

Cut out errors from the start to deliver high-quality code on time



BoundsChecker®

Automatic error detection and debugging

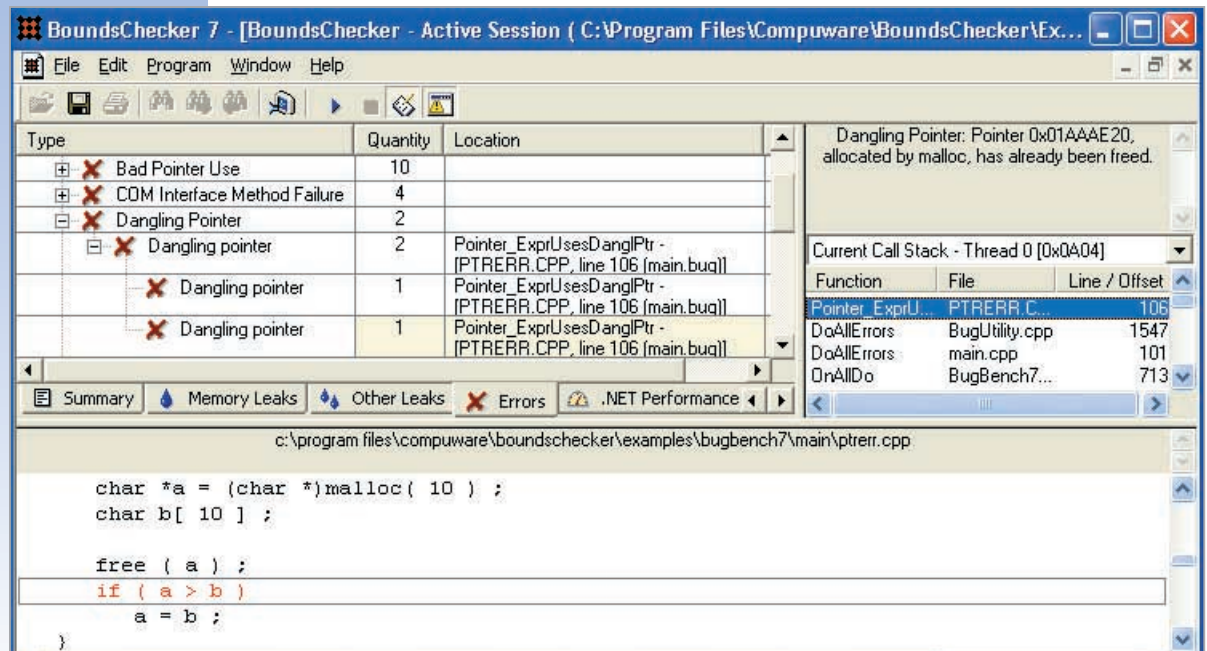
By industry estimates, developers take roughly 12 hours to find and fix a complicated software problem using typical trial-and-error techniques. However, you can address critical software errors in a fraction of the time using the automated error-detecting and debugging power of BoundsChecker from Compuware.

Multi-tiered applications and calls between new .NET applications and your existing code can leave you walking the fine line between managed and unmanaged (native) code, wondering where errors originated. Is it the managed code, the native code or the interaction between the two? You end up spending up to 80 percent of your time testing and debugging Visual C++ applications instead of actually developing them.

BoundsChecker automatically pinpoints and analyzes errors in source code for Visual C++ applications. It finds problems

as they pop up in both native Windows code and mixed managed/native code, capturing detailed analysis data at application runtime. You receive the specific information needed to fix problem code fast. Since BoundsChecker lets you spot and repair poor code up front, you:

- cut down the development time spent debugging and testing
- boost overall code quality
- deliver high-performing Visual C++ applications and components on time.



BoundsChecker error detection provides an interactive display of errors and other bugs, with stack information and the ability to automatically find and highlight the line of code generating the error.



Spot coding errors quickly at application runtime

Comprehensive error detection and memory monitoring

BoundsChecker tracks how much your application uses static, stack and heap memory, pens, bitmaps and other resources. When your application drains system resources or applies them incorrectly, BoundsChecker automatically reports where the error occurred with a call stack, so you can quickly trace the path back to the target line of code. With BoundsChecker, you gain:

- resource tracking for static destructors
- leak detection for static destructors
- enhanced support for use count analysis.

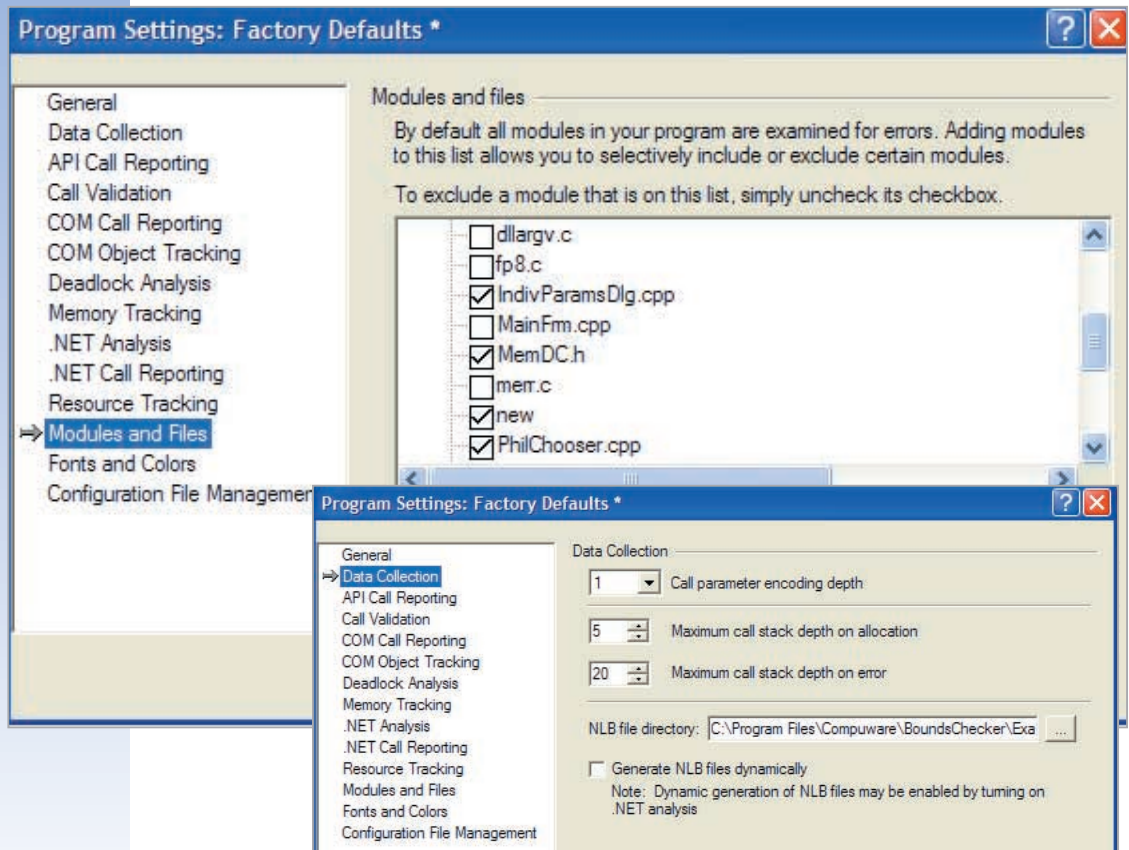
Runtime memory and resource analysis

Most analysis tools determine memory or resource leaks only after the application

finishes executing. You learn little about usage during the course of a program's execution. The BoundsChecker Memory and Resource Viewer provides a snapshot taken at any point. You can also set the Memory and Resource Viewer to examine memory and resource allocations during a specific segment of your application's execution. With these capabilities, you can:

- verify memory use on 24/7 server applications prior to production use
- prevent applications hanging from resource exhaustion
- identify early on when an application consumes large amounts of memory, rather than waiting until the end of an execution
- spot when large volumes of leaks occur before they dilute the signal at the end of the process
- determine if third-party components have memory or resource leaks.

BoundsChecker error detection is highly customizable, allowing you to select which files to check or suppress.



Extensive base of Windows programming interfaces

With thousands of Windows application programming interfaces (APIs) and OLE methods available to you, being an expert at each one is impossible. So making the mistake of passing the wrong data to a specific API or OLE method is easy. With BoundsChecker, you can check the accuracy of Windows API calls. BoundsChecker has a broad knowledge base that covers more than 9,000 Windows APIs and OLE methods, including ODBC, ActiveX, DirectX, COM and Internet APIs. BoundsChecker compares every call in the application to its extensive methods base and generates a detailed report of errors. You use API and OLE methods correctly right at the start of development.

On-demand problem summary

BoundsChecker sums up runtime problems with an easy-to-use interface that you control. Simply turn the option on and BoundsChecker immediately tells you when it detects an error. BoundsChecker automatically stores each error for subsequent review after the execution of the application. View the error list in its entirety or by summary report. Because the summary report sorts results by memory impact or error type, you can quickly prioritize which errors to resolve first.

Analyze errors at the managed-unmanaged boundary

Calling native code components from .NET applications is no problem with BoundsChecker. This automated error detection tool helps ensure platform calls

The thread deadlock capability identifies both potential and actual deadlocks encountered.

are correct by monitoring events that occur between managed and native code such as:

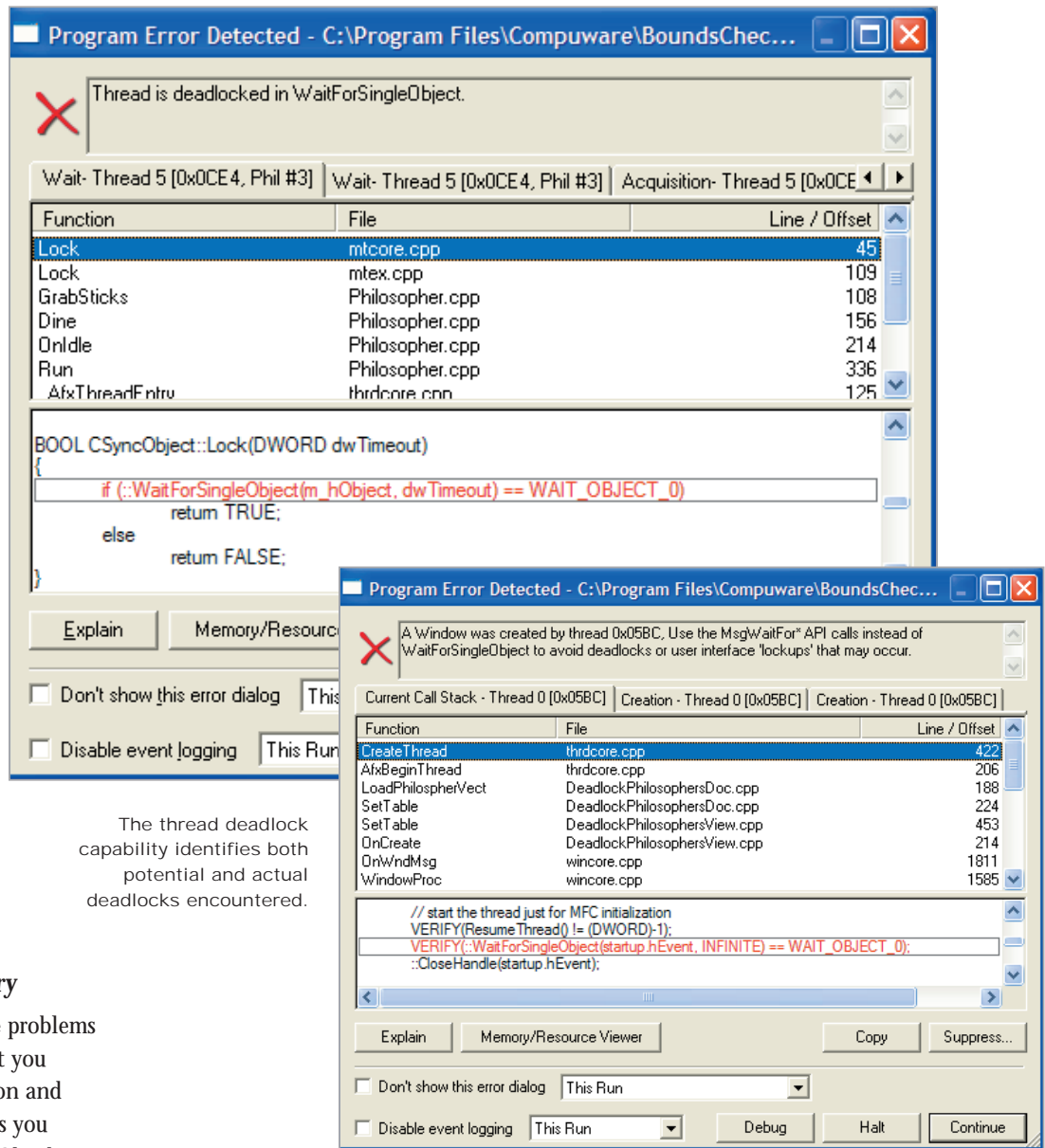
- exceptions thrown in native code that are passed into managed code
- Platform Invoke (P/Invoke) calls from managed code into native code
- COM calls from managed code into native code.

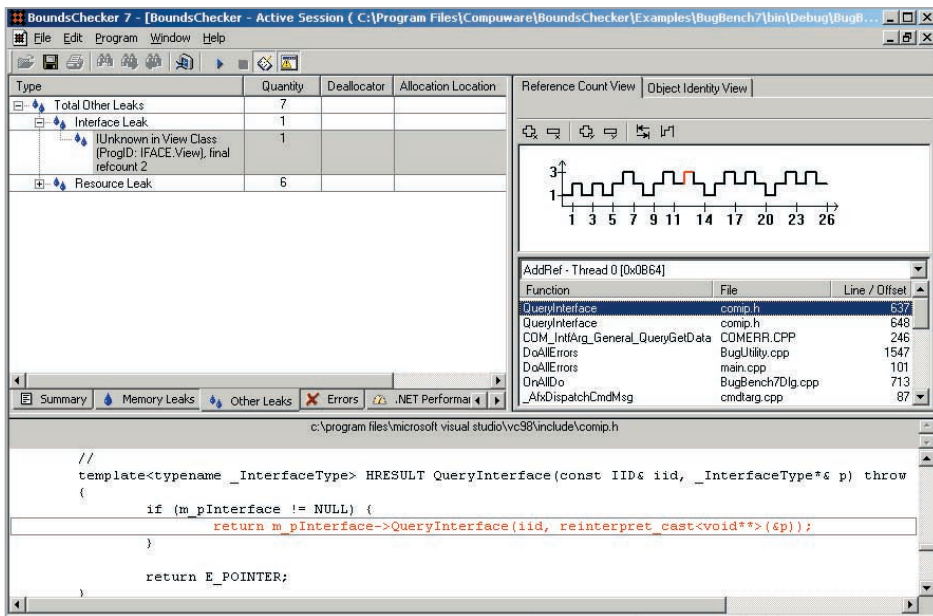
BoundsChecker distinguishes when a call is made, tracks the call, and then identifies and diagnoses any exceptions.

You verify a wider range of platform calls before faults corrupt the managed portion of your application.

Incorporate all-round error detection in the Visual Studio IDE

BoundsChecker is a natural fit for the Visual Studio Integrated Development Environment (IDE). Its versatile capabilities allow you to use BoundsChecker as a separate stand-alone component with Visual Studio .NET 2003, Visual Studio .NET 2002 or Visual Studio 6.





BoundsChecker monitors COM component calls and graphically displays missing or mismatched AddRef/Release calls.

Integrate BoundsChecker within the Visual Studio IDE and get seamless access to powerful error detection and debugging at any point in the development process.

Identify thread deadlocks easily

The use of threads in applications can be challenging, even for experienced developers. It's not always easy to predict when contention for shared system resources such as shared memory, variables, files, handles and Windows resources will take place. When two or more threads are waiting for resources that never become available, a deadlock occurs. BoundsChecker monitors all synchronization objects in your application and tells you if a deadlock potentially could occur. You're provided with detailed information describing how the deadlock will occur, making it easy to modify your code to prevent the problem.

Clear-cut, problem-oriented technical help at your fingertips

BoundsChecker comes ready to offer valuable tips and information about common and esoteric memory and pointer coding errors, Windows API calling parameters and Windows API support across Windows operating systems. With this built-in programming knowledge, you have expert assistance available 24 hours a day, seven days a week right from your own desktop. As programming problems occur, BoundsChecker locates them, recommends corrective actions and provides detailed information and examples to help fix coding errors fast.

Technical Requirements:

- Pentium III or Pentium 4, 500 Mhz PC or faster
- 256 MB memory
- 500 MB disk space
- Microsoft Windows Server 2003, Windows XP Professional Service Pack 1a, or Windows 2000 Professional or Server Editions Service Pack 3
- IDE integration requires Visual Studio 6, Service Pack 5, Visual Studio .NET 2002 or Visual Studio .NET 2003.

To learn more about DevPartner products, visit us at: www.compuware.com/devpartner

Compuware products and professional services—delivering quality applications

Compuware is a leading global provider of software products and professional services which IT organizations use to develop, integrate, test and manage the performance of the applications that drive their businesses. Our software products help optimize every step in the application life cycle—from defining requirements to supporting production service levels—for web, distributed and mainframe platforms. Our services professionals work at customer sites around the world, sharing their real-world perspective and experience to deliver an integrated, reliable solution.

Please contact us to learn more about how our comprehensive products and services can help your organization improve productivity, create higher quality applications and ensure performance in production.

