CSS version 2.1

File Format

Table of Contents

[1. Introduction 3](#_Toc366492850)

[1.1. CPU Profiling 3](#_Toc366492851)

[1.2. Call-Stack Sampling 3](#_Toc366492852)

[2. Overview 4](#_Toc366492853)

[2.1. General Description 4](#_Toc366492854)

[2.2. Field Data Types 4](#_Toc366492855)

[3. File Format Layout 5](#_Toc366492856)

[3.1. Overall Layout 5](#_Toc366492857)

[3.2. Header 5](#_Toc366492858)

[Header Record Format 5](#_Toc366492859)

[3.3. Call-Stacks Array 6](#_Toc366492860)

[CallStack Record Format 6](#_Toc366492861)

[CallStack\_CallSite Record Format 6](#_Toc366492862)

[CallStack\_Event Record Format 6](#_Toc366492863)

[CallStack\_LeafNode Record Format 6](#_Toc366492864)

[3.4. Leaf Nodes Array 7](#_Toc366492865)

[LeafNode Record Format 7](#_Toc366492866)

[LeafNode\_CallStack Record Format 7](#_Toc366492867)

[LeafNode\_Event Record Format 7](#_Toc366492868)

[LeafNode\_EvThread Record Format 7](#_Toc366492869)

[LeafNode\_EvStack Record Format 8](#_Toc366492870)

[3.5. Call-Sites Array 8](#_Toc366492871)

[CallSite Record Format 8](#_Toc366492872)

[CallSite\_CallStack Record Format 8](#_Toc366492873)

[3.6. Functions Array 9](#_Toc366492874)

[Funtion Record Format 9](#_Toc366492875)

[3.7. Symbols Array 9](#_Toc366492876)

[Symbol Record Format 9](#_Toc366492877)

[3.8. Source Files Array 9](#_Toc366492878)

[SourceFile Record Format 9](#_Toc366492879)

[3.9. Modules Array 10](#_Toc366492880)

[Module Record Format 10](#_Toc366492881)

# Introduction

## CPU Profiling

AMD CodeXL has profiling suite that helps software developers to identify, investigate and improve the performance of applications, drivers and system software. It helps find time critical hotspots and diagnose performance issues, precisely with CPU sampling and call-graph profiling features.

AMD CodeXL may perform process-specific and system-wide profiling, and supports the analysis of both user applications and kernel-mode software. It provides three main profiling types:

* Time-based profiling (TBP)
* Event-based profiling (EBP)
* Instruction-based sampling (IBS)

**For further details please refer to the AMD CodeXL documentation.**

## Call-Stack Sampling

As part of the different CPU profiling types, the user may choose to enable **Call-Stack Sampling (CSS)**. This feature collects information about the call-stacks of the sampled functions, for the profiled process (in system-wide profiling, the CSS is activated only for the session’s base process – as if the profiling session ran without system-wide profiling enabled).

After the profiling has finished and the collected data has been processed, AMD CodeXL provides an interface for viewing the Call-Graph information traversed from the collection of sampled call-stacks.

# Overview

## General Description

The CSS (Call-Stack Sampling) file format is required for saving the CSS related data recorded in the CPU profiling session. This data is first gathered into different objects (**described in details in the Call Graph Traversal ADD**), and then these objects are serialized and written into the CSS file.

Without the CSS file, the CSS data would need to be recomputed from the *Profiling Database* (the recorded profiling raw-data) – an operation that usually takes a significant amount of time, in the better case, and it may even be lost, in the worst case.

## Field Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Size** | **Description** | |
| gtByte | 1 | Signed 8-bit integer [-128, 127] | |
| gtUbyte | 1 | Unsigned 8-bit integer [0, 255] | |
| char | 1 | ANSI String character | |
| wchar\_t | 2 | Windows | UTF-16 String character |
| 4 | Linux | UTF-32 String character |
| short | 2 | Signed 16-bit integer [-32768, 32767] | |
| gtInt16 |
| unsigned short | 2 | Unsigned 16-bit integer [0, 65535] | |
| gtUInt16 |
| int | 4 | Signed 32-bit integer [-2147483648, 2147483647] | |
| gtInt32 |
| unsigned int | 4 | Unsigned 32-bit integer [0, 4294967295] | |
| gtUInt32 |
| gtInt64 | 8 | Signed 64-bit integer [-9223372036854775808, 9223372036854775807] | |
| gtUInt64 | 8 | Unsigned 64-bit integer [0, 18446744073709551615] | |
| bool | 1 | Boolean value (true = 1, false = 0) | |
| gtVAddr | 8 | Virtual Address | |
| gtRVAddr | 4 | Relative Virtual Address (relative to an image base address) | |
| ProcessIdType | 4 | An OS’s Process identification number | |
| ThreadIdType | 4 | An OS’s Thread identification number | |
| EventMaskType | 4 | A Profiling Event identification number | |

# File Format Layout

## Overall Layout

|  |
| --- |
| Header |
| Call-Stacks Array |
| Leaf Nodes Array |
| Call-Sites Array |
| Functions Array |
| Symbols Array |
| Source Files Array |
| Modules Array |

## Header

#### Header Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Signatrue | gtUInt32 | FourCC identifying the file type [Must be FCC('CACS')] |
| Version.Minor | gtUInt16 | File format minor version number [Must be 1] |
| Version.Major | gtUInt16 | File format major version number [Must be 2] |
| ProcessId | gtUInt64 | The OS’s Process ID of the sampled process |
| ModulesArrayOffset | gtUInt32 | Offset, in bytes, into the file, of the *Modules Array* |
| ModulesCount | gtUInt32 | The number of modules in the *Modules Array* |
| CallStacksCount | gtUInt32 | The number of call-stacks in the *Call-Stacks Array* |
|  | gtUInt32 | Reserved |
| CharacterSetSize | gtUInt64 | The size, in bytes, of the Character Set used (char, wchar\_t, …) |

Let the macro FCC be defined as follows:

#define FCC(ch4) ((((gtUInt32)(ch4) & 0xFF) << 24) | \

(((gtUInt32)(ch4) & 0xFF00) << 8) | \

(((gtUInt32)(ch4) & 0xFF0000) >> 8) | \

(((gtUInt32)(ch4) & 0xFF000000) >> 24))

The Signature must be FCC('CACS'), equals 0x53434143 as a gtUInt32.

## Call-Stacks Array

Each *Call-Stack* is composed of an ordered list of call-sites (not including the leaf nodes), a list of leaf nodes, and a list of sampled events associated with it.

#### CallStack Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | gtUInt64 | Unique Identification number |
| CountSites | gtUInt64 | Number of call-sites |
| CallSites[CountSites] | CallStack\_CallSite | The call-sites composing this call-stack |
| CountEvents | gtUInt64 | Number of event types |
| Events[CountEvents] | CallStack\_Event | The sampled events for this call-stack |
| SelfTicks | gtUInt64 | Self ticks |
| TimesObserved | gtUInt64 | Number of times this call-stack was observed |
| CountLeaves | gtUInt64 | Number of Leaf Nodes |
| LeafNodes[CountLeaves] | CallStack\_LeafNode | The Leaf Nodes composing this call-stack |

#### CallStack\_CallSite Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ModuleBaseAddr | gtUInt64 | The base virtual address of the module in which the parent function of the call-site resides |
| Address | gtUInt64 | The call-site’s traversed address |
| ID | gtUInt64 | Unique Identification number |

#### CallStack\_Event Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | gtUInt64 | Unique Identification number |
| Count | gtUInt64 | The number of times this event was sampled |

#### CallStack\_LeafNode Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | gtUInt64 | Unique Identification number |

## Leaf Nodes Array

A *Leaf Node* is a call-site that was sampled while in the profiling session recording. This call-site is the beginning (top) of a call-stack.

**The array is preceded by a gtUInt64 of the number of elements within.**

#### LeafNode Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ModuleBaseAddr | gtUInt64 | The base virtual address of the module in which the parent function of the Leaf Node resides |
| Address | gtUInt64 | The Leaf Node’s traversed address |
| TimesObserved | gtUInt64 | Number of times this call-stack was observed |
| ID | gtUInt64 | Unique Identification number |
| FunctionID | gtUInt64 | Parent function ID |
| SourceLine | gtUInt64 | Associated source file’s line number |
| SourceFileID | gtUInt64 | Owning source file ID |
| CountStacks | gtUInt64 | Number of call-stacks |
| Stacks[CountStacks] | LeafNode\_Stack | The call-stacks associated with this Leaf Node |
| CountEvents | gtUInt64 | Number of event types |
| Events[CountEvents] | LeafNode\_Event | The sampled events for this Leaf Node |
| CountEvThreads | gtUInt64 | Number of threads sampled |
| EvThreads[CountEvThreads] | LeafNode\_EvThread | The thread specific sampling information |
| CountEvStacks | gtUInt64 | Number of aggregated call-stacks |
| EvStacks[CountEvStacks] | LeafNode\_EvStack | The call-stacks by aggregated by event type |

#### LeafNode\_CallStack Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | gtUInt64 | Unique Identification number |

#### LeafNode\_Event Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | gtUInt64 | Unique Identification number |
| Count | gtUInt64 | The number of times this event was sampled |

#### LeafNode\_EvThread Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| EventID | gtUInt64 | Event Identification number |
| Count | gtUInt64 | The sampled event’s count for this thread |
| ThreadID | gtUInt64 | The OS’s Thread ID |

#### LeafNode\_EvStack Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| StackID | gtUInt64 | Call-stack Identification number |
| EventID | gtUInt64 | Event Identification number |
| Count | gtUInt64 | The sampled event’s count for this thread |

## Call-Sites Array

A *Call-Site* is essentially a traversed address within a call-stack. The call-site may be associated with more than one call-stack.

**The array is preceded by a gtUInt64 of the number of elements within.**

#### CallSite Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ModuleBaseAddr | gtUInt64 | The base virtual address of the module in which the parent function of the call-site resides |
| Address | gtUInt64 | The call-site’s traversed address |
| ID | gtUInt64 | Unique Identification number |
| TimesObserved | gtUInt64 | Number of times this call-stack was observed |
| FunctionID | gtUInt64 | Parent function ID |
| SourceLine | gtUInt64 | Associated source file’s line number |
| SourceFileID | gtUInt64 | Owning source file ID |
| CountStacks | gtUInt64 | Number of call-stacks |
| Stacks[CountStacks] | CallSite\_Stack | The call-stacks associated with this call-site |

#### CallSite\_CallStack Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | gtUInt64 | Unique Identification number |

## Functions Array

This array lists the sampled functions information.

**The array is preceded by a gtUInt64 of the number of elements within.**

#### Function Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ModuleBaseAddr | gtUInt64 | The base virtual address of the module in which the function resides |
| Address | gtUInt64 | The function’s starting address |
| ID | gtUInt64 | Unique Identification number |
| SourceLine | gtUInt64 | Associated source file’s line number |
| SourceFileID | gtUInt64 | Owning source file ID |
| SymbolID | gtUInt64 | Matching symbols ID |

## Symbols Array

This array lists the symbols information of the matching the sampled function.

**The array is preceded by a gtUInt64 of the number of elements within.**

#### Symbol Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | gtUInt64 | Unique Identification number |
| Size | gtUInt64 | Size, in bytes, of the following Name |
| Name | Character-Set Type | The symbol’s name (zero terminated, and should be converted to the Character Set type declared in the file’s Header record) |

## Source Files Array

This array lists the source files’ information in which the sampled functions reside.

**The array is preceded by a gtUInt64 of the number of elements within.**

#### SourceFile Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ID | gtUInt64 | Unique Identification number |
| Size | gtUInt64 | Size, in bytes, of the following Name |
| Name | Character-Set Type | The source file’s name (zero terminated, and should be converted to the Character Set type declared in the file’s Header record) |

## Modules Array

This array lists the sampled modules information.

#### Module Record Format

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ModuleBaseAddr | gtUInt64 | The module’s base virtual address |
| Size | gtUInt64 | Size, in bytes, of the following Name |
| Name | Character-Set Type | The module’s file name (zero terminated, and should be converted to the Character Set type declared in the file’s Header record) |