Easily Find Threading and Memory Errors before They Happen

Intel® Parallel Inspector is the easiest multithreading error checking tool for Microsoft Visual Studio® C/C++ developers. Intel Parallel Inspector detects challenging threading and memory errors and provides guidance to help ensure application reliability. Unlike other error checkers on the market, Intel Parallel Inspector is the fastest and most comprehensive method to pinpoint latent multithreading and memory errors.

Proactively find latent threading and memory errors to help ensure application reliability with Intel Parallel Inspector.

• Find memory and threading errors with one easy-to-use tool
• Give both experts and novices greater insight into parallel code behavior
• Help ensure that shipped applications run error-free on customer systems
• Find latent bugs within the increasing complexity of parallel programs
• Reduce support costs and increase productivity

“Intel® Parallel Inspector and Intel® Parallel Amplifier greatly simplified the task of finding hotspots and memory leaks. We were pleased with the 2X overall performance improvement and the elimination of several previously unidentified memory leaks.”

Vlad Romashko
Software Development Manager
OpenCascade S.A.S.
Memory and Thread Checking in One Easy-to-Use Tool

Both memory and thread checking are fully integrated into Microsoft Visual Studio with an easy-to-use interface. Intel Parallel Inspector provides root-cause analysis of crash-causing threading and memory defects. These features, combined with problem set analysis that summarizes related bugs, make this the most comprehensive tool for finding memory and threading errors. Competitive products only support serial applications or do not provide comprehensive memory and thread correctness checking using one tool.

Dynamic Instrumentation That Works on Standard Builds

Intel Parallel Inspector doesn’t require special builds or compilers, and utilizes dynamic instrumentation to acquire test data. Since it only instruments the code that’s executed, analysis can run in less time and work on larger applications.

Thread-Aware Memory Checker

Not all memory checkers are capable of performing analysis of threaded applications. Intel Parallel Inspector performs comprehensive memory checks (e.g., memory leaks, invalid memory read/write, dangling pointer detection, use of uninitialized data) on both single and multithreaded applications.

Excellent Value

Intel Parallel Inspector is aggressively priced for a combined memory and threading correctness tool, so it’s an excellent value compared with competitive products. Intel Parallel Inspector is included in Intel® Parallel Studio, which is a comprehensive suite of products for developing, debugging, and tuning parallel C/C++ applications.

Quickly finds memory errors, including leaks and corruptions, in single and multithreaded applications. This decreases support costs by finding memory errors before an application ships.

Accurately pinpoints latent threading errors including deadlocks and data races, which helps reduce stalls and crashes due to common errors not found by debuggers and other tools.

Intuitively guides the developer by grouping related issues together. When you fix one problem, Intel® Parallel Inspector shows you all of the related locations where the same fix needs to be applied.
Result suppression reduces the information that has to be analyzed by suppressing results that are not relevant.

Simple analysis configuration enables developers to control the depth of analysis vs. collection time.

- L1 analysis finds memory leaks and deadlocks.
- L2 analysis identifies the existence of a problem.
- L3 analysis provides root cause information to fix problems.
- L4 provides the most comprehensive level of problem identification and detail.

---

focused observation: main.cpp:50 - Read

```c
48; 49  c = p[0];
50  c = p[1];
```

related observation: main.cpp:44 - Allocation site

```c
42; 43  void doitx()
44  {
45; 46  char* p = (char*)malloc(4);
47  p[4] = 'a';
48  char c = p[5];
49  c = p[0];
50  c = p[1];
```

---

Observations in Problem Set: Uninitialized memory access

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Source</th>
<th>Function</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>X3</td>
<td>Allocation site</td>
<td>main.cpp:44</td>
<td>doitx</td>
<td>update_system.exe</td>
</tr>
<tr>
<td>X5</td>
<td>Read</td>
<td>main.cpp:50</td>
<td>doitx</td>
<td>update_system.exe</td>
</tr>
</tbody>
</table>

Click on an identified problem to reveal source code to go directly to the offending code to make changes quickly.
Intel® Parallel Studio

Designed for today’s serial applications and tomorrow’s software innovators.

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

**Intel® Parallel Studio:** Create optimized serial and parallel applications with the ultimate all-in-one parallelism toolkit

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

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

**Intel® Parallel Amplifier:** Quickly find bottlenecks and tune parallel applications for scalable multicore performance

---

**Features**

- Fully integrated with Microsoft Visual Studio®
- Find memory errors in single and multithreaded applications
  - Memory checking includes uninitialized load detection, use of invalid memory references, mismatched memory allocation and deallocation, memory leaks detection, stack memory checks, and stack trace with controllable stack trace depth
- Find threading errors
  - Data race detection, deadlock detection, depth configurable call stack analysis, diagnostic guidance, built-in knowledge of Intel® Threading Building Blocks, OpenMP®, and Windows® Threads
- Works with any standard debug build
  - No special test builds or compilers required, so it’s easier to test code more often
- Dynamic instrumentation enables testing code without the source; test larger applications because less memory is needed since only executed code is instrumented

**System Requirements**

- Microsoft Visual Studio
- For the latest system requirements, go to: [www.intel.com/software/products/systemrequirements/](http://www.intel.com/software/products/systemrequirements/)

**Compatibility**

- Compilers: Microsoft Visual C++® Compiler 2005 and 2008 and Intel® C++ Compiler
- Threading methodologies: Intel® Threading Building Blocks, OpenMP®, Windows® Threads
- Processors: Designed for and tested on Intel® IA-32 and Intel® 64 processors including Intel® Core™2 and Core™ i7 processors. It can be used on compatible processors, although proprietary instructions may cause it to function incorrectly. Please note that Intel® Parallel Composer (compiler and libraries) supports Intel IA-32, Intel 64, and all compatible processors.

**Support**

Intel Parallel Studio products include access to community forums and a knowledge base for all your technical support needs, including technical notes, application notes, documentation, and all product updates.

For more information, go to [http://software.intel.com/sites/support/](http://software.intel.com/sites/support/)

Beta Versions Available Now

Download and register for the user forums at: [www.intel.com/software/ParallelStudioBeta/](http://www.intel.com/software/ParallelStudioBeta/)