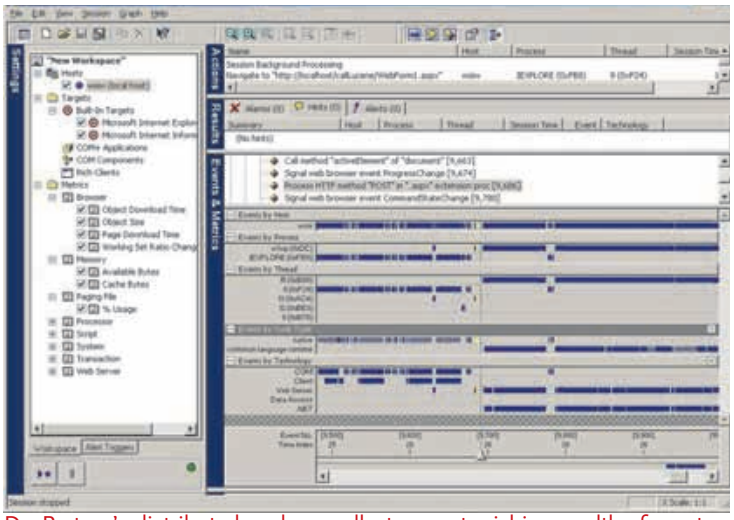
InfoWorld®
December 1, 2003 ■ ISSUE 47
GET TECHNOLOGY RIGHT

APP DEV

DevPartner Straddles Worlds

Visual Studio companion provides deep insight into .Net and Win32 code

THE TRANSITION TO Microsoft's brave new world of managed code will probably take a decade, during which time Windows programmers will struggle with the complexities of a hybrid managed/unmanaged environment. Instrumenting these very different programming environments — so that developers can analyze, profile, and more effectively debug programs straddling the unmanaged and managed worlds — is a big challenge that Compuware's latest offering, DevPartner Studio 7.1, tackles fearlessly.



DevPartner's distributed analyzer collects an astonishing wealth of event detail as you run an application. Here you can see the interplay between native and .Net code, the flow of events as the browser makes the request and the server handles it, and much more.

The DevPartner suite supports the following activities in addition to run-time error detection: source-code review, code coverage analysis, memory analysis, profiling, and analysis of distributed applications. To perform one of these activities, select it on the VS (Visual Studio) .Net toolbar, optionally adjust the instrumentation that DevPartner will insert (in the case of unmanaged code), then debug (or in some cases just run) the application and analyze results in the VS .Net document pane. Resulting data files accumulate, by activity type, in DevPartner's Solution Explorer.

To exercise the suite, I worked with open source indexing and search engine NLucene, which is a C# port of the popular Java-based engine, Lucene. I used DevPartner Studio to analyze the NLucene library, to analyze a program that indexes a batch of HTML files, and finally to analyze an ASP .Net application that searches that index.

The source-code review feature, available for C#, VB .Net, and VB6, scanned the C# files and produced a report that was sometimes pedantic, and sometimes cogent and educational. Warnings about recursion, hard-coded strings, or member-variable names fall into the pedantic category. Much more interesting are warnings that teach developers about platform subtleties. The warning ("Type not excluded from use by untrusted code") falls into this latter category. When an assembly is unsigned, as my build of NLucene was, you can still enforce trusted access by adding appropriate attributes to the source code. DevPartner identified the problem and illustrated the fix — a nice example of how an intelligent automated assistant can work with a developer to improve code.

Using the DevPartner profiler to explore the behavior of NLucene during the indexing phase, I uncovered no surprises but made some interesting observations. The split between time spent in application vs. system code, for example, was 75:25. Most of the 25 percent spent in system code was in the .Net run time (22 percent), only a little in the Windows kernel (3 percent). At a method level within NLucene, low-level I/O was the dominating factor. My use of an option to optimize the generated index (by merging segments) cost very little extra time.

A snapshot of memory use while running the indexer showed me how the optimizer's consumption of memory was spread among short-, medium-, and long-lived objects. I was also able to invoke the garbage collector and observe its effects.

Once the index was built, I created an ASP .Net-based Web form to search it. Then I fired up the distributed analyzer, which injected itself into the browser's process and also the Web server's ASP .Net process. From these vantage points, it collected an astonishing wealth of event detail as I ran the application (see screen shot). Here, as everywhere in DevPartner, vast quantities of data are reduced to a clean and effective set of linked information displays that are easy to understand, navigate, and customize.

Although DevPartner Studio's newest features are .Net-related, you can also still use all of its pre-existing features for analyzing COM and ActiveX code written in C or C++ and for validating Win32 API calls. The ability to explore both the .Net and "legacy" realms deeply — and simultaneously — is a powerful asset that's likely to remain valuable for a long time to come.

— Jon Udell

DevPartner Studio Professional Edition 7.1

Compuware compuware.com

VERY GOOD 8.3

Ease of use (30%)	7
Implementation (30%)	9
Integration (30%)	9
Value (10%)	8

COST: $1,495

PLATFORMS: Windows 2000, XP, 2003; IIS 5 or later; Visual Studio 6, .Net 2002, .Net 2003

BOTTOM LINE: DevPartner Studio 7.1 adds .Net memory analysis and enhanced .Net source code analysis to what was already a commanding suite of tools for the professional Windows programmer. The wealth of capability will take time to master, but will repay developers working across the full range of Microsoft programming technologies.