

THE ULTIMATE ALL-IN-ONE PERFORMANCE TOOLKIT

Intel® Parallel Studio 2011

Product Brief

Intel® Parallel Studio 2011



"Intel® Parallel Studio makes the new Envivio 4Caster* Series Transcoder's development faster and more efficient, shortens overall software development time, and globally speeds up our time-to-market."

*Eric Rosier
V.P. Engineering
Envivio*

Accelerate Development of Reliable, High-Performance Serial and Threaded Applications for Multicore

Intel® Parallel Studio 2011 is a comprehensive tool suite that provides C/C++ developers using Microsoft Visual Studio 2005*, 2008*, and 2010* with tools for serial and threaded application design. Offering innovative features and capabilities, the suite equips developers to design, build and debug, verify, and tune their applications. Intel Parallel Studio enables you to design reliable, high-performance threaded applications for multicore systems. It also offers compelling value to developers working on serial applications today and moving to threaded applications in the future.

Intel Parallel Studio enhances developer productivity through each phase of the development lifecycle, resulting in higher application reliability and performance, faster time to market, and increased ROI.

Intel® Parallel Studio 2011 tool suite overview:

Design Phase	INNOVATIVE THREADING ASSISTANT  Intel® Parallel Advisor	Threading design guide tool simplifies, demystifies, and speeds parallel application design	- Identifies the areas in applications that can benefit most from parallelism - Provides step-by-step guidance for threading applications
Build and Debug Phase	COMPILER AND THREADED LIBRARY  Intel® Parallel Composer	Optimizing compiler boosts performance and threaded application design	- C++ Compiler and libraries - Code coverage - Debugger - Intel® Parallel Building Blocks—Set of comprehensive parallel models that supports multiple ways to exploit parallelism
Verify Phase	MEMORY AND THREADING ERROR CHECKER  Intel® Parallel Inspector	Error detection analysis tool for higher code reliability and quality	- Finds memory leaks and corruption - Finds data races and deadlocks
Tune Phase	THREADING AND PERFORMANCE PROFILER  Intel® Parallel Amplifier	Tuning analysis for optimized performance and scalability	- Performance and scalability analysis - Locks and waits analysis

Intel Parallel Studio 2011 Tools—Reliable, High-Performance, Future Proof

Intel Parallel Studio advances all development lifecycles, providing powerful tools that can be used independently or together to meet the needs of workflows and development teams. From one-person startups to enterprises with thousands of developers working on a single application, Intel Parallel Studio enhances productivity and enables reliable applications that leverage legacy serial code, take advantage of multicore, and scale for manycore.

Benefits:

- Access to productivity tools for higher application reliability, quality, and performance
- Provides an easy path to parallelism, helps minimize the learning curve, and improves productivity to accelerate return on hardware and software investments
- Forward scaling enables applications to take advantage of multicore and scale for manycore, while minimizing code changes.
- Offers more choices and flexibility for both data and task parallel programming models
- Preserves existing investments in source code and development environments that work on serial code and parallel code today and into the future

DESIGN

INNOVATIVE THREADING ASSISTANT

Intel® Parallel Advisor 2011

The Intel® Parallel Advisor threading assistant tool for Microsoft Visual Studio* C/C++ developers simplifies adding threading to existing source code. It guides developers and architects through the process of exploring threading, identifying those areas with the greatest threading potential. It provides tools for proposing threading, and for evaluating the performance and correctness of the proposed threading. This insight helps you make better design decisions, providing information on the consequences of those choices and suggesting ways to resolve issues at the design stage—before major effort has been committed.

Simplify the threading of your applications

- Get step-by-step guidance for modeling threading in your applications
- Make better design decisions by analyzing threading proposals
- Model parallel code to applications at points of greatest impact
- Get a clear roadmap for threading your application
- Save time, increase productivity, and speed time to market

Empowers architects with threading design insight and analysis for best results

With the Intel Parallel Advisor workflow approach, you get guidance and the tools for the best approach to implementing parallelism and exploiting the power of multicore.

Intel Parallel Advisor analyzes the executing serial program as developers work through the methodology.

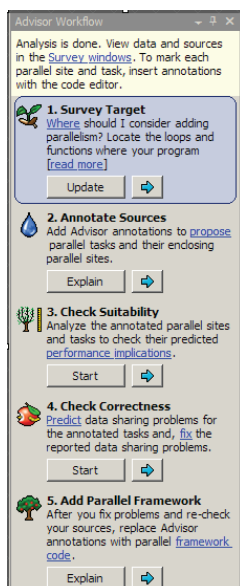


Figure 1.

Survey Target—Focuses on the hot call trees and loops as locations to experiment with threading

Annotate Sources—Inserts Intel Parallel Advisor annotations into your sources to describe parallel experiments

Check Suitability—Evaluates the performance of your threading experiment by displaying the performance projection for each parallel site and how each site's performance impacts the entire program

Check Correctness—Identifies data issues (races) in the threading experiment

Benefits:

- Migrate serial applications to parallelism and take threaded applications to the next level
- Speed time to market
- Enable feature-rich applications
- Use with any C/C++ application
- Take advantage of interoperability with Microsoft* and Intel® Compilers
- Find solutions for applications that are not easily threaded

BUILD AND DEBUG

COMPILER AND THREADED LIBRARY

Intel® Parallel Composer 2011

Intel® Parallel Composer includes Intel® C++, a performance-oriented, optimizing compiler; Intel® Integrated Performance Primitives (Intel® IPP); and Intel® Parallel Building Blocks (Intel® PBB). All are compatible with Microsoft Visual Studio*, and together they enhance application performance and improve productivity as you develop application software.

Intel Parallel Composer makes it easier for Visual* C++ developers to deliver outstanding application performance. With features like auto-vectorization and auto-parallelization, just recompiling with Intel C++ will do the job. Intel PBB provides an efficient way to add performance through parallelism utilizing simple concurrency functions, data parallel arrays, and pre-threaded domain-specific libraries. It includes Intel® Threading Building Blocks (Intel® TBB), an STL-like collection of basic functions; Intel® Array Building Blocks (beta) for implementing parallelism in arrays; and Intel® Cilk™ Plus for simple loop and task parallelism. These tools enable Intel Parallel Composer to provide a variety of broadly applicable and application-specific capabilities, making it easier to deliver performance as a feature in application software.

Build high-performance serial and threaded applications for multicore

- Intel Composer components are compatible with Microsoft Visual Studio, which supports the way you develop software, while safeguarding your investment and delivering improved performance.
- Intel PBB's parallel programming models offer more choices to match the parallel programming needs of businesses today and into the future.
 - Intel® Threading Building Blocks (Intel® TBB) is a C++ template library solution that can be used to enable general parallelism. It includes scalable memory allocation, load balancing, work-stealing task scheduling, a thread-safe pipeline and concurrent containers, high-level parallel algorithms, and numerous synchronization primitives. Intel TBB is for C++ developers who write general-purpose loop and task parallelism applications.
 - Intel® Cilk Plus is an Intel® C/C++ Compiler-specific implementation of parallelism. It offers superior functionality by combining vectorization features with high-level, loop-type data parallelism and tasking. Intel Cilk Plus is for C++ software developers who write simple loop, data, and task parallel applications.
 - (In beta) Intel® Array Building Blocks is an API backed by a sophisticated runtime library. It provides a generalized data parallel programming solution that frees application developers from dependencies on particular low-level parallelism mechanisms or hardware architectures. It produces scalable, portable, and deterministic parallel implementations from a single high-level, maintainable, and application-oriented specification of the desired computation. Intel Array Building Blocks is for software developers who write compute-intensive, data parallel algorithms. Intel Array Building Blocks is currently available in beta at: <http://software.intel.com/en-as/data-parallel/>.
- Enhance the capabilities of Microsoft Visual Studio 2005, 2008 and 2010, keeping your productivity high as you improve the performance of applications targeted at multicore
- Simplify adding parallelism with the newest versions of Intel® compilers and libraries for parallelism, and thousands of lines of ready-to-use code
- Save time and take advantage of multicore processors with automatic functions
- Support the entire spectrum of parallel expression from simple to complex, data to task, by offering application-class-specific, pre-threaded, and thread-safe libraries. This is a real time saver yielding excellent performance.
- Novice and advanced developers both benefit from Intel Parallel Composer capabilities and can quickly find help and advice in forums.

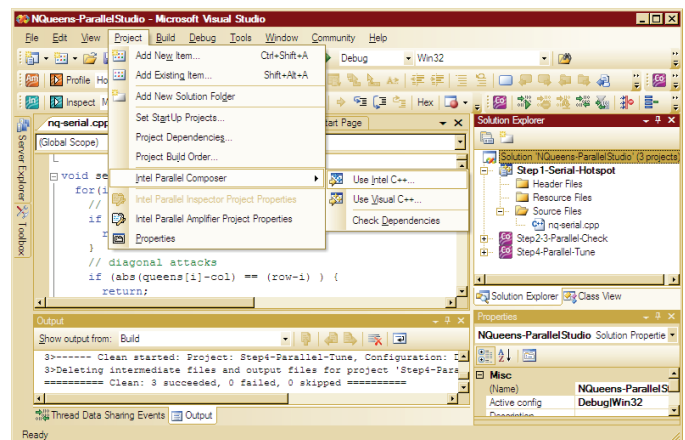


Figure 2. Intel® Parallel Composer integrates into Microsoft Visual Studio*. The solution on display shows how to switch to the Intel® C++ compiler. You can easily change to Visual C++* from the Project menu or by right-clicking over the solution or project name.

Benefits:

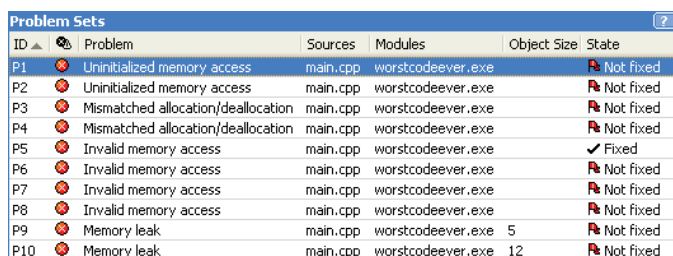
- Improve productivity
- Reduce the learning curve for developing threaded applications with effective, standard abstractions
- Protect your investment by building applications that scale for multicore and manycore

BUILD AND DEBUG

MEMORY AND THREAD ERROR CHECKER

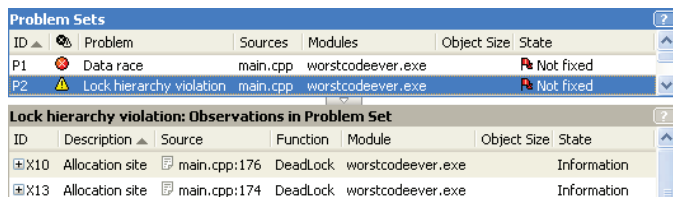
Intel® Parallel Inspector 2011

Intel® Parallel Inspector, a dynamic analysis tool, proactively finds coding defects in Windows* C/C++ serial and parallel applications. Unlike traditional error checkers, Intel Parallel Inspector detects hard-to-find memory and threading errors in one tool. It rapidly provides root-cause analysis for crash-causing threading and memory defects by monitoring the runtime behavior of the code, and then maps errors to the source-code line, call stack, and memory reference. This is especially critical for nondeterministic errors (i.e., where the execution sequence changes from run to run) that are difficult to reproduce.



ID	Problem	Sources	Modules	Object Size	State
P1	Uninitialized memory access	main.cpp	worstcodeever.exe		Not fixed
P2	Uninitialized memory access	main.cpp	worstcodeever.exe		Not fixed
P3	Mismatched allocation/deallocation	main.cpp	worstcodeever.exe		Not fixed
P4	Mismatched allocation/deallocation	main.cpp	worstcodeever.exe		Not fixed
P5	Invalid memory access	main.cpp	worstcodeever.exe		Fixed
P6	Invalid memory access	main.cpp	worstcodeever.exe		Not fixed
P7	Invalid memory access	main.cpp	worstcodeever.exe		Not fixed
P8	Invalid memory access	main.cpp	worstcodeever.exe		Not fixed
P9	Memory leak	main.cpp	worstcodeever.exe	5	Not fixed
P10	Memory leak	main.cpp	worstcodeever.exe	12	Not fixed

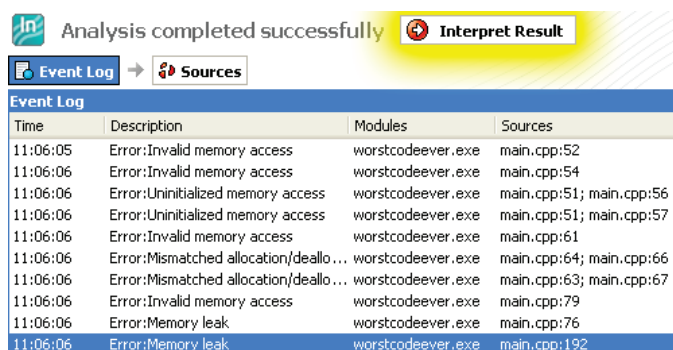
Figure 3. Quickly find memory errors, including leaks and corruptions, in single and multithreaded applications. This decreases support costs by locating memory errors before an application ships.



ID	Problem	Sources	Modules	Object Size	State
P1	Data race	main.cpp	worstcodeever.exe		Not fixed
P2	Lock hierarchy violation	main.cpp	worstcodeever.exe		Not fixed

ID	Description	Source	Function	Module	Object Size	State
X10	Allocation site	main.cpp:176	DeadLock	worstcodeever.exe		Information
X13	Allocation site	main.cpp:174	DeadLock	worstcodeever.exe		Information

Figure 4. Accurately pinpoint latent threading errors, including deadlocks and data races. This helps reduce stalls and crashes due to threading errors not found by debuggers and other tools.



Analysis completed successfully Interpret Result

Event Log → Sources

Time	Description	Modules	Sources
11:06:05	Error:Invalid memory access	worstcodeever.exe	main.cpp:52
11:06:06	Error:Invalid memory access	worstcodeever.exe	main.cpp:54
11:06:06	Error:Uninitialized memory access	worstcodeever.exe	main.cpp:51; main.cpp:56
11:06:06	Error:Uninitialized memory access	worstcodeever.exe	main.cpp:51; main.cpp:57
11:06:06	Error:Invalid memory access	worstcodeever.exe	main.cpp:61
11:06:06	Error:Mismatched allocation/deallo...	worstcodeever.exe	main.cpp:64; main.cpp:66
11:06:06	Error:Mismatched allocation/deallo...	worstcodeever.exe	main.cpp:63; main.cpp:67
11:06:06	Error:Invalid memory access	worstcodeever.exe	main.cpp:79
11:06:06	Error:Memory leak	worstcodeever.exe	main.cpp:76
11:06:06	Error:Memory leak	worstcodeever.exe	main.cpp:192

Figure 5. Click the [Interpret Result](#) button to see related issues grouped together. When you fix one problem, Intel Parallel Inspector shows you all of the related locations where the same fix needs to be applied.

- Increase efficiency when debugging threaded applications
- Improve performance by recompiling with Intel® C++ Compiler
- Enable portability, reliability, scalability, and simplicity

Catching software defects early in the development cycle can save you time and cost, and increase your ROI

- Get memory and threading error checking in one tool for serial and threaded code
- Support multiple threading methodologies
- Work on standard debug build
- Map errors to the source-code line, call stack, and memory reference

Benefits:

- Enjoy application reliability and quality
- Find latent memory errors, such as memory corruption and leaks
- Accurately pinpoint latent threading errors, such as data races and deadlocks, in threaded code
- Work with standard debug builds
- Run Intel Parallel Inspector on any functional serial or threaded application
- Get related source locations and contextual information for easier problem solving

TUNE

THREADING AND PERFORMANCE PROFILER

Intel® Amplifier 2011

Intel® Parallel Amplifier is a threading and performance profiler for Microsoft Visual Studio C/C++ developers who need to understand an application's parallel behavior to improve scalability. Though able to tune for serial performance, Intel Parallel Amplifier was, unlike other profilers, specifically designed for threaded applications and includes threading analysis to pinpoint multicore performance bottlenecks. As developers unlock the potential of parallelism and applications are enriched, Intel Parallel Amplifier can help fine-tune for optimal performance, ensuring cores are fully exploited and new capabilities are supported.

Optimize Performance and Scalability

- Intel Parallel Amplifier is an intuitive performance profiler for both serial and threaded applications.
- Hotspot analysis locates the bottleneck and calling sequence.
- Concurrency analysis highlights hotspots with underutilized cores.
- Locks and waits analysis finds synchronization object performance issues.
- Compare results from multiple runs
- Access low overhead statistical call-tree profiling
- View profiling results on your source.

Benefits:

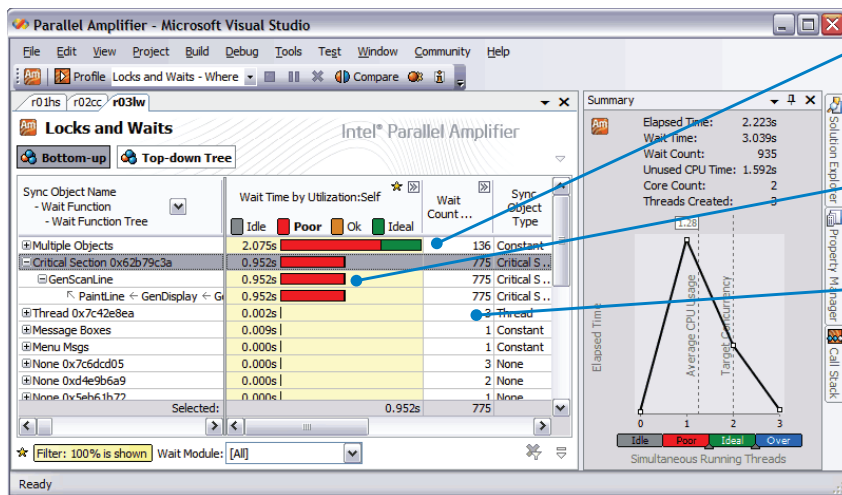
- Quickly find multicore performance bottlenecks without needing to know the processor architecture or assembly code
- Tune for performance and scalability
- Benefit from integration with Microsoft Visual Studio

The screenshot displays the Intel Parallel Amplifier interface within Microsoft Visual Studio. The main window is titled 'Parallel Amplifier - Microsoft Visual Studio'. The 'Hotspots' tab is selected, showing a 'Function - Caller Function Tree' table. The table lists functions and their CPU times, with 'Fractal.exe!GenColors' being the top hotspot at 2.252s. The 'Call Stack' panel on the right shows the call stack for the selected function, with 'Fractal.exe!GenColors' at the top. The 'Summary' panel at the bottom right provides overall statistics: Elapsed Time: 2.316s, CPU Time: 2.799s, Unused CPU Time: 1.833s, Core Count: 2, and Threads Created: 3. Blue arrows point to various features: 'Choose analysis type' points to the 'Hotspots' tab; 'View call stack bottom-up or top-down' points to the 'Bottom-up' and 'Top-down Tree' buttons; 'Call stacks for selected function' points to the 'Call Stack' panel; 'CPU times for hot functions' points to the 'Function - Caller Function Tree' table; 'Elapsed time, CPU time, and unused CPU time' points to the 'Summary' panel; and 'Filter results' points to the 'Filter: 100% is shown' dropdown.

Function	Caller Function Tree	CPU Time:Self	Module
GenColors		2.252s	Fractal.exe
GenScanLine	PaintLine	1.744s	Fractal.exe
PaintLine	GenDisplay	0.508s	Fractal.exe
GenDisplay	GetModuleFileNameA	0.458s	Fractal.exe
KFastSystemCallRet		0.028s	ntdll.dll
NTWaitForSingleObject		0.015s	ntdll.dll
strnicmp		0.015s	MSVCR80
RtlLeaveCriticalSection		0.015s	ntdll.dll
PaintLine		0.015s	Fractal.exe
[Unknown]		0.015s	SynTPFcs

Hotspot analysis: Where is my application spending time?

Find the functions in your application that consume the most time. This is where you'll want to tune or add threading to make your program faster. Intel Parallel Amplifier also shows the stack, so you know how the function is being called. For functions with multiple calling sequences, this lets you see if one of the call stacks is hotter than the others.



Length of bar is wait time, color is number of cores utilized during the wait.

Waiting with underutilized cores hurts performance.

Wait count helps identify interesting waits.

Locks and waits analysis: Where are the bad waits?

Waiting too long on a lock is a common source of performance problems. It's not bad to wait while all the cores are busy (green), but it is bad to wait when there are unused cores available (red).

"The performance benefits of multicore and manycore are critical to SIMULIA's business. Intel® Parallel Inspector provides a powerful way to develop parallel code compared to traditional methods, which can be lengthy and costly—especially if the price of unstable code is paid by the consumer."

Mark Dunbar
Chief Architect
SIMULIA

Evaluation Guides

<http://software.intel.com/en-us/articles/evaluation-guides/>

System Requirements

- Microsoft Visual Studio 2005*, 2008*, 2010*
(except the Express Edition)
- For the latest system requirements, go to:
www.intel.com/software/products/systemrequirements/

Support

Purchase of Intel® Parallel Studio products include Premium Support service which allows you to submit questions, access to product updates, and technical documentation.

For more information, go to:

<http://software.intel.com/sites/support/>

Download a Trial Version Today

Evaluation copy available at:

www.intel.com/software/products/ParallelStudio/

The Ultimate All-in-One Performance Toolkit—Intel® Parallel Studio 2011

Designed for today's serial applications and tomorrow's software innovators

Intel brings simplified threading to Microsoft Visual Studio* C++ developers with a complete productivity solution designed to optimize serial and new threaded applications for multicore and scale for manycore.

INNOVATIVE THREADING ASSISTANT

Intel® Parallel Advisor 2011: Demystify and speed threaded application design.

COMPILER AND THREADED LIBRARIES

Intel® Parallel Composer 2011: Develop effective applications with a C/C++ compiler and advanced threaded libraries.

MEMORY AND THREADING ERROR CHECKER

Intel® Parallel Inspector 2011: Ensure application reliability with proactive parallel memory and threading error checking.

THREADING AND PERFORMANCE PROFILER

Intel® Parallel Amplifier 2011: Quickly find bottlenecks and tune threaded applications for scalable multicore performance.

