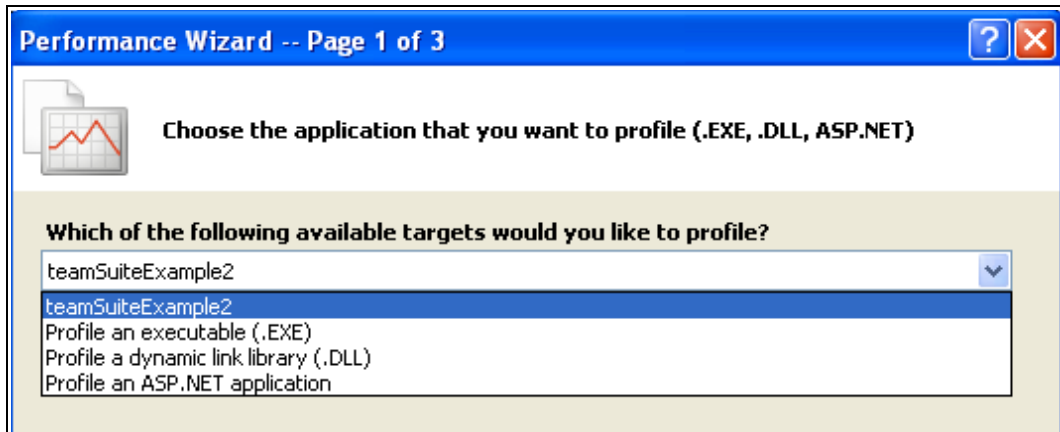
Profilowanie czasu wykonania aplikacji jest nieodłącznym elementem tworzenia zaawansowanych aplikacji. Głównym zadaniem narzędzi profilujących jest zbadanie testowanego oprogramowania pod kątem czasu poświęcanego na poszczególne linie kodu, funkcje czy też większe moduły. Pozwala to ukierunkować proces optymalizacji na fragmenty kodu, które zajmują najwięcej czasu wykonania. Dzięki temu tzw. wąskie gardła (ang., bottle neck) mogą zostać szybciej zidentyfikowane i zlikwidowane. Jednym z takich narzędzi jest moduł „performance tool” w pakiecie programistycznym Visual 2005 Team Suite. Umożliwia on profilowanie aplikacji stworzonych za pomocą kodu zarządzanego, niezarządzanego oraz aplikacji ASP.NET

2. Profilowanie aplikacji w środowisku .NET za pomocą „Performance tool”

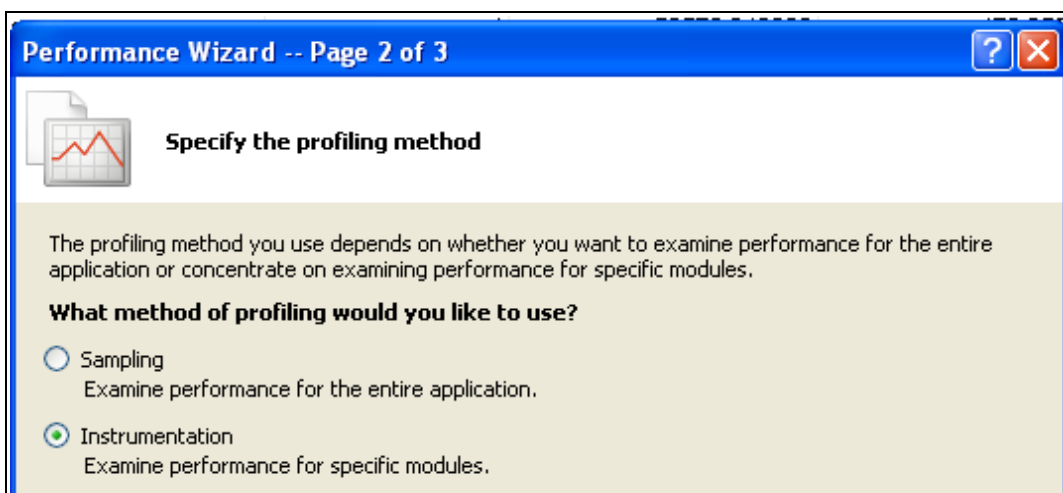
- aktywacja profilowania (dodanie sekcji „performance” do rozwiązania)



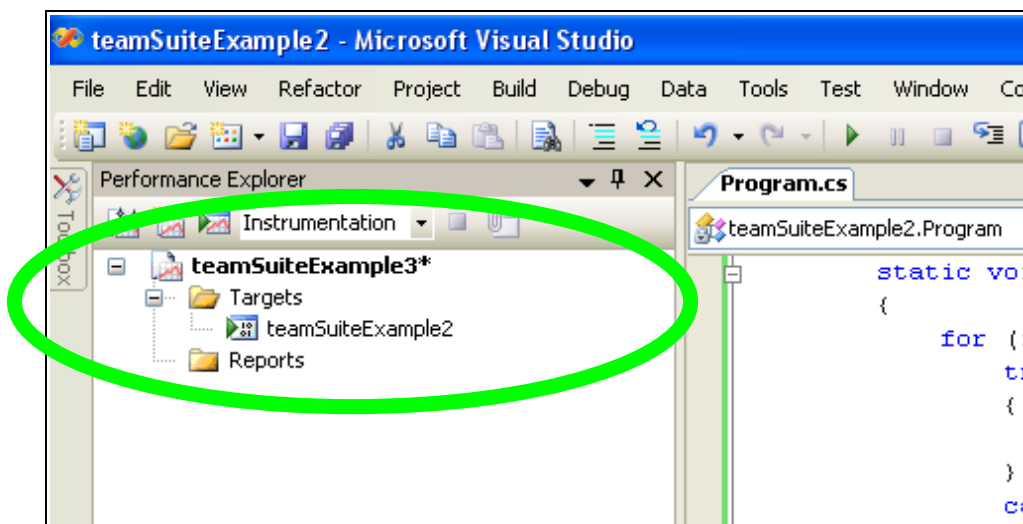
istnieje możliwość użycia kreatora sekcji „performance”



Wybór aplikacji.

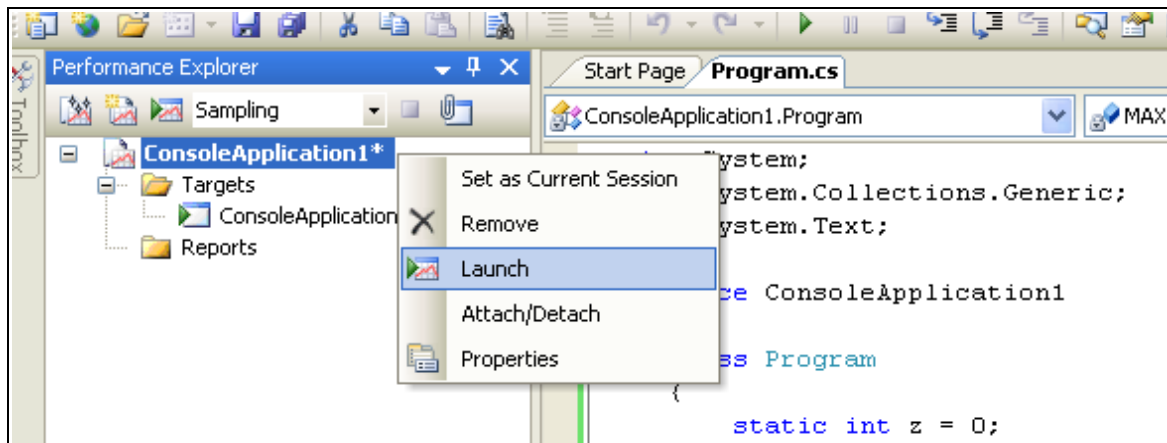


Wybór trybu profilowania.

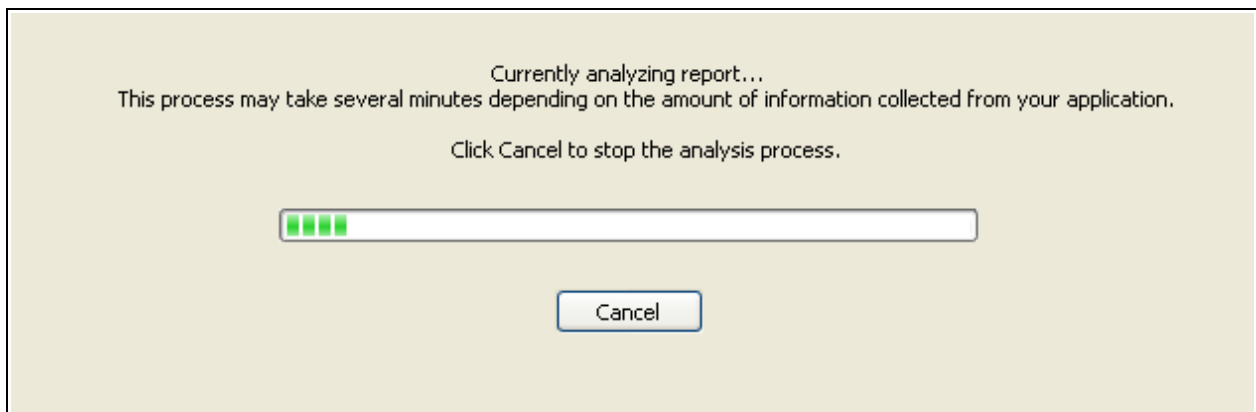


Nowo utworzona sekcja „performance”.

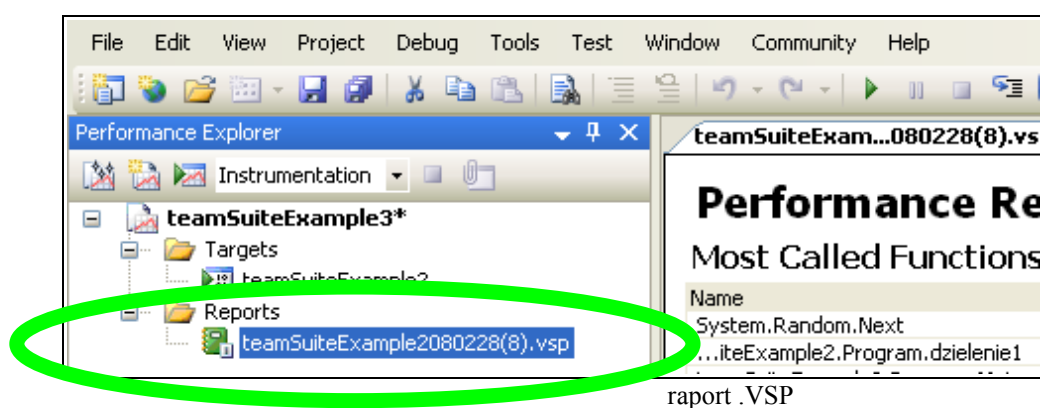
- uruchomienie profilowania



uruchomienie profilowania aplikacji (properties – szereg opcji profilowania).



tworzenie graficznej reprezentacji wyników (automatycznie po zakończeniu wykonani aplikacji).



raport .VSP

- analiza wyników (raport .VSP)

Performance Report Summary

Most Called Functions

Name	Number of Calls	%
System.Random.Next	2000000	100.000
...iteExample2.Program.dzielenie1	1	0.000
teamSuiteExample2.Program.Main	1	0.000

Functions With Most Individual Work

Name	Time (msecs)	%
System.Random.Next	732.151847	61.947
teamSuiteExample2.Program.Main	449.743026	38.053
...iteExample2.Program.dzielenie1	0.003804	0.000

Functions Taking Longest

Name	Time (msecs)	%
...iteExample2.Program.dzielenie1	53877.538025	96.866
System.Random.Next	983.732957	1.769
teamSuiteExample2.Program.Main	690.675896	1.242

Summary | Functions | Caller/Callee | Call Tree | Allocation | Objects Lifetime

Podsumowanie.

Function Name	Number of Calls	Elapsed Exclusive Time	Application Exclusive Time	Elapsed Inclusive Time	Application Inclusive Time
teamSuiteExample2.exe	4	54636.688199	449.746830	55620.533549	1181.898677
teamSuiteExample2.Program.Main(string[])	1	690.675896	449.743026	55619.783707	1181.898677
teamSuiteExample2.Program.dzielenie1()	1	53877.538025	0.003804	53877.538025	0.003804
teamSuiteExample2.Program.dzielenie2()	1	67.724436	0.000000	67.724436	0.000000
teamSuiteExample2.Program..cctor()	1	0.749842	0.000000	0.749842	0.000000
mscorlib.dll	2000001	983.845351	732.151847	983.845351	732.151847

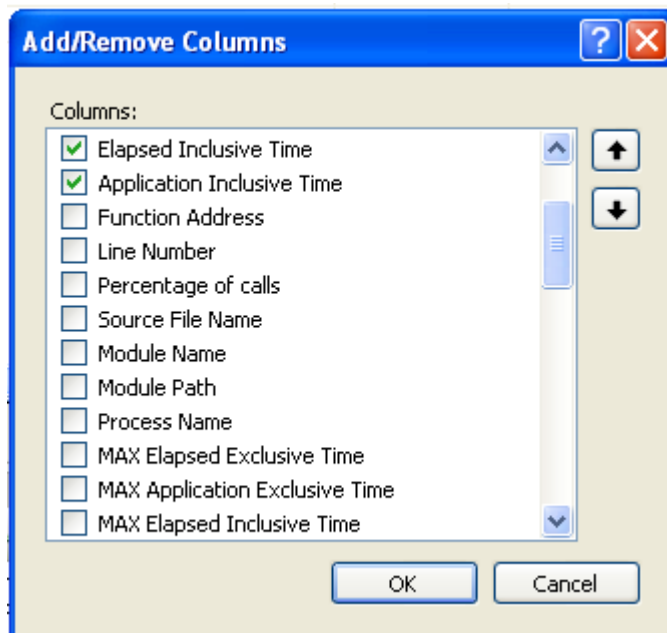
Summary | Functions | Caller/Callee | Call Tree | Allocation | Objects Lifetime

Zbiornicza informacja na temat modułów i funkcji.

- number of calls – liczba wywołań funkcji
- elapsed exclusive time – czas spędzony w funkcji oraz funkcjach z niej wywoływanych
- application exclusive time - czas spędzony w funkcji oraz funkcjach z niej wywoływanych (bez czasu w „kernel mode”)
- elapsed inclusive time – czas spędzony w funkcji
- application inclusive time - czas spędzony w funkcji (bez czasu w „kernel mode”)

Istnieje możliwość wyboru większej ilości statystyk (menu podręczne aktywne na tabeli z wynikami). Ich opis znajduje się na końcu dokumentu.

	Exclusive Time	A
	0.749842	
	690.675896	
1	0.112393	
2000000	983.732957	
1	53877.538025	
1	67.724436	



Function Name	Number of Calls	Elapsed Excl...	Application E...	Elapsed Inclu...	Application I...
Top of Stack	-	-	-	-	-
Current function:					
teamSuiteExample2.Program.Main	1	690.675896	449.743026	55619.783707	1181.898677
Functions that were called by teamSuiteExample2.Program.Main					
teamSuiteExample2.Program.dzielenie1()	1	53877.538025	0.003804	53877.538025	0.003804
System.Random.Next(int32,int32)	2000000	983.732957	732.151847	983.732957	732.151847
teamSuiteExample2.Program.dzielenie2()	1	67.724436	0.000000	67.724436	0.000000
System.Random..ctor()	1	0.112393	0.000000	0.112393	0.000000

Szczegółowe informacje na temat wybranej funkcji w zakładce „functions”.

Type/Allocating Function	Instances	Total Bytes Allocated	% of Total Bytes
System.SByte[]	500336	18012096	16.969
System.Reflection.TypeFilter	2	64	0.000
System.Text.EncoderReplacementFallt	2	32	0.000
System.Resources.RuntimeResourceS	1	36	0.000
System.Reflection.CustomAttributeTy	1	20	0.000
System.IO.BinaryReader	2	80	0.000
System.Runtime.Serialization.Format	1	40	0.000
System.Reflection.PropertyInfo[]	48	1372	0.001
System.Text.UTF8Encoding	2	56	0.000

Summary | Functions | Caller/Callee | Call Tree | **Allocation** | Objects Lifetime

Informacje o wykorzystaniu pamięci (aby były dostępne należy w opcjach profilowania ustawić flagę „Collect .NET object allocation information” w performance->general).

Class Name	Gen 0 Instances Collected	Gen 1 Instances Collected	Gen 2 Instances Collected	Large Object H
System.SByte[]	499526	0	0	
System.Reflection.TypeFilter	0	0	0	
System.Text.EncoderReplacement	1	0	0	
System.Resources.RuntimeResou	0	0	0	
System.Reflection.CustomAttribut	1	0	0	
System.IO.BinaryReader	1	0	0	
System.Runtime.Serialization.Forn	0	0	0	
System.Reflection.PropertyInfo[]	47	0	0	

Summary | Functions | Caller/Callee | Call Tree | Allocation | **Objects Lifetime**

informacje o „długości życia” obiektów (3 poziomy pamięci Garbage Collector'a, patrz dokument CLRProfiler) (aby były dostępne należy w opcjach profilowania ustawić flagę „Also collect .NET object lifetime information” w performance->general).

Przykładowe dostępne statystyki (MSDN):

Exclusive Allocations	Allocations in a function excluding all other allocations in subroutines that it called.
Inclusive Allocations	Allocations in a function including all allocations in subroutines that it called .
Exclusive Bytes Allocated	Bytes allocated in a function excluding all other byte allocations in subroutines that it called.
Inclusive Bytes Allocated	Bytes allocated in a function including all other byte allocations in subroutines that it called.
Exclusive Bytes Percent	(Bytes allocated in a function / total bytes) * 100
Inclusive Bytes Percent	(Bytes allocated in a function and all other subroutines that it called / total bytes) * 100

Number of Calls	The number of calls made to the instances of this function that were called by the parent function in the call tree.
Exclusive Bytes Percent	$(\text{Bytes allocated in a function} / \text{total bytes}) * 100$
Inclusive Bytes Percent	$(\text{Bytes allocated in a function and all other subroutines that it called} / \text{total bytes}) * 100$
Percentage of Calls	The number of instances of this function that were called by the parent function in the call tree as a percentage of the total number of calls to all functions in the specified data range.
Elapsed Exclusive Time	Calculated time for a function or summary item that includes samples that contain transition events, and excludes time from collection probes and the elapsed time of subroutines called by the function.
Application Exclusive Time	Calculated time for a function or summary item that excludes samples that contain transition events, time from collection probes, and the time of subroutines called by the function.
Elapsed Inclusive Time	Calculated time for a function or summary item that includes samples that contain transition events and time from subroutines called by the function, but excludes time from collection probes.
Application Inclusive Time	Calculated time for a function or summary item that excludes samples that contain transition events and time from collection probes, but includes the application time of subroutines called by the function.
MAX Elapsed Exclusive Time	The greatest elapsed exclusive clock time (processor cycles), or performance counter value for any one instance of this function that was called by the parent function in the call tree.
MAX Application Exclusive Time	The maximum single application exclusive clock time (processor cycles), or performance counter value for any one instance of this function that was called by the parent function in the call tree.
MAX Elapsed Inclusive Time	The greatest elapsed inclusive clock time (processor cycles), or performance counter value for any one instance of this function that was called by the parent function in the call tree.
MAX Application Inclusive Time	The greatest application inclusive clock time (processor cycles), or performance counter value for any one instance of this function that was called by the parent function in the call tree.
MIN Elapsed Exclusive Time	The least elapsed exclusive clock time (processor cycles), or performance counter value for any one instance of this function that was called by the parent function in the call tree.
MIN Application Exclusive Time	The minimum application exclusive clock time (processor cycles), or performance counter value for any one instance of this function that was called by the parent function in the call tree.
MIN Elapsed Inclusive Time	The least elapsed inclusive clock time (processor cycles), or performance counter value for any one instance of this function that was called by the parent function in the call tree.
MIN Application Inclusive Time	The least application inclusive clock time (processor cycles), or performance counter value for any one instance of this function that was called by the parent function in the call tree.

AVG Elapsed Exclusive Time	The average elapsed exclusive clock time (processor cycles), P6 performance or user-defined counter value for all instances of the function in the specified data range.
AVG Application Exclusive Time	The average application exclusive clock time (processor cycles), P6 performance or user-defined counter value for all instances of this function in the specified data range.
AVG Elapsed Inclusive Time	The average elapsed inclusive clock time (processor cycles), P6 performance or user-defined counter value for all instances of the function in the specified data range.
AVG Application Inclusive Time	The average application inclusive clock time (processor cycles), P6 performance or user-defined counter value for all instances of the function in the specified data range.
% Elapsed Exclusive Time	The percentage of the total global clock time (processor cycles), or performance counter values of the specified data segment that was spent in the elapsed exclusive time of all instances of the function in this context.
% Application Exclusive Time	The percentage of the total global clock time (processor cycles), or performance counter values of the specified data segment that was spent in the application exclusive time of all instances of the function in this context.
% Elapsed Inclusive Time	The percentage of the total global clock time (processor cycles), or performance counter values of the specified data segment that was spent in the elapsed inclusive time of all instances of the function in this context.
% Application Inclusive Time	The percentage of the total global clock time (processor cycles), or performance counter values of the specified data range that was spent in the inclusive time of all instances of the function in this context.
% Time Exclusive Probe Overhead	The percentage of the global clock time (processor cycles), or performance counter values for the specified data range that is attributed to profiling probes in the exclusive time of all instances of this function that were called by the parent function in the call tree.
% Time Inclusive Probe Overhead	The percentage of the global clock time (processor cycles), or performance counter values for the specified data range that is attributed to profiling probes in the inclusive time of all instances of this function that were called by the parent function in the call tree.
Exclusive Transitions	The number of transition events that occurred in all instance of this function, excluding functions called by the function.
Inclusive Transitions	The number of transition events that occurred in all instance of this function, including functions called by the function.
Exclusive Transitions Percentage	The percentage of transition events that occurred in the exclusive time of the instances of this function that were called by the parent function in the call tree compared to the total number of exclusive transitions in the data range.
Inclusive Transitions Percentage	The percentage of transition events that occurred in the inclusive time of the instances of this function that were called by the parent function in the call tree compared to the total number of inclusive transitions in the data range.
Root Node Recursive	Indicates if this function was directly or indirectly called recursively in this context.

Time Exclusive Probe Overhead	The total time attributed to profiling probes in the exclusive time of the instances of this function that were called by the parent function in the call tree.
The total time attributed to prof	Time Inclusive Probe Overheadiling probes in the inclusive time of the instances of this function that were called by the parent function in the call tree.
Function Name	Name of the function.
Function Address	Function address in hexadecimal format.
Inclusive Percent	$([\text{Inclusive samples for the function}] / [\text{Total inclusive samples taken during profiling}]) * 100$
Exclusive Percent	$([\text{Exclusive samples for the function}] / [\text{Total exclusive samples taken during profiling}]) * 100$
Inclusive Samples	Total number of performance data collected for the function including performance data for other functions that were called by it.
Exclusive Samples	Total number of performance data collected for the function excluding performance data for other functions that were called by it.
Line Number	Location of the start of the function in the source file.
Module Name	Name of the module that contains the function.
Module Path	Directory location of the module.
Process ID	Numeric identifier of the process.
Process Name	Name of the process.
Parent Function Address	Address in memory of the parent function that called another function.
Source File Name	Name of the source file that contains the function.
Unique ID	Hexadecimal number that identifies the function.
Unique Process ID	Unsigned integer to track the process.