PARASOFT®

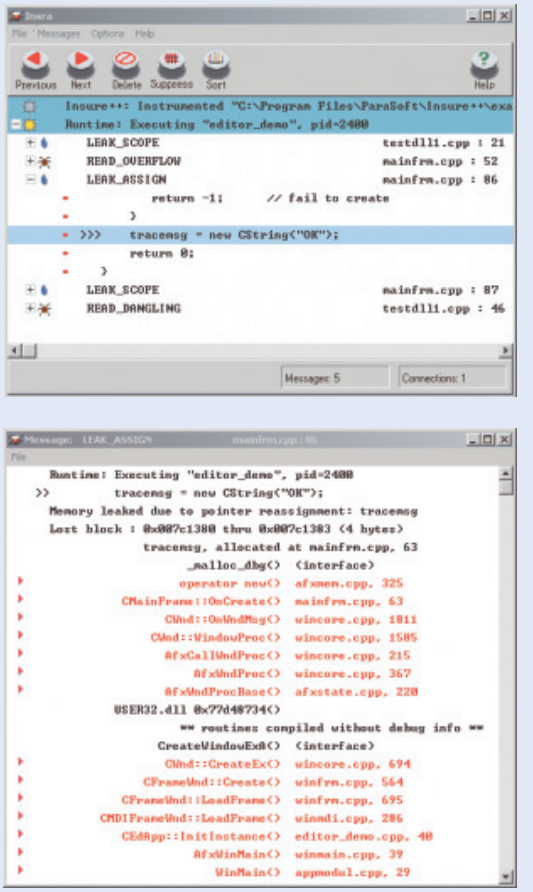
PARASOFT Insure++®

Insure++ is a runtime memory analysis and error detection tool for C and C++ that automatically identifies a variety of difficult-to-track programming and memory-access errors, along with potential defects and inefficiencies in memory usage. Errors such as memory corruption, memory leaks, access outside of array bounds, invalid pointers, and the like often go undetected during normal testing, only to result in application crashes in the field. Insure++ will help you find and eliminate such defects in your applications to ensure the integrity of their memory usage.

Errors Detected

During testing, Insure++ checks all types of memory references, including those to static (global), stack, and shared memory — both in user's code and in third party libraries. Errors that Insure++ detects include:

- Corrupted heap and stack memory
- Use of uninitialized variables and objects
- Array and string bounds errors on heap and stack
- Use of dangling, NULL, and uninitialized pointers
- All types of memory allocation and free errors or mismatches
- All types of memory leaks
- Type mismatches in global declarations, pointers, and function calls
- Some varieties of dead code (compile-time)



On Windows and UNIX operating systems, you can send error messages to Insra, the error display GUI, and then click to see full error explanations and stack trace information.

Multiple Use Modes

Insure++ has highly detailed memory analysis capabilities that are based on patented* source instrumentation algorithms. Source code instrumentation enables Insure++ to detect more error types than other memory error detection technologies, and also provides complete information indicating the root causes of the errors found, using a full database of program elements and memory structures.

There are two ways to use Insure++ for memory analysis and error detection. The first and most detailed analysis is achieved with **full source code instrumentation**. This requires that application sources be compiled and linked with Insure++, which generates its own instrumented files that are passed to the actual compiler.

The second option is **linking with Insure++**, which provides a trade-off between the extent of error reporting and the actual time to build and run an instrumented application. In this mode, Insure++ can detect and report most of the error types, including leaks, bad memory references, standard API usage errors, and so on.

Features

- Detection of memory corruption on heap and stack
- Detection of uninitialized variables, pointers, and objects
- Detection of memory leaks and other memory allocation/free errors
- STL checking** for proper usage of STL containers and related memory errors
- Compile-time checks for type- and size-related errors
- Runtime tracing of function calls
- GUI and command line interface
- Memory error checking in 3rd party static and dynamic libraries
- Direct interfaces with Visual Studio debugger

Benefits

- Finds memory errors before they become runtime problems
- Finds common errors during 64 bit porting
- Helps optimize memory usage of applications
- Reduces development and support costs
- Easily integrates with regression test suites in "smoke alarm" mode
- Provides detailed stack traces of errors to help understand their causes

TCA Test Coverage

- Calculates line and block coverage
- Reports line, block, class, function, and file coverage
- Text reports and interactive code browser with coverage highlighting

Inuse Memory Monitor

- Visualizes memory leaks
- Displays memory use in real time
- Helps correlate memory usage with program events

** Available for any Unix users with GCC 3.0 and above.

Parasoft® Inuse® and Parasoft® TCA®

Along with the runtime memory error detection engine, Insure++ includes two components that increase the tool's scope of analysis:

- TCA (provides total coverage analysis)
- Inuse (provides application memory usage analysis)

TCA analyzes and reports block code coverage and lets you get "beneath the hood" of your program to see which parts are actually tested and how often each block is executed. In conjunction with a runtime error detection tool like Insure++ and a comprehensive test suite, this can dramatically improve the efficiency of your testing and promote faster delivery of more reliable programs.

Inuse visualizes how an application uses memory. This component provides a graphical view of all memory allocations, over time, with specific visibility into overall heap usage, block allocations, possible outstanding leaks, and so on. By providing insight into an application's memory usage patterns, Inuse allows you to effectively analyze and optimize runtime memory usage and performance.

* Parasoft holds patents #5,581,696 and #6,085,029 for its source instrumentation algorithms.

Supported Platforms

Microsoft Windows (32-bit, 64-bit)
- Visual C++

Linux 32 and 64 bit
- GNU gcc/g++
- Intel ICC

Solaris UltraSparc Processor
- Forte Developer
- Sun Studio
- GNU gcc/g++

IBM AIX, PowerPC 32 and 64 bit processor
- IBM Visual Age
- IBM Visual Age (xIC compilers)
- GNU gcc/g++