



The Software Developer's First Tool for Quality

"I expect MuTek's BugTrapper to help my team deliver the highest quality code efficiently. We tested BugTrapper with Microsoft® Word and BugTrapper traced its execution without any modifications to winword.exe. BugTrapper allows us to record and debug problems that are encountered outside our developers' offices. Reproducing each problem on a developer's machine won't be required, nor will hands-on debugging on the system demonstrating the problem, saving substantial time and money."

– Mike Kelly, Lead Software Engineer, Office Group,
Microsoft Corporation

"You have a wonderful product. BugTrapper should be on the shelf of every developer."

– Danny Schummer, Development Manager
Magic Software Enterprises

The software development process continues to be labor-intensive, unpredictable and, hence, prone to human error. Accordingly, software developers have sought tools to improve software quality and reliability, while reducing development and testing time. Similarly, development, quality assurance and support managers have sought integrated solutions to enable them to predict and shorten the software release schedule, while reducing development and product support costs. BugTrapper™ is a software tool designed for use from initial software development through quality assurance and software deployment. It significantly reduces troubleshooting and debug time by completely eliminating the tedious, difficult and time-consuming task of reproducing the user's actions, data, and environment.

When unexpected software behavior occurs, BugTrapper already knows what happened. Akin to a "black box" aircraft flight recorder, BugTrapper traps bugs by generating a record of how and where bugs occurred, providing a direct path to the cause of the problem behavior. As a result, users are able to deliver significantly improved quality and more reliable code, while reducing the software release cycle.

Bugs that occur in large Windows programs, Internet controls, time-dependent, or multithreaded applications create havoc for developers. With BugTrapper, the notion of unreproducible bugs does not exist. BugTrapper points to their precise execution path and the conditions under which they occurred. It generates a log of what happened inside the code during the final minutes before the bug occurred—serving as a remote sensor for the developer. Once the execution has been recorded, analysis and debugging can be performed off-line in BugTrapper's familiar debugger-like environment.

Most importantly, BugTrapper yields immediate results. It is an automatic tracing system that requires no preparation of application sources, object or executable files, eliminating the tedious manual entry of trace points into the source code. There is no need to recompile or relink the application in order to use it.

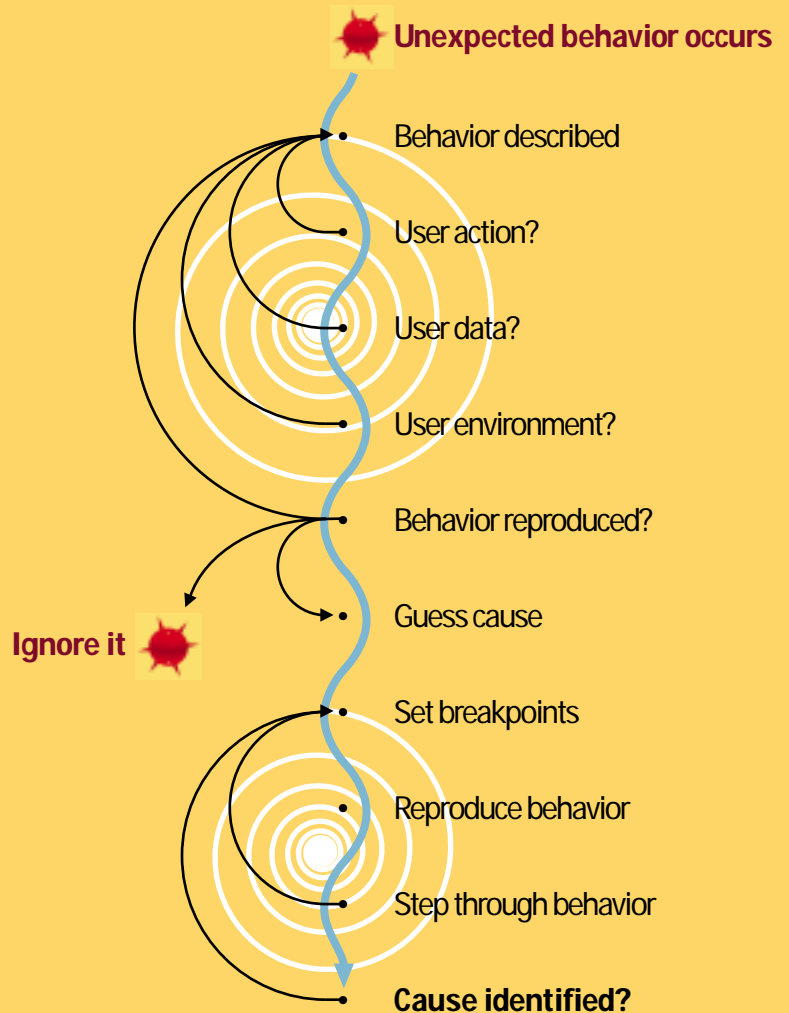
WHY REPRODUCE BUGS...

According to a leading authority on software quality, in small applications a single bug is introduced every 100 lines of code. In large applications, the bug rate exceeds 20 lines of code. Having to reproduce unexpected software behavior is a time-consuming, tedious and very difficult task.

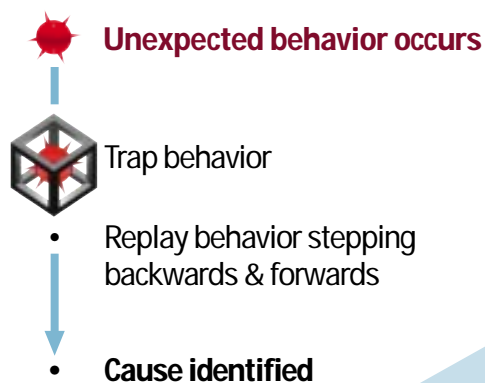
When unexpected software behavior occurs, software developers must reproduce the user's actions, data and environment. **Since no two runs are exactly alike**, the infinite process of "Run-Analyze-Iterate" begins. This usually happens because the developer does not know the exact steps that led to the problem. Certain information, for example the order in which menu items were selected, is in fact crucial to reproducing the bug. Furthermore, files and memory size may be different, system libraries such as databases may have different versions, and configuration information may not be identical. GUI applications and multiprocess environment add another layer of complexity to an already complex situation.

The "Run-Analyze-Iterate" process can take hours, days...even weeks. Some bugs are never reproduced, and consequently are never fixed. For example, parallel systems (multiprocess or multithreaded) and distributed systems are notorious for having bugs that can not be reproduced. They remain in the product as time bombs ready to explode, affecting the product's reputation for quality. This level of peril is only heightened when the software application is mission-critical.

Once a bug is reproduced, analysis takes place. The software developer is once again faced with a highly iterative process: guessing the cause, setting breakpoints then single-stepping through the code trying to understand what went wrong in the execution. There is no telling if inserting a breakpoint will capture the error or not. Therefore, the developer repeats the process until the cause has been successfully identified. The major problem with this approach lies in the fact that the execution that caused the problem in the first place, and the execution under the debugger, are very different. The tool used to analyze the run actually changes the run's characteristics.



Revolutionizing the Developer's Quest For Software Quality



...WHEN YOU CAN TRAP THEM

Because BugTrapper really does trap bugs, the 'Run-Analyze-Iterate' process is completely eliminated. Instead of trying to debug the program while running it, BugTrapper introduces a novel approach comprising of two steps: tracing and off-line debugging. BugTrapper provides the developer with all the information needed to fix the problem behavior. It generates a record of what happened inside the code during the final minutes before the problem behavior occurred, trapping the bug and its cause. In essence, it serves as a remote sensor for the developer, eliminating the need to reproduce the user's action, data and environment.

Once a bug is trapped, analysis can be performed off-line. The developer can view the trace information like a source code animation, scrolling back and forth along the execution path with no need to reproduce the bug in the application. The off-line debugging capability also provides a practical solution to the problem of remote troubleshooting by allowing the developer to analyze a log of what happened in the code at the remote site.

Once a bug is trapped, it cannot escape until it's fixed.

"I recently spent hours trying to find a very 'sneaky' bug using the debugger without success. I used BugTrapper, and after a few minutes I understood what the problem was and fixed it. GREAT tool!"

– Meni Hillel, Senior Software Engineer,
Mercury Interactive Corporation

"It's painful to remember beta testing before we used BugTrapper. Now I can see the exact flow leading up to the bug. We save time and money for our customers and for us. Way cool!"

– Ted Bardusch, R&D Manager, Tools & Products,
MAJIQ Systems & Software, Inc.

New Debugging Paradigm

BugTrapper was developed to take advantage of MuTek's unique hooking technology. It is able to introduce trace points to the executable by modifying it in a way that preserves its functionality. In this way, function calls, returns and any other source line that can be debugged can be traced. This instrumentation is done on the image of the executable in memory, and there is no need to preprocess or modify the executable file itself. Support is provided for analysis of multithreaded and multiprocess applications, dynamically loaded DLLs and spawned sub-processes.

BugTrapper contains a powerful filtering mechanism that allows the user to define exactly which points in the program to trace and what data to collect. The user can refine the filtering dynamically during the program run to focus on the behavior of interest.

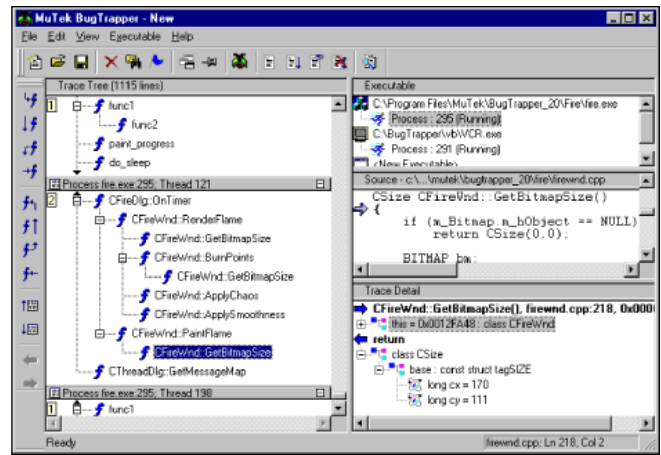
Log file browsing capabilities give the programmer the power of a debugger without the need to reproduce the bug. The user can step through the execution path, and even step backwards, to discover the original cause of the problem.

Key Product Features

- No preparation of application sources, object, or executable files
- BugTrapper Agent is used as a remote sensor for the developer providing remote troubleshooting capabilities
- Attach to/detach from process while running
- Works with Release-built optimized executables, as well as Debug version
- Supports *any* executable — including optimized builds, multithreading and multiprocessing, longjmps, signals, exceptions, recursion, etc.
- DLL support — Supports run-time loading (LoadLibrary), and components such as OCX, ISAPI DLL, or CORBA
- Quick logging of online trace information, with minimal impact on performance
- Data tracing — includes the value of all relevant variables, including arguments, return values, local and global variables in the trace information. The developer can decide which variables are traced when, with an intelligent default that is normally good for all cases
- Full stack information on crash: data is provided at the symbolic level

Supported Platforms

BugTrapper supports programs compiled by Microsoft Visual Studio™ version 4.1 and higher, and runs under Windows™ 95/98 and Windows NT™ version 4.0 and higher.



BugTrapper "trace tree" shows exactly what happened in the code just before a problem occurs. Context switches among processes and threads are clearly seen and can be easily analyzed.

"BugTrapper looks like the tracing tool of my dreams. Great for finding hard problems and it looks superior to a debugger for many easy problems, too. For our class of software—multiprocess, multithread message passing—it's clearly a superior product."

— Bill Ricker, Development Engineer
Avicenna Systems Corporation

www.mutek.com

North America Toll Free Sales
1-877-BUG-TRAP (284-8727)

North America Headquarters
MuTek Solutions, Inc.
1325 Borregas Ave.
Sunnyvale, CA 94089
USA
Tel: (408)822-5656
Fax: (408)822-5657
E-Mail: info_us@mutek.com

US East Coast
MuTek Solutions
222 Kearsing Parkway, Suite C
Monsey, NY 10952
USA
Tel: (914) 356-8747
Fax: (914) 356-8462

International Headquarters
MuTek Solutions Ltd.
24 Hacharoshet St.
P.O. Box 47
Or-Yehuda 60200
Israel
Tel: +972.3.6343223
Fax: +972.3.6343225
E-Mail: info@mutek.com

Germany
MuTek Solutions Ltd.
Rudolf-Diesel-Ring 9
83607 Holzkirchen b.München
Germany
Tel: +49.8024.30 45 80
Fax: +49.8024.30 45 83
E-Mail: info_d@mutek.com

Japan
TOYO Corporation
1-6, Yaesu 1-chome, Chuo-ku
Tokyo 103-824
Japan
Tel: +81.3.3279 0771
Fax: +81.3.5205 2030
E-Mail: ss_sales@toyo.co.jp

