CodeDynamics 2018.3 Release Notes

Updated: October 2018

These release notes contain a summary of new features and enhancements, late-breaking product issues, migration from earlier releases, and bug fixes.

PLEASE NOTE: The version of this document in the product distribution is a snapshot at the time the product distribution was created. Additional information may be added after that time because of issues found during distribution testing or after the product is released. To be sure you have the most up-to-date information, see the version of this document on the Rogue Wave web site:

://www.roguewavhttpe.com/support/product-documentation/codedynamics.aspx

Additions and Updates

Improved CUDA Debugging Support

For 2018.3, CodeDynamics provides a number of advances with its support for debugging CUDA applications:

- CUDA 9.2 and CUDA 10.0
  This release provides official support of CUDA 9.2 and CUDA 10.0. CodeDynamics now supports CUDA 8.0, 9.0-9.2 and 10.0.
- Multi-GPU debugging improvements
  The CodeDynamics development team has been collaborating with customers utilizing architectures with multi-GPU enables compute nodes to improve the capabilities, reliability and performance of debugging on advanced multi-GPU capable applications.
- New-UI CUDA debugging improvements
  Continued improvements to the CodeDynamics new UI have been made to improve debugging of CUDA applications. Enhancements include a new GPU navigation bar that allows for easy navigation between the Logicial or Physical coordinates of the GPU, performance improvements when displaying breakpoints for GPU code and other stability improvements.

CLANG/LLVM and GDB JIT Interface Support

The latest versions of CLANG/LLVM are able to generate compiled code on the fly using the GDB Just-In-Time (JIT) compilation interface. Debugging this code can be difficult as the debugger needs to pick up the generated debug information during the debug session. This latest version of CodeDynamics detects the use of JIT'd code and will automatically read the new code and debug information so it can be debugged.

User Interface Improvements

The CodeDynamics UI continues to have improvements, fixes and performance enhancements added to it. If you have any feedback about the new user interface, requests for new or missing features or any problems please send email to tv-beta@roguewave.com.

- Barrier Point Support
  The ability to create Barrier Points in order to synchronize threads and processes has been added to the new CodeDynamics user interface.
- Set Breakpoint
  In addition to simply clicking a line number in the source area, the ability to set a breakpoint for a selected line has also been added as an option to the Action Points menu through the Set Breakpoint menu item.
- Bug fixes and performance improvements
  Numerous bug fixes and performance improvements have been made to the new UI including improving the performance of UI updates when stepping applications and displaying large source files with many breakpoints.

Bug Fixes for 2018.3

TVT-26657	Source code does not refresh when relaunching application after changing code, recompiling and linking.
TVT-26654	Stepping in the UI is visibly slower.
TVT-26568	Performance problem w/ CUDA ELF images with a large number of kernels/asects
TVT-26467	Configure TotalView with TV::dlopen_always_recalculate==false to improve performance.
TVT-26466	TotalView server crashing with a std::bad_alloc message
TVT-26292	TV::scope in reference guide has loader_sym_by_regexp description for code_unit_for_soid

Deprecation Notices

Mac OS X El Capitan

CodeDynamics release 2018.1 was the last release to support Mac OS X El Capitan.

RedHat Enterprise Linux 5

Beginning with release, 2018, CodeDynamics no longer supports RHEL 5.

32-bit Mac OS X Application Debugging Support on High Sierra and later OS versions

Apple is phasing out support for 32-bit applications beginning with their latest High Sierra release and because of this CodeDynamics will no longer support debugging 32-bit applications on High Sierra and later versions of Mac OS X.

Known Issues

Python Debugging

Anaconda

CodeDynamics supports debugging of the python interpreter in release 4 of Anaconda but is not working with the recent release of Anaconda 5. Debugging python with GDB also does not work. Something in the way they build the python interpreter has broken the ability to debug python.

Ubuntu 16.04

When debugging python on Ubuntu 16.04 CodeDynamics is not detecting the python interpreter automatically and does not turn on python filtering. Filtering can be turned on by clicking the "filter" icon in the toolbar of the Call Stack view.

macOS

Physical console access needed when running CodeDynamics on macOS High Sierra

Due to new security changes in macOS High Sierra, CodeDynamics will only run from the console and cannot be run through a remote desktop technology such as VNC. We are still assessing what changes need to be made to CodeDynamics so that it will run remotely on macOS High Sierra systems.

Reverse Debugging with Replay Engine

Turning on Replay on Skylake processors triggers the following ASSERT at the terminal

```
**************************************************************
Assertion failure (ignored): asm_xsavesize() <= sizeof(sys_x86_xsave_t)
Additional information:
Location: instr/cpuid_x86.c:145:have_xsave [411493:411493]
**************************************************************
```

The ASSERT is benign and does not interfere with the functionality our Replay support. However, if the application is using additional features of the CPU such as PKU (protection keys) then you may encounter other problems. Also, if you try and record an application using the newer instruction sets, such as AVX-512, on this processor you may also see some SIGILLs.

The ASSERT should be lifted in a future release of CodeDynamics that has a newer version of our Replay engine.