DiffAnalyst User's Manual

DiffAnalyst Software
for Linux®

Publication: 45919

Revision 1.0

Publication date : March 4, 2010

Disclaimers

The contents of this documentation are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication. The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Trademarks

AMD, the AMD Arrow logo, AMD Athlon, AMD Opteron, and combinations thereof, and 3DNow! are trademarks of Advanced Micro Devices, Inc.

HyperTransport is a licensed trademark of the HyperTransport Technology Consortium.

Linux is a registered trademark of Linus Torvalds.

Microsoft, Windows, and Windows Vista are registered trademarks of Microsoft Corporation.

MMX is a trademark of Intel Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Table of Contents

List of Figures

Chapter 1. Introduction

1.1. Introduction

The AMD DiffAnalyst software is a profile differential (diff) tool used in conjunction with the AMD CodeAnalyst tool. The AMD DiffAnalyst tool compares any two profiles collected by using CodeAnalyst. The AMD DiffAnalyst tool is distributed as part of the CodeAnalyst performance analyst and tuning suite. DiffAnalyst is designed to help identify performance differences of any two binaries (i.e., executables or libraries). Users can compare performance data starting from the module level all the way down into each function and disassembly instruction.

Examples of cases for using the tool are:

- Compare performance from multiple profile runs of an application—The tool can help set up application behavior under different inputs, variables, or environment settings. An example of such a case is a scalability study where the application runs using various problem sizes to determine the upper limit of where performance starts to diminish. Profile comparison can help discover any limiting factors.

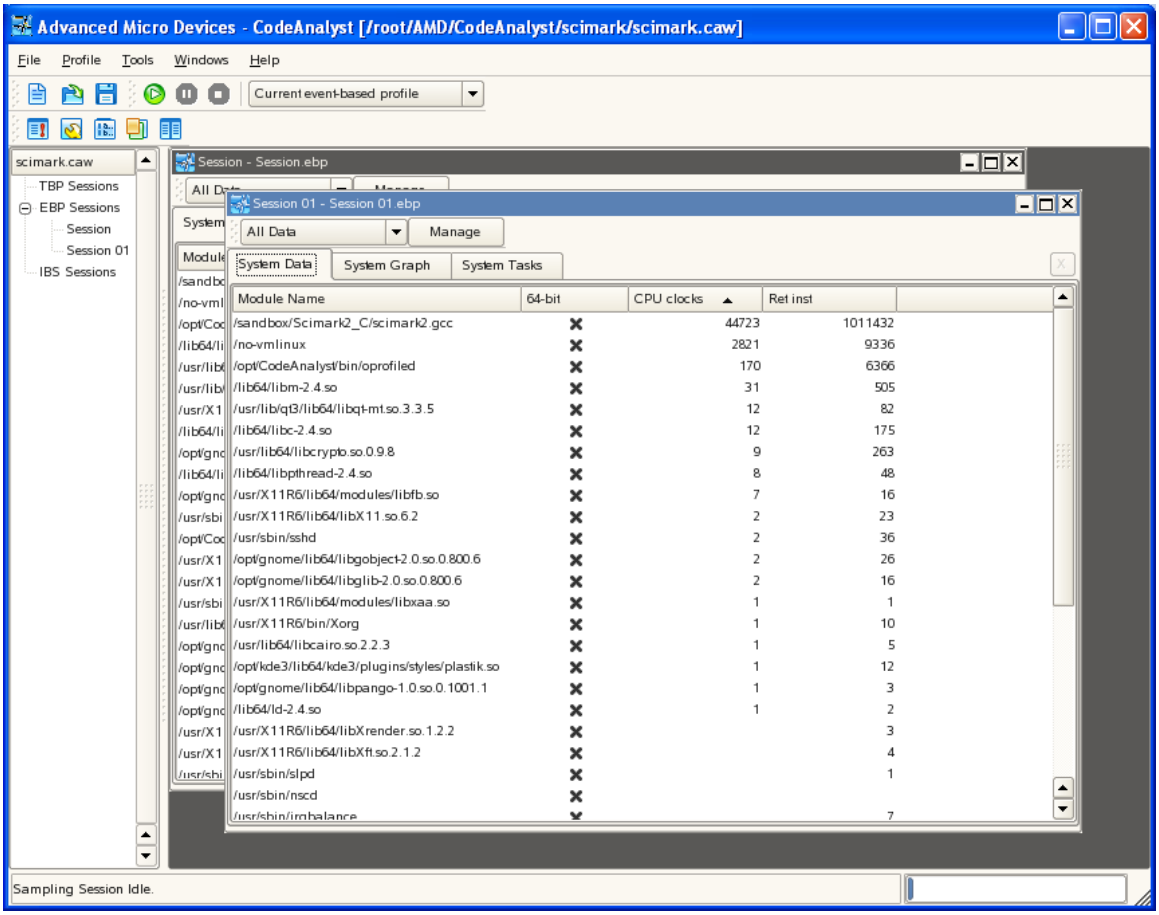
Another example is tuning an application where input is fixed and the application runs under different variables or different environment settings. Users can set up experiments, collect profiles, and then compare the profile data to reveal any interesting behaviors.

- Compare performance of various binaries of an application that are generated differently (i.e., different compilers or compiler options)—The tool can help determine how the binaries generated by different compilers may behave differently, or how compiler options might improve or worsen the overall performance of an application.

1.2. Overview of the AMD DiffAnalyst Tool

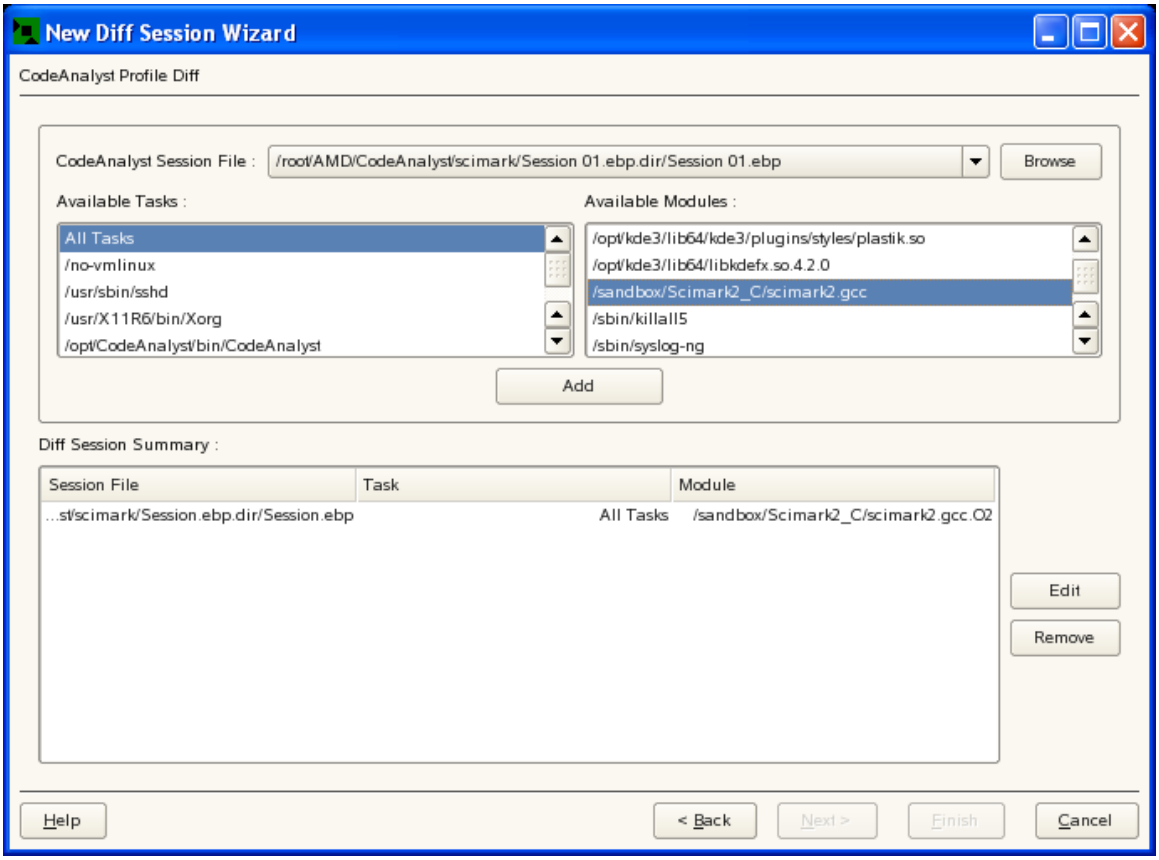
The AMD DiffAnalyst tool compares TBP or EBP files generated by the AMD CodeAnalyst tool. These files contain system-wide profiling data (known as “profiling sessions”) from a time-based, event-based, or instruction-based profile generated by the AMD CodeAnalyst tool. In DiffAnalyst, users begin by selecting any two profiling sessions. The following example shows two .ebp (event-based) profiling sessions.

Figure 1.1. Two Event-Based Profiling Sessions



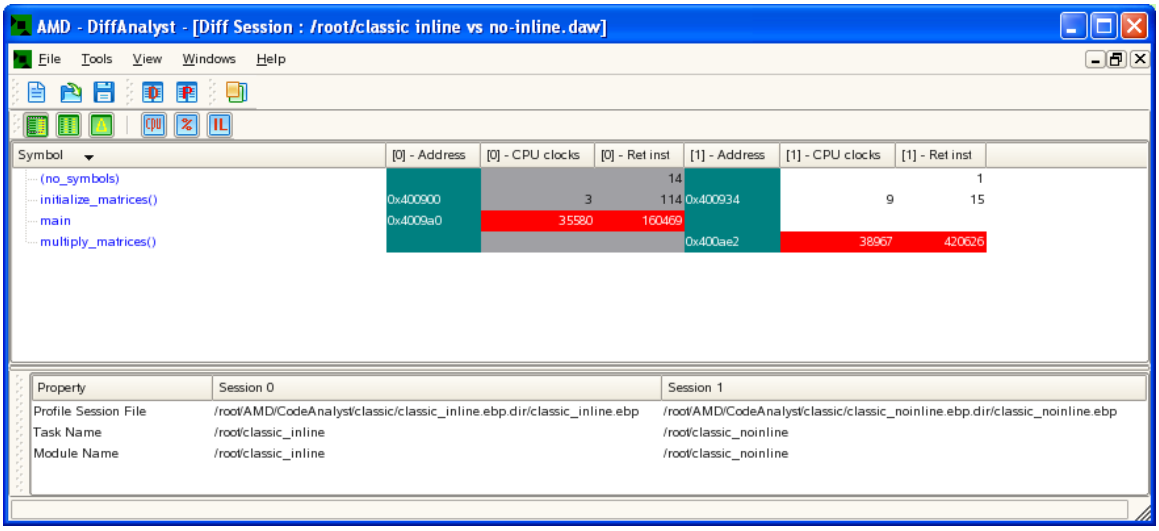
A system-wide profile generally consists of modules (i.e., an executable and various shared libraries). A TBP/EBP file stores profiling data of these modules during a profiling session. The AMD DiffAnalyst tool allows users to select any two modules and compares them based on the module's **symbol** information. Typically, a **symbol** is a compiler-generated equivalent of a function in C/C++ programs.

Figure 1.2. New Diff Session Wizard



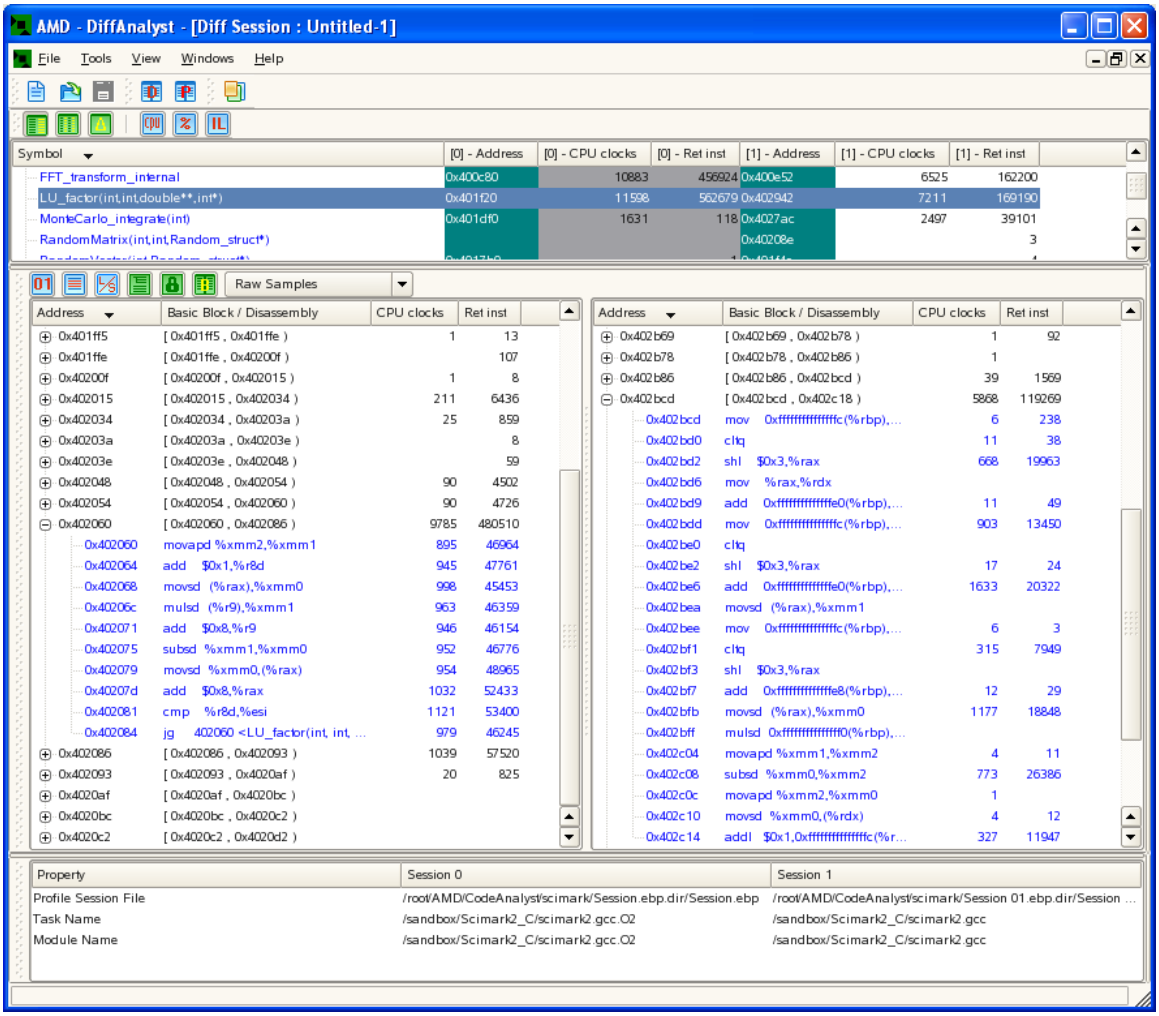
In the AMD DiffAnalyst Symbol Diff view, the tool compares profile data from different modules based on the fully-qualified function name. For each function, DiffAnalyst presents two sets of data, each from the selected modules. The user can choose different ways to view the data, which is discussed in Symbol Diff View [exploring_workspace_gui.dita].

Figure 1.3. DiffAnalyst Symbol Diff view



After examining the module, comparing the disassembly of any particular function can be done in the Disassembly Diff View. This view displays disassembly instructions, basic block, identify load/store instructions, and identifies in-line instances.

Figure 1.4. Disassembly Diff View



Chapter 2. Features

2.1. Exploring the Workspace and GUI

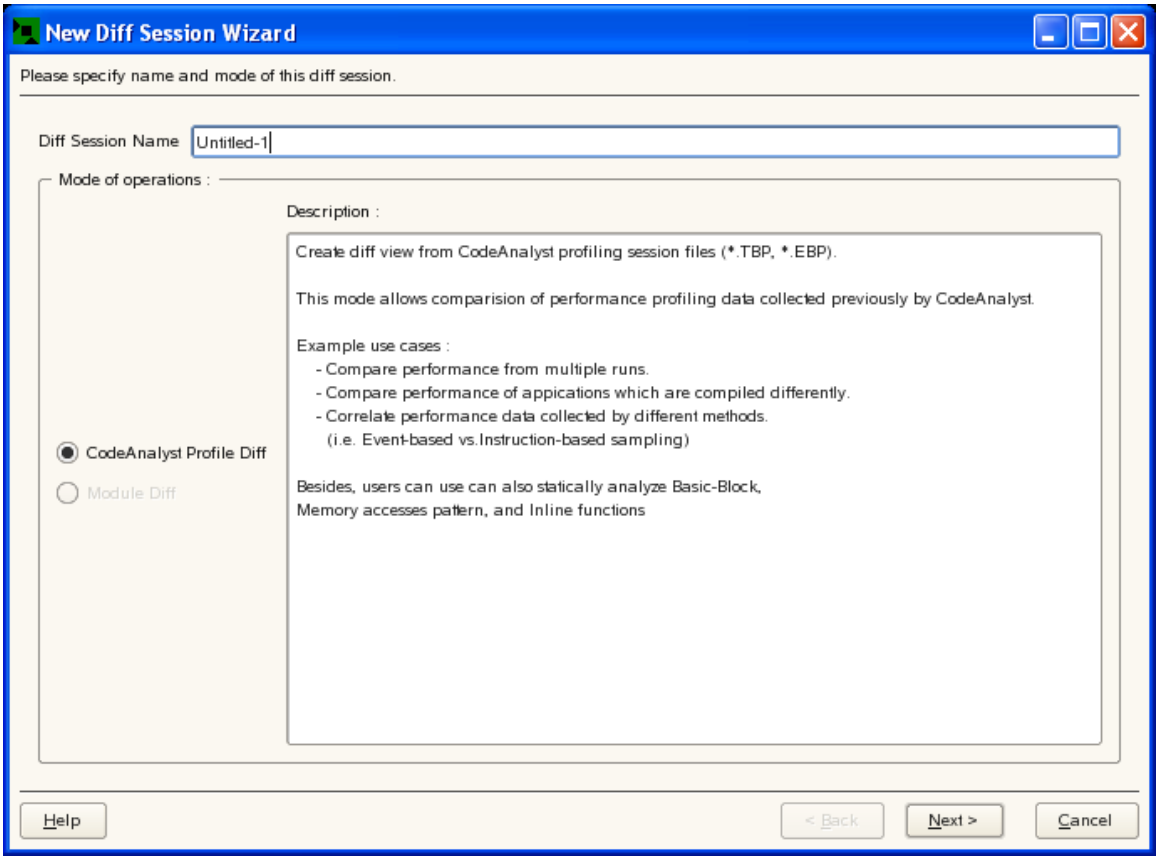
2.1.1. New Diff Session Wizard and Dialog

The New Diff Session Wizard is the first step in generating a session. The wizard can be accessed by one of the following steps:

- On the DiffAnalyst menu bar select **File > New**.
- On the DiffAnalyst menu bar select New Tool icon.
- On the CodeAnalyst toolbar select Create diff session icon.
- On the CodeAnalyst project navigator, right-click on **Session** and select **Create diff session**.

Once the New Diff Session Wizard is open, create a Diff session name. This name identifies each diff session because multiple diff sessions can be viewed at a time.

Figure 2.1. New Diff Session Wizard

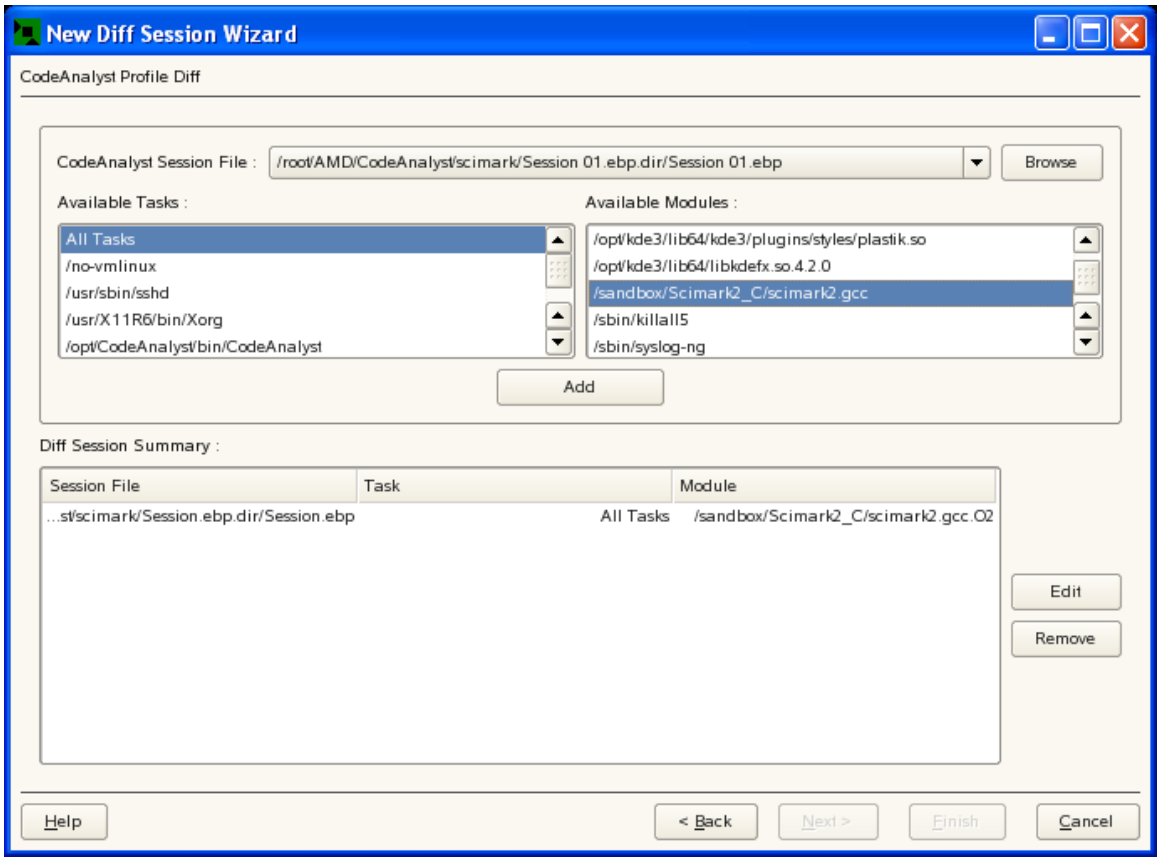


Next, specify a mode of operation. A mode description displays when the mode is selected. Available modes are:

- **CodeAnalyst Profile Diff Mode:** This mode compares any CodeAnalyst profiling output (. tbp or . ebp file). Use the **Browse** button to open a file navigation dialog to select a . tbp or . ebp file.

Once a . tbp or . ebp file is selected, a list of available tasks (application processes) displays. Use the list to choose a particular task or all tasks. The list of available modules update accordingly. When done, click **Add** to add the current selection to the Diff Session Summary list. Use **Edit** or **Remove** to manipulate the list. Once two profiling sessions are selected, click **Finish** to generate the specified diff session.

Figure 2.2. CodeAnalyst Profile Diff

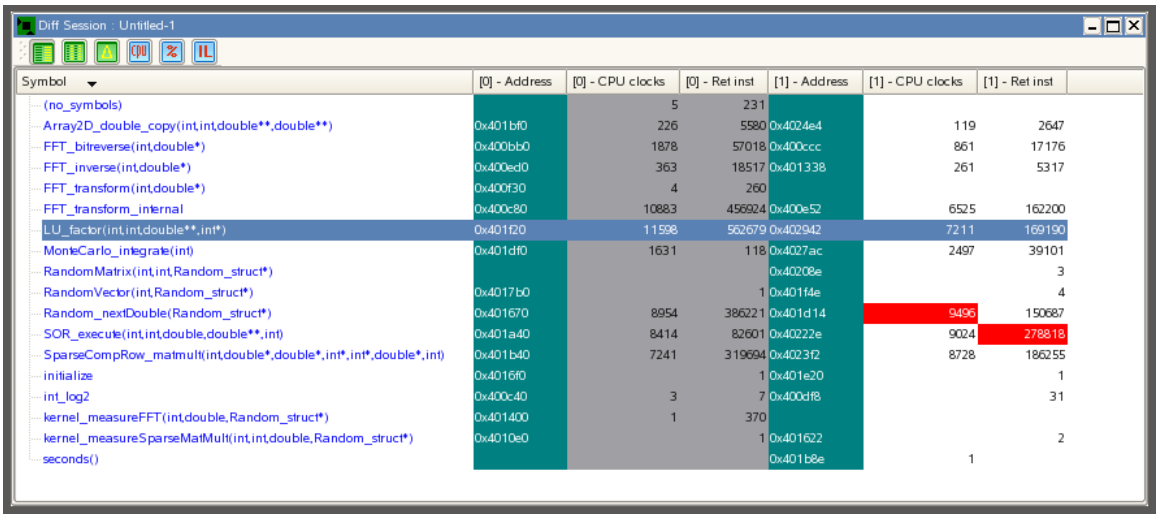


- **Module Diff Mode:** This mode compares any binary files without profiling data. (Not yet available)

2.1.2. Symbol Diff View

Symbol Diff View displays functions using their symbol names. Names display in the “Symbols” column when the function contains samples. The following figure shows an example of Symbol Diff view.

Figure 2.3. Symbol Diff View



Symbol	[0] - Address	[0] - CPU clocks	[0] - Ret inst	[1] - Address	[1] - CPU clocks	[1] - Ret inst
(no_symbols)		5	231			
Array2D_double_copy(int,int,double**,double**)	0x401bf0	226	5580	0x4024e4	119	2647
FFT_bitreverse(int,double*)	0x400bb0	1878	57018	0x400ccc	861	17176
FFT_inverse(int,double*)	0x400ed0	363	18517	0x401338	261	5317
FFT_transform(int,double*)	0x400f30	4	260			
FFT_transform_internal	0x400c80	10883	456924	0x400e52	6525	162200
LU_factor(int,int,double**,int*)	0x401f20	11598	562679	0x402942	7211	169190
MonteCarlo_integrate(int)	0x401df0	1631	118	0x4027ac	2497	39101
RandomMatrix(int,int,Random_struct*)				0x40208e		3
RandomVector(int,Random_struct*)	0x4017b0		1	0x401f4e		4
Random_nextDouble(Random_struct*)	0x401670	8954	386221	0x401d14	9496	150687
SOR_execute(int,int,double,double**,int)	0x401a40	8414	82601	0x40222e	9024	278818
SparseCompRow_matmult(int,double*,double*,int*,int*,double*,int)	0x401b40	7241	319694	0x4023f2	8728	186255
initialize	0x4016f0		1	0x401e20		1
int_log2	0x400c40	3	7	0x400df8		31
kernel_measureFFT(int,double,Random_struct*)	0x401400	1	370			
kernel_measureSparseMatMult(int,int,double,Random_struct*)	0x4010e0		1	0x401622		2
seconds()				0x401b8e	1	

Each row represents a symbol. Each symbol has two address columns (white text with green background) showing symbol address associated with each module. Performance data sets are shown in two groups differentiated by color (black text with white background and black text with gray back ground).

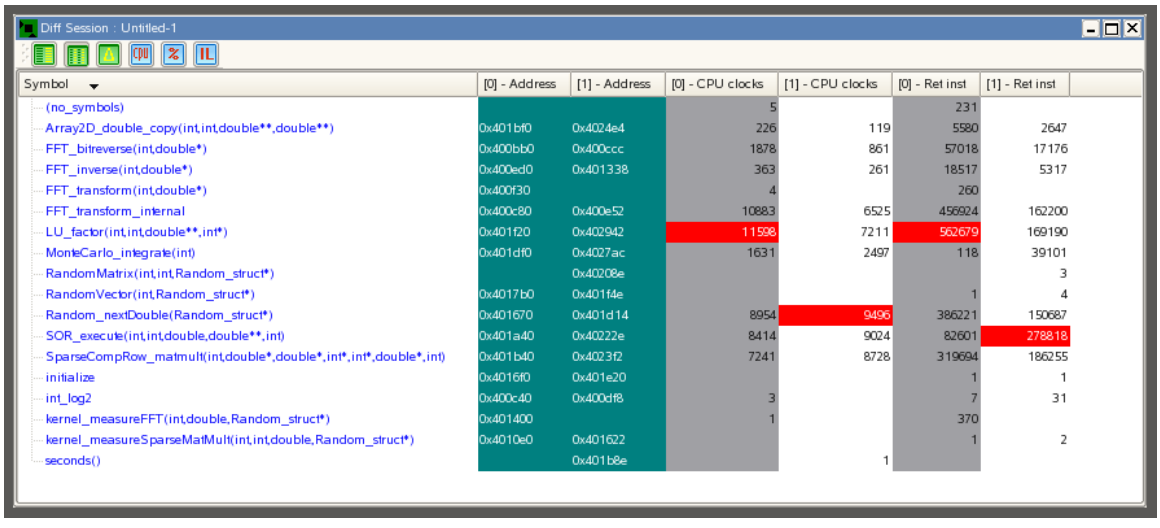
This view shows the symbol only if the function contains samples. For example, function main() does not contain any samples and is not shown here. Also, if a symbol is only present in one module, Symbol Diff View displays it but leaves the other module blank.

Cells with white text and red background contain the maximum value within the column, helping to identify hotspots of each performance event.

2.1.2.1. Symbol Diff View Toolbar

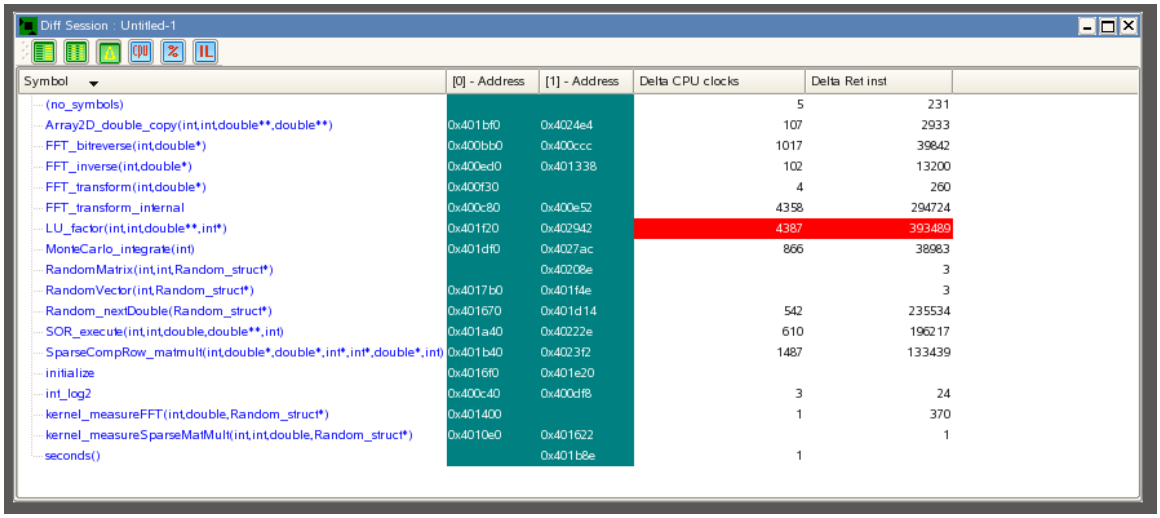
The Symbol Diff view toolbar contains the following icons and options for viewing:

- **Left-Right View:** This view groups performance data of each profiling session together.
- **Side-by-Side View:** This view groups each performance event from each profiling session together.



Symbol	[0] - Address	[1] - Address	[0] - CPU clocks	[1] - CPU clocks	[0] - Ret inst	[1] - Ret inst
(no_symbols)			5		231	
Array2D_double_copy(int,int,double**,double**)	0x401bf0	0x4024e4	226	119	5580	2647
FFT_bitreverse(int,double*)	0x400bb0	0x400ccc	1878	861	57018	17176
FFT_inverse(int,double*)	0x400ed0	0x401338	363	261	18517	5317
FFT_transform(int,double*)	0x400f30		4		260	
FFT_transform_internal	0x400c80	0x400e52	10883	6525	456924	162200
LU_factor(int,int,double**,int*)	0x401f20	0x402942	11598	7211	562679	169190
MonteCarlo_integrate(int)	0x401df0	0x4027ac	1631	2497	118	39101
RandomMatrix(int,int,Random_struct*)		0x40208e				3
RandomVector(int,Random_struct*)	0x4017b0	0x401f4e			1	4
Random_nextDouble(Random_struct*)	0x401670	0x401d14	8954	9496	386221	150687
SOR_execute(int,int,double,double**,int)	0x401a40	0x40222e	8414	9024	82601	278818
SparseCompRow_matmult(int,double*,double*,int*,int*,double*,int)	0x401b40	0x4023f2	7241	8728	319694	186255
initialize	0x4016f0	0x401e20			1	1
int_log2	0x400c40	0x400df8	3		7	31
kernel_measureFFT(int,double,Random_struct*)	0x401400		1		370	
kernel_measureSparseMatMult(int,int,double,Random_struct*)	0x4010e0	0x401622			1	2
seconds()		0x401b8e		1		

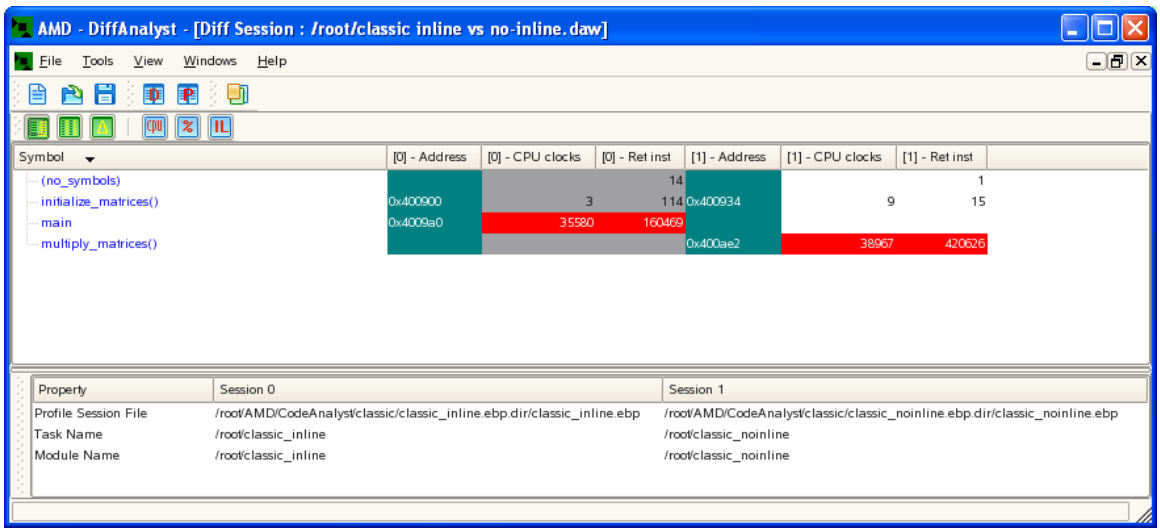
- **Delta View:** This view shows the delta value of each performance event from each profiling session.



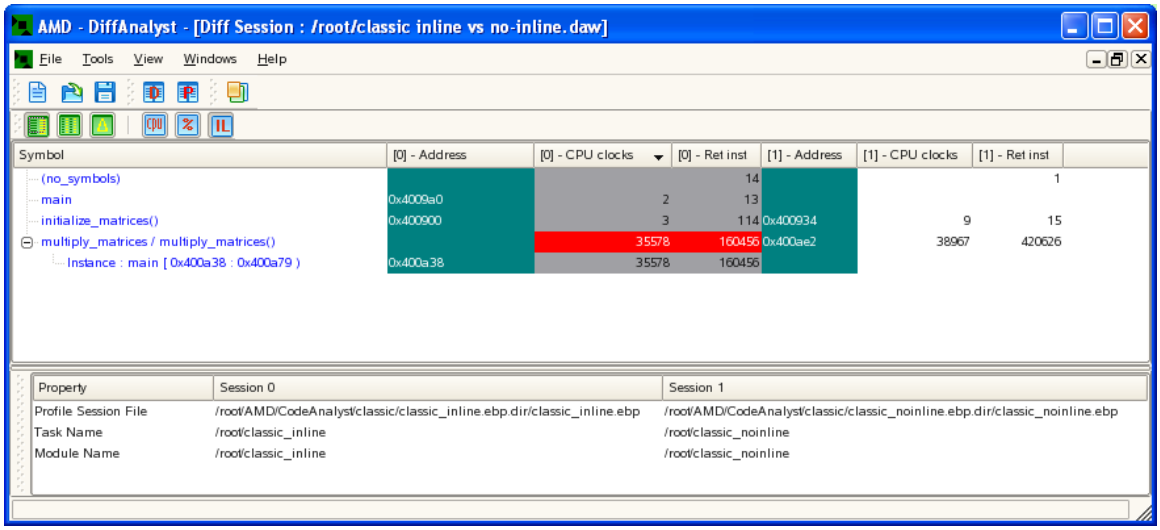
- **Separated by CPUs:** This option shows samples of performance events collected separately on each CPU.

- **Show Percentage:** This option shows sample percentage of each symbol with respect to the number of samples within this module for a particular performance event.

- **Aggregate Samples into In-line Function:** When the module contains in-line functions, this option aggregates samples into the original in-line function instead of the caller function.



In each in-line function, samples are separated into a different in-line instance. In session 1 (right) of above figure, multiply_ matrices () is declared as in-line function, and called by main(). Therefore, the samples that belong to the function multiply_ matrices () becomes part of main().



In session 1 (right) of above figure, the last item in the list shows the in-line instance of function multiply_ matrices () inside main().

2.1.2.2. Merge Symbol

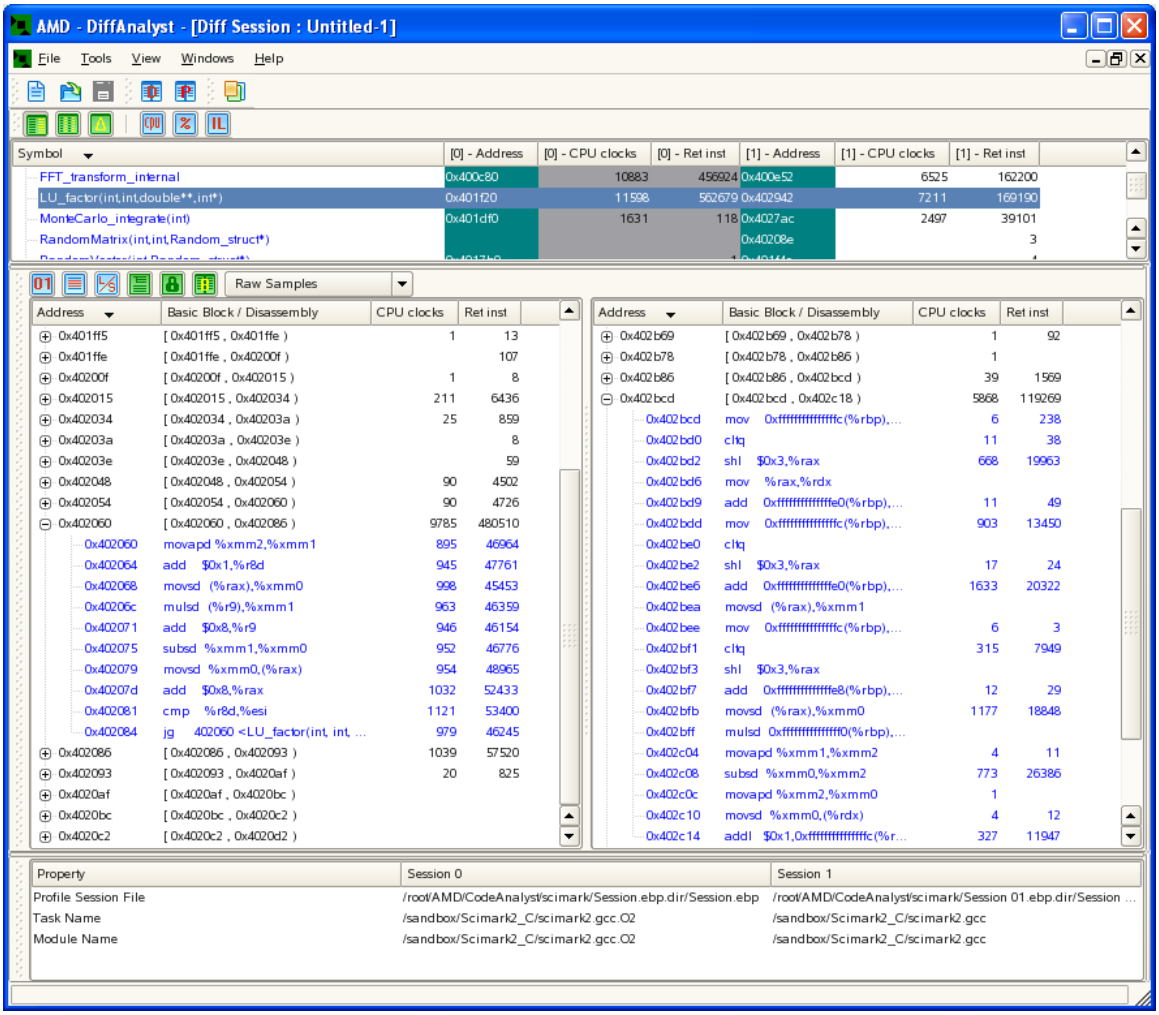
Different compilers usually have different symbol naming conventions. When compiling an application with different compilers, output binaries often contain functions with mismatched symbol names. This mismatching affects the Symbol Diff view because the tool tries to match symbols using the fully-qualified symbol name. In this case, a single symbol may be shown as two different symbols.

The Symbol Diff view tries to account for most of these scenarios in its symbol name-matching algorithm. However, if this problem continues, the user can manually merge the entries of the symbols. Select any two symbols, right-click, and select Merge symbol data. The data of selected symbols merge into one entry in the Symbol Diff view.

2.1.3. Disassembly Diff View

The DiffAnalyst disassembly diff view is a docked window consisting of two separate lists for viewing disassembly of the function pair. To open to this view, double-click any entry in Symbol Diff view. This docked window can be moved or undocked. It can also be hidden by closing (**X**) or clear the check box for the option under **Tools > Show Dasm Diff View**.

Figure 2.4. Disassembly Diff View



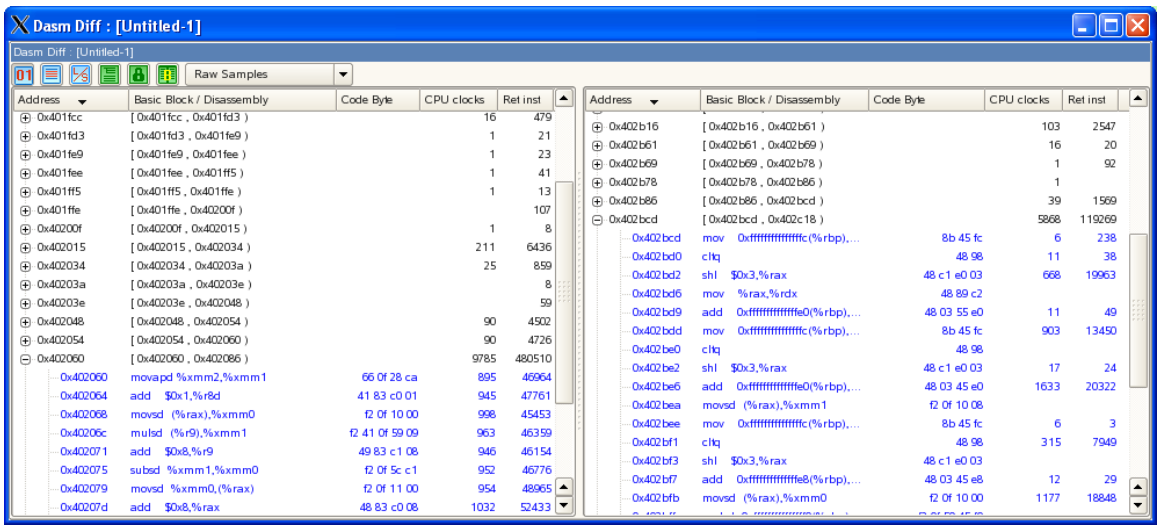
The first level of each list contains basic block information sorted by address (in black text with white background). The Basic Block / Disassembly column displays a section of code that represents a serialized execution path, which does not contain any kind of control transfer instruction (i.e., jump or call.). The beginning of a basic block is usually the destination of a single/multiple control transfer instruction and the ending is a control transfer instruction.

Each basic block is denoted using an address range notation. For example, “[0x400875, 0x4008c7)” means the basic block starts from address 0x400874 to 0x4008c6. Each basic block expands to reveal the contained disassembly instructions (displayed in blue text with white background).

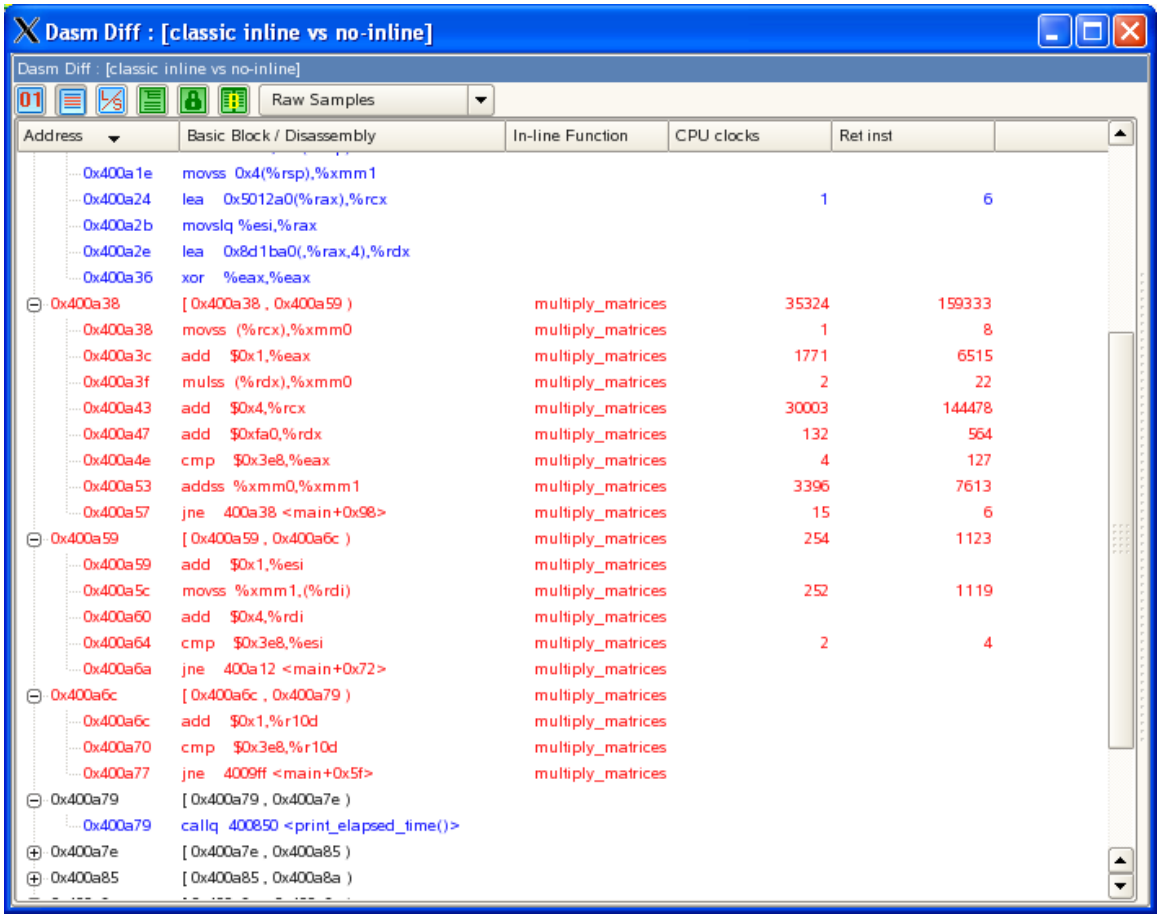
2.1.3.1. Disassembly Diff View Toolbar

This option displays the Disassembly Diff View toolbar which contains various options for the view. Available options are:

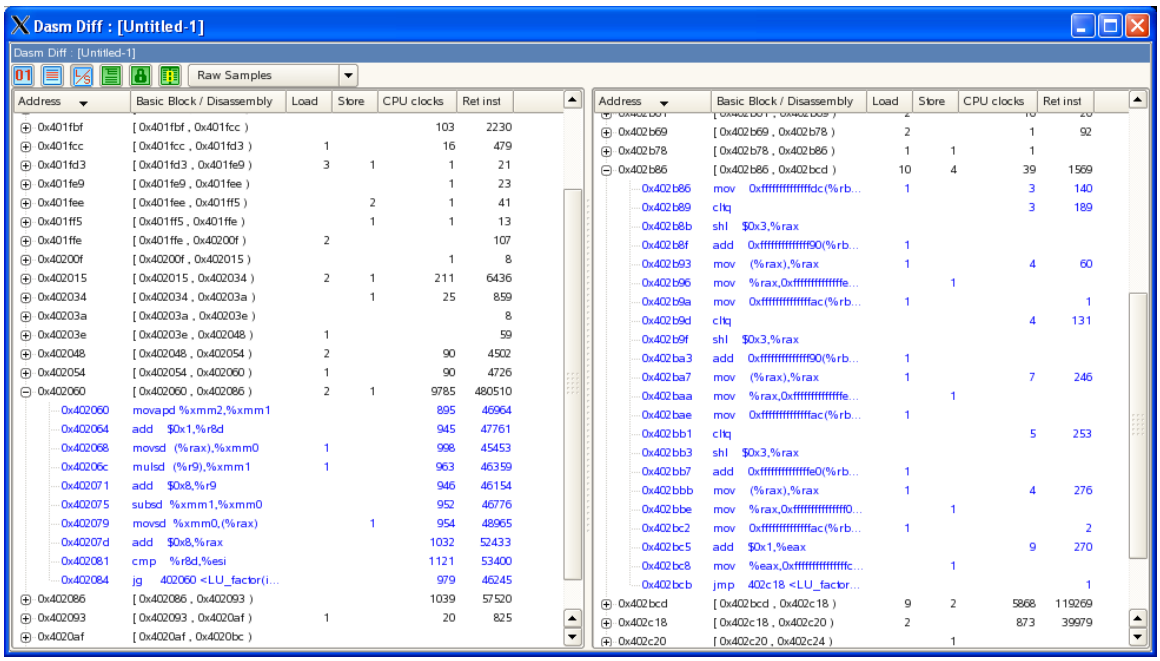
- **Show/Hide Code Byte:** Toggles to show or hide a code byte column.



- **Show In-line Instance:** Toggles to show or hide the in-line function name column. In-line instances are shown in red text with white background.



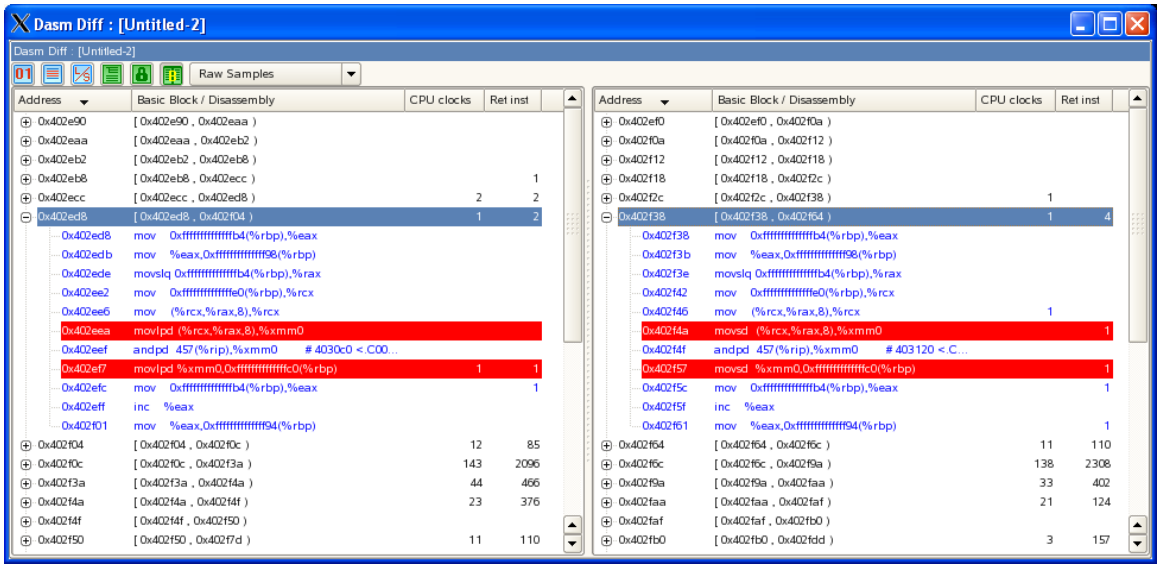
- **Show Load/Store Statistics:** Toggles to show or hide the load or store statistics column.



- **Expand/Collapse All Basic Block:** Expands or collapses all basic blocks in both disassembly lists.

- **Synchronize Disassembly View:** Allows for synchronizing (locking) scrolling actions. If left and right basic blocks are aligned, expand/collapse actions are also synchronized.

- **Highlight Difference within Basic Block:** Compares disassembly instructions within the selected basic blocks in each list.



- **Show Percentage Combo Box:** Combo box selection of what type of information displays in each performance event column. The available options are:

  - Raw Samples—Shows the amount of samples for each performance event.

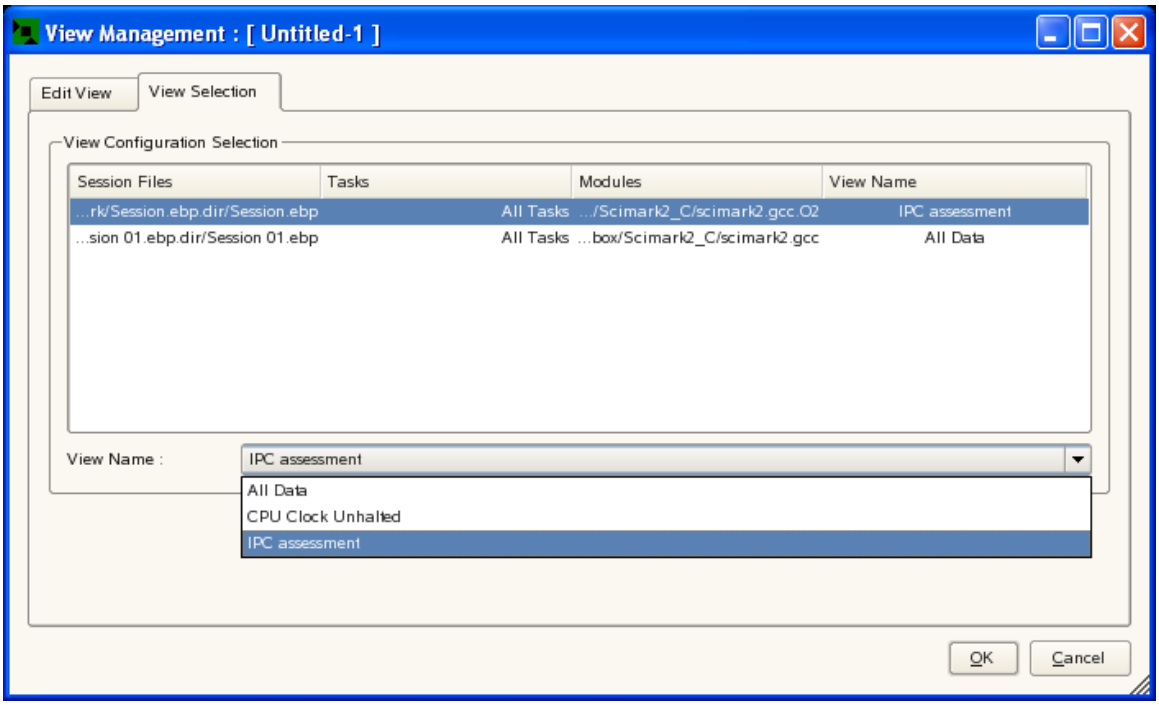
- Function Percentage—Shows the percentage of samples for each performance event with respect to the function.
- Basic Block Percentage—Shows the percentage of samples for each performance event with respect to the basic block.

2.1.4. View Management Dialog

View Management dialog allows for customization of the performance events being shown. Each view is specific to hardware platforms and performance events available in each profiling session. A set of predefined views are provided for user's convenience.

The View Management dialog contains two tabs—Edit View and View Selection—as shown in the following example.

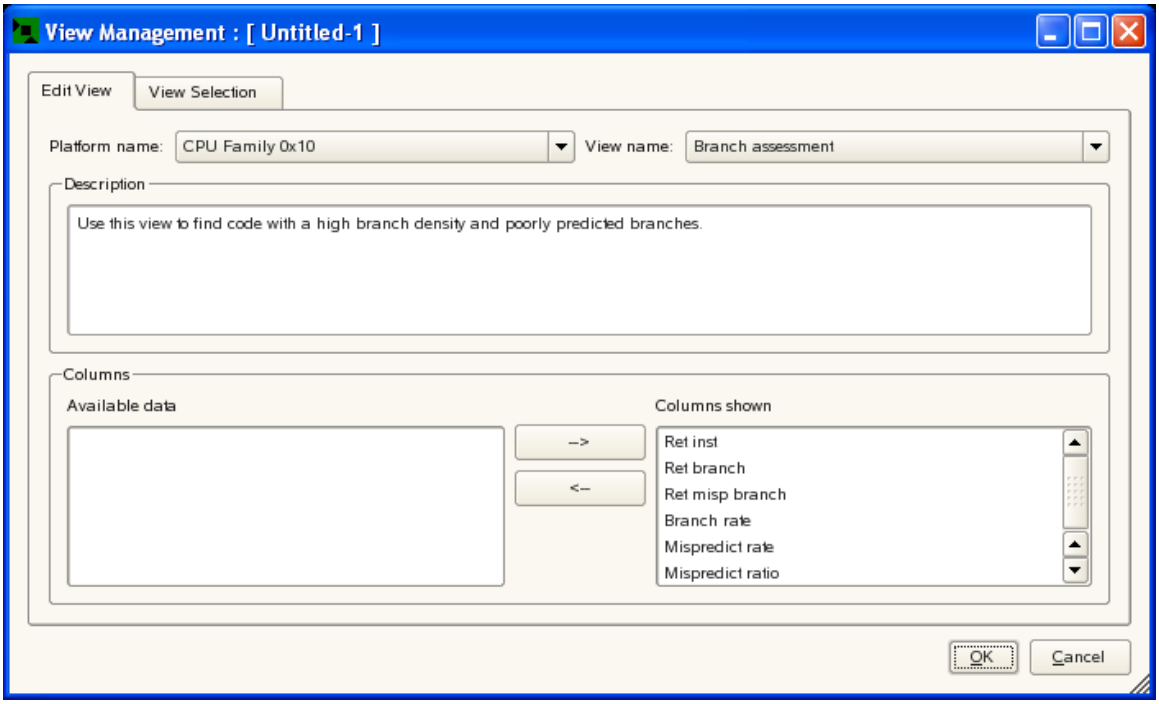
Figure 2.5. View Management Dialog



2.1.4.1. Edit View Tab

The Edit View tab allows users to customize the set of performance events to be shown in each view. Users begin by selecting a **Platform name**, which updates the **View name** combo box with available views for the selected platform. Users can use the right-arrow button to remove, and left-arrow button to add or delete performance events between the **Available data** and **Columns shown** lists.

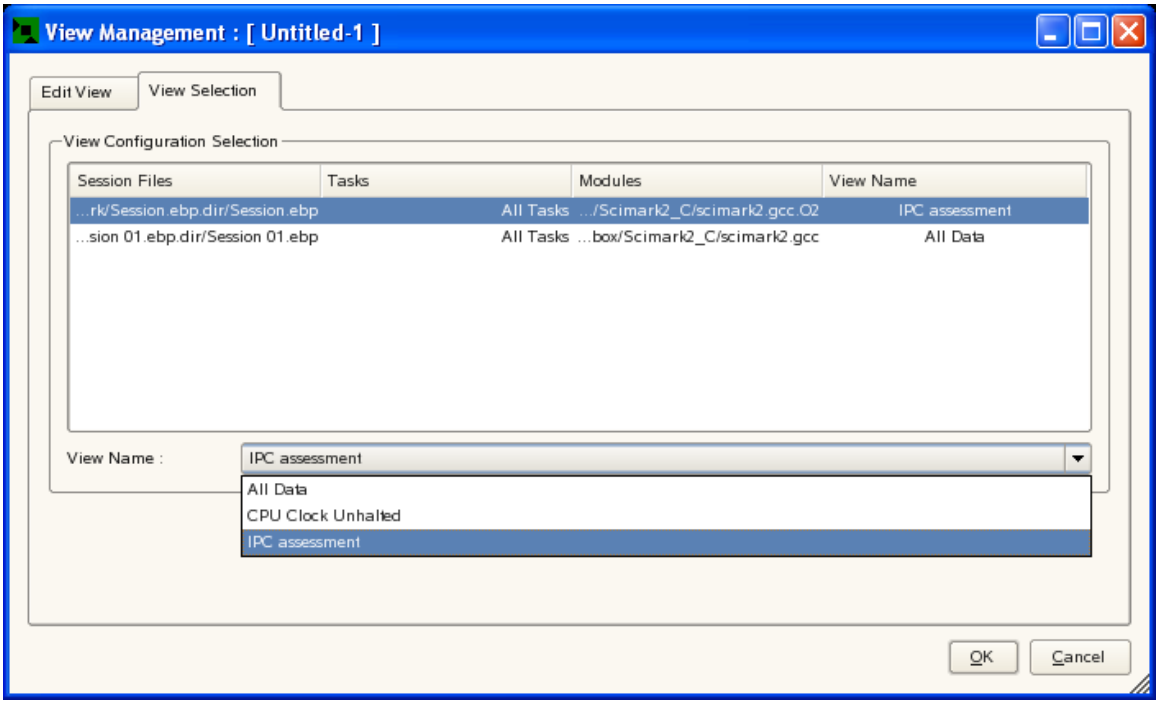
Figure 2.6. Edit View Tab



2.1.4.2. View Selection Tab

The View Selection tab manages view selection of each profiling session to be compared. The user selects a profiling session from the **View Configuration Selection** list and uses the **View Name** combo box to choose the view of interest.

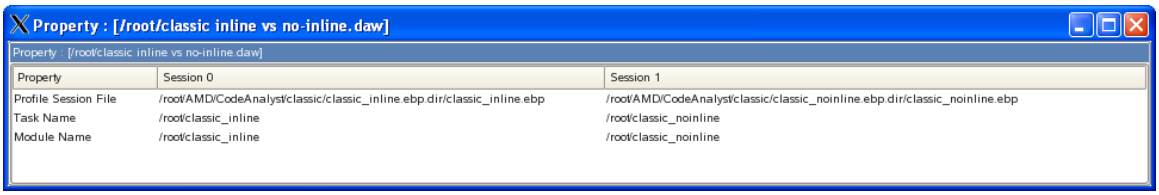
Figure 2.7. View Selection Tab



2.1.5. Property View

Property View is a docked window that contains a list of currently viewed profiling sessions. This dock window can be moved or undocked. It can also be hidden by closing (**X**) or clearing the check box for the option under **Tools > Show Property View**. Properties shown are profiling session files, task names, and module names.

Figure 2.8. Property View



Chapter 3. Support

3.1. Enhancement Request

Please email the following information about a desired enhancement or change to CodeAnalyst.support@amd.com [mailto:CodeAnalyst.support@amd.com]:

- State which version of AMD CodeAnalyst you are using. Choose Help > About to view the About AMD CodeAnalyst dialog box.
- Describe the desired enhancement or change.
- Indicate to us how important this is to you using a scale of 1 to 5 where 1 is most important and 5 least important.

3.2. Problem Report

If a problem is found, take the following action:

1. Run careport.sh script which is located in CodeAnalyst root directory of the source tree, or /opt/CodeAnalyst/bin/careport.sh. This script will generate a report file called CAReport.txt.
2. Please provide the following information:
   - Give a description of the problem or issue.
   - Briefly describe the steps or sequence of events leading to the observation.
   - State how frequently problem occurred.
   - Describe the messages AMD CodeAnalyst displayed.
   - State which version of the AMD CodeAnalyst was used (under Help > System Info or opcontrol --version).
   - Describe the application analyzed.
3. Please send the report file (CAReport.txt) in step 1 and information in step 2 to CodeAnalyst.support@amd.com [mailto:CodeAnalyst.support@amd.com].

Appendix A. GNU General Public License

Version 2, June 1991

Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor,
Boston, MA 02110-1301
USA

Version 2, June 1991

A.1. Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

1. copyright the software, and

2. offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

A.2. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

A.2.1. Section 0

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

A.2.2. Section 1

You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

A.2.3. Section 2

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section A.2.2, “Section 1” above, provided that you also meet all of these conditions:

a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: If the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on

the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

A.2.4. Section 3

You may copy and distribute the Program (or a work based on it, under Section A.2.3, “Section 2” in object code or executable form under the terms of Section A.2.2, “Section 1” and Section A.2.3, “Section 2” above provided that you also do one of the following:

a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

A.2.5. Section 4

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.2.6. Section 5

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

A.2.7. Section 6

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

A.2.8. Section 7

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

A.2.9. Section 8

If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

A.2.10. Section 9

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

A.2.11. Section 10

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free

Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

A.2.12. NO WARRANTY Section 11

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

A.2.13. Section 12

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

A.3. How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type “show w”. This is free software, and you are welcome to redistribute it under certain conditions; type “show c” for details.

The hypothetical commands “show w” and “show c” should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than “show w” and “show c”; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program “Gnomovision” (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Bibliography

BIOS and Kernel Developer's Guide (BKDG)

BIOS and Kernel Developer's Guide (BKDG) For AMD Family 11h Processor
[http://support.amd.com/us/Processor_TechDocs/41256.pdf]

BIOS and Kernel Developer's Guide (BKDG) For AMD Family 10h Processor
[http://support.amd.com/us/Processor_TechDocs/31116.pdf]

BIOS and Kernel Developer's Guide for AMD Athlon™ and AMD Opteron™ Processors (Rev. A-E)
http://support.amd.com/us/Processor_TechDocs/26094.PDF

BIOS and Kernel Developer's Guide for AMD NPT Family 0Fh Processors (Rev. F-G)
http://support.amd.com/us/Processor_TechDocs/32559.pdf

General Documentation

Basic Performance Measurements for AMD Athlon™ 64, AMD Opteron™ and AMD Phenom™ Processors
http://developer.amd.com/Assets/Basic_Performance_Measurements.pdf

Increased performance with AMD CodeAnalyst software and Instruction-Based Sampling (on Linux)
http://developer.amd.com/Assets/amd_ca_linux_june_2008.pdf

An introduction to analysis and optimization with AMD CodeAnalyst Performance Analyzer
http://developer.amd.com/Assets/Introduction_to_CodeAnalyst.pdf

Improving program performance with AMD CodeAnalyst for Linux®
http://developer.amd.com/assets/Linux_Summit_PJD_2007_v2.pdf

Instruction-Based Sampling: A New Performance Analysis Technique for AMD Family 10h Processors
http://developer.amd.com/assets/AMD_IBS_paper_EN.pdf