

=====

Intel (R) Thread Checker 3.1 for Linux*

=====

Release Notes

=====

Contents

- Overview
- Product Contents
- What's New
- System Requirements
- Known Issues and Limitations
- Technical Support
- Related Products

Overview

Intel (R) Thread Checker for Linux* is designed to help you create threaded applications more quickly by detecting thread safety issues in programs that use POSIX* and OpenMP* threads. Use it to identify source locations where shared memory or synchronization conflicts could cause non-deterministic results or deadlock.

Linux* Remote Data Collection is no longer part of the Windows* product, but it is still available for legacy usage. Contact customer support for more information.

Product Contents

Intel (R) Thread Checker for Linux* to be installed on Linux* systems.

What's New

- Support for the latest Intel (R) multi-core processors, including new Intel (R) Core(TM) Duo, Intel (R) Core(TM) Quad processors, Dual-Core Intel (R) Xeon and Dual-Core Intel (R) Itanium(R) 2 series.
- A new command line report format. Formats include: text, html and csv.
- Performance optimizations by descriptor-based static binary instrumentation which
 - reduces the instrumented image size
 - decreases the runtime overhead

System Requirements

Linux* systems with Pentium(R) processors

Minimum Hardware Required

Intel (R) Pentium(R) 4 processor
512 MB of RAM
100 MB of disk space

Recommended Hardware

Pentium(R) 4 processor supporting Hyper-Threading Technology or
Intel (R) Xeon(R) processor or higher
2 GB of RAM

Required Software

Red Hat* Enterprise Linux* 3.0, Red Hat* Enterprise Linux* 4.0,
Red Hat* Enterprise Linux* 5.0, Fedora Core* 5, Red Flag* DC Server 5.0,
SUSE* Linux* Enterprise Server 9.0, SUSE* Linux* Enterprise Server 10.0,
GreatTurbo* Enterprise Server 10, SGI* Propack 4.0,
SGI* Propack 5.0, Miracle* Linux* 4.0, Mandriva*/Mandrake* 10.1,
HaanSoft* Linux* Server 2006.

When Intel (R) compilers are used with compiler instrumentation(-tcheck),
Intel (R) compiler version 9.1 or higher must be used for platforms with
Intel (R) 64 architecture.

Required Software for OpenMP* Analysis or Source Instrumentation

Intel (R) C++ Compiler for Linux* 8.1, Package ID: I_cc_p_8.1.026 or higher
Intel (R) Fortran Compiler for Linux* 8.1, Package ID: I_fc_p_8.1.023 or higher

Linux* systems with Itanium 2 (R) processors

Minimum Hardware Required

Intel (R) Itanium(R) processor
512 MB of RAM
100 MB of disk space

Recommended Hardware

Dual Intel (R) Itanium(R) 2 processors or higher
2 GB of RAM

Required Software

Red Hat* Enterprise Linux* 3.0, Red Hat* Enterprise Linux* 4.0,
Red Flag* DC Server 5.0, SUSE* Linux* Enterprise Server 9.0,
SUSE* Linux* Enterprise Server 10.0, GreatTurbo* Enterprise Server 10, SGI*
Propack 4.0,
SGI* Propack 5.0, Miracle* Linux* 4.0, Mandriva*/Mandrake* 10.1, HaanSoft* Linux*
Server 2006.

When Intel (R) compilers are used with compiler instrumentation(-tcheck),
Intel (R) compiler version 9.1 or higher must be used for platforms with
Intel (R) Itanium(R) platforms.

Required Software for OpenMP* Analysis or Source Instrumentation

Intel (R) C++ Compiler for Linux* 9.0, Package ID: I_cc_p_9.0.021 or higher
Intel (R) Fortran Compiler for Linux* 9.0, Package ID: I_fc_p_9.0.021 or higher

Known Issues and Limitations

This release does not support Unicode* file names.

Intel (R) Thread Checker for Linux* does not support binary instrumentation on
Intel (R) Itanium(R) platforms.

Release_Notes

Intel (R) Thread Checker for Linux* does not support Intel (R) VTune(TM) Performance Environment Command Line interface(vtl).

Intel (R) Thread Checker for Linux* requires full path to the modules while using -m and -x options.

Intel (R) Thread Checker may face some incompatibility issues with the libguide library provided by older versions of Intel (R) Math Kernel Library. Please upgrade to the latest version of Intel (R) Math Kernel Library for Linux* (8.0 and higher) if you encounter error message similar to "undefined references to _kmpc_global_thread_num".

Source instrumentation on Intel (R) Itanium(R) platforms requires the -ltcdata switch for the Intel (R) compiler.

If you are using counted FLEXlm* license files during the installation of the client applications, please make sure that either the FLEXlm* server is not running or the FLEXlm* server is running with the complete license files which are used during the installation process.

The binary instrumentation technology used by Intel (R) Thread Checker for Linux* may cause some applications to change behavior or terminate abnormally. If this happens, try lowering the instrumentation levels of the problematic modules.

Intel (R) Thread Checker for Linux* does not detect threading errors for threads and synchronization objects shared across processes (executables).

Intel (R) Thread Checker for Linux* cannot analyze processes that are already running. You must either specify the executable as the application to launch, or specify it as a module of interest.

Intel (R) Thread Checker for Linux* supports analysis of native binaries and does not support intermediate executable representations intended for managed runtime environments.

If an application has data races on a global variable, the definition information of the variable will not be available unless source instrumentation is used.

Software that defines functions with names that match system API names such as fread() or pthread_create() may have link failures when built with source instrumentation using "-tcheck". Such software may crash when run with Intel (R) Thread Checker. To avoid this problem, ensure that your software uses function names that do not match system API names to avoid these issues.

When using source instrumentation with functions that contain inline assembly, Thread Checker may generate invalid diagnostics if data is collected outside of the Intel (R) Thread Checker for Linux* Command Line or VTune(TM) Performance Environment. To avoid this, compile with the "-tcheck" option and run your software in the Intel (R) Thread Checker for Linux* Command Line or VTune(TM) environment.

Linux* Analysis

To use source instrumentation on Linux*, you must set up the environment before compiling with "-tcheck" or starting the ittserver, for example:

```
Linux* 32-bit:  
  . /opt/intel/itt/tcheck/bin/32/tcheckvars.sh  
or  
  source /opt/intel/itt/tcheck/bin/32/tcheckvars.csh
```

```
Linux* 64-bit on Intel (R) 64 architecture:  
  . /opt/intel/itt/tcheck/bin/32e/tcheckvars.sh
```

Release_Notes

```
or
source /opt/intel/it/tcheck/bin/32e/tcheckvars.csh

Linux* 64-bit on Itanium(R) processors:
. /opt/intel/it/tcheck/bin/64/tcheckvars.sh
or
source /opt/intel/it/tcheck/bin/64/tcheckvars.csh
```

This release does not support the following types of Linux* binaries:

- Linux applications statically linked with C/C++ runtime library or the POSIX* thread library.
- The Linux* kernel and any kernel modules.
- The Linux* dynamic linker/loader (/lib/ld.linux.so.*, /lib/ld.*.so).

Asynchronous signal handling is not fully supported in this release. Programs that use asynchronous signals may stall when analyzed with Intel (R) Thread Checker.

When an instrumented application calls the exec system function, the whole image context is replaced with the new image. If the application calls exec with the name of the instrumented image, then the instrumented image is used. In the other cases, the original (non-instrumented) image is called, and no results will be generated from this point in the run.

Intel (R) Thread Checker for Linux* does not support setuid images. The setuid mechanism is used to give a user process the effective user ID of another user, usually root. You can use Intel (R) Thread Checker for Linux* and run the setuid executable only if you are logged in as the same user as the owner of the setuid executable.

The application to analyze cannot be a command-line application that expects interactive user input.

If pthread_exit() is called from the main thread, and if actual deadlock and stall detection or real time diagnostic display is turned on and forceful program termination upon deadlock or program exit is turned off, the program may hang under Intel (R) Thread Checker analysis. If this happens, please turn off deadlock and stall detection and real time diagnostic display or turn on forceful program termination upon deadlock or program exit.

Source and Binary instrumentation may dramatically increase the use of stack space. On Linux* you can use the limit or ulimit commands to specify the stack space used by applications. You may need to increase the stack space to complete the data collection. However, be careful with an unlimited stack space specification. This can result in insufficient space to create the pthread stacks which can cause the data collection to fail.

Technical Support

The product support web site (<http://support.intel.com/support/performance/tools/threadchecker/>) contains frequently asked questions, product documentation, product errata, as well as solutions to common issues.

To receive technical support for this product or product updates, you need to register for an Intel (R) Premier Support account at the Intel (R) Registration Center (<http://www.intel.com/software/products/registrationcenter/>).

When submitting an issue to Intel (R) Premier Support

Release_Notes

(<https://premier.intel.com/>), be sure to select Intel (R) Thread Checker for Linux* from the Product Name drop down list.

When submitting an issue please provide the product build number. This information can be found in the tcSupport.txt file. This file is located on the

Linux* system in the directory where Intel (R) Thread Checker for Linux* was installed. The default installation directory is "/opt/intel/itt".

Once you have contacted us with your suggestion or problem using your Premier Support account, a technical support engineer will respond within one Intel business day.

If you have not received or have lost your Premier Support Login ID or password, or are having trouble with access, please visit <https://registrationcenter.intel.com/support> for assistance.

Related Products and Services

Information about Intel (R) Software Development Products is available at <http://www.intel.com/software/products>.

Some of the related products include:

- Intel (R) Thread Profiler is a performance tuning tool for parallel programs that use Win32*, Win64*, POSIX*, OpenMP* or custom synchronization. (<http://www.intel.com/software/products/threading>)
- VTune(TM) Performance Analyzer enables you to evaluate how your application is utilizing the CPU and helps you determine if there are modifications you can make to improve your application's performance. (<http://www.intel.com/software/products/vtune>)
- Intel (R) Compilers are an important part of making software run at top speeds with full support for the latest Pentium(R) and Itanium(R) processors. (<http://www.intel.com/software/products/compilers>)
- Intel (R) Cluster Tools can help developers create, analyze and optimize high-performance applications on clusters of Intel (R) processor-based systems. (<http://www.intel.com/software/products/cluster>)
- Intel (R) Performance Library Suite provides a set of routines optimized for various Intel processors. (<http://www.intel.com/software/products/perflib>)
- Intel (R) Software College provides training for developers on leading-edge software development technologies. Training consists of online and instructor-led courses covering all Intel architectures, platforms, tools, and technologies. (<http://www.intel.com/software/college>)

Disclaimer and Legal Information

The information in this document is subject to change without notice and Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. This document and the software described in it are furnished under license and may only be used or copied in accordance with the terms of the license. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. The information in this document is provided in connection with Intel products and should not be construed as a commitment by Intel Corporation.

Release_Notes

EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The software described in this document may contain software defects which may cause the product to deviate from published specifications. Current characterized software defects are available on request.

Intel, the Intel logo, Intel SpeedStep, Intel NetBurst, Intel NetStructure, MMX, i386, i486, Intel 386, Intel 486, Intel 740, Intel DX2, Intel DX4, Intel SX2, Celeron, Intel Centrino, Intel Xeon, Intel XScale, Itanium, Pentium, Pentium II Xeon, Pentium III Xeon, Pentium M, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.

Copyright (c) Intel Corporation 2007.