

# David Kopf Instruments

COBOL Analyzer supports an easy, fast, and cost-effective developer on-boarding process.



## Overview

David Kopf Instruments designs and manufactures a complete line of high quality stereotaxic instruments and accessories for neurophysiological research.

## Challenge

David Kopf Instruments is a long-established business. Because of the specialized nature of its manufacturing, decades ago it developed a solution to help account for labor and manufacturing costs. Over the years modules were added to the application, such as accounts payable and receivable and a general ledger. In 2007, the many development languages used in the system were converted into COBOL and the solution ran using OpenText™ Net Express. By now, the system had grown to 750,000 lines

**“Micro Focus (now part of OpenText™) made the whole process very smooth and seamless, with no disruption to our business at all. We feel COBOL Analyzer came along just at the right time to set us up perfectly for the future.”**

**JOANNE SIMMONS**  
IT Coordinator  
David Kopf Instruments

of code. A regular development effort ensures that the solution remains current in respect to ever-changing tax regulation.

When David Kopf Instruments’ main developer had plans to retire, the company realized that a new developer would need to be taught the ins and outs of a system which by now was 40+ years old. JoAnne Simmons, IT Coordinator at David Kopf Instruments, explains this challenge: “Even with a developer who knew the system inside out, making changes was never easy. He would need to work out exactly how changes could affect our system. If a mistake was made this could have dire consequences. A new person coming in to the role would not have the knowledge to make the right decisions and with no system documentation to refer to he would be totally in the dark. We realized we needed help in mapping out the code infrastructure of the system so that proper impact analysis could be done.”

## Solution

It was at this opportune moment that OpenText™ announced COBOL Analyzer. COBOL Analyzer gives developers the ability to continuously analyze their code before and after changes are made within their local environment and before committing those changes to the production system. It visualizes how code



## At a Glance

- **Industry**  
Manufacturing
- **Location**  
USA
- **Challenge**  
Gain visibility into the code structure of a decades-old COBOL solution to provide a smooth developer on-boarding process. Kopf Instruments wanted to conduct continuous code impact analysis which would make it easier to make changes to the system.
- **Products and Services**  
COBOL Analyzer
- **Success Highlights**
  - + Realized substantial time and cost savings with productivity in hours instead of months
  - + Easier on-boarding of new developers
  - + Enhanced system insight to facilitate future solution development

**“The information COBOL Analyzer provides can make or break a business. Within half an hour of installation I had a general synopsis of what I was dealing with and I could dive in very quickly without having complete business knowledge.”**

**ERIC DUCHACEK**  
Developer  
David Kopf Instruments

**Connect with Us**

[OpenText CEO Mark Barrenechea's blog](#)



components fit together, their dependencies and relationships and it provides visibility into the program structure. This identifies all code affected by the planned code change. Developers can then easily investigate and take next steps to complete the code change.

David Kopf Instruments participated in the COBOL Analyzer Beta program just as a new developer came on board. Eric Duchacek joined the team with a lot of COBOL experience and had in the past written his own programs to perform this kind of analysis into an unfamiliar system, as he comments: “In previous projects I would charge up to \$30,000 and spend at least four months mapping out the dependencies within the solution before I could really start any active development work on it. I had worked with Enterprise Analyzer—the Micro Focus (now part of OpenText™) solution to analyze and understand mainframe solutions—in the past and was pleased to discover its COBOL cousin.”

With some OpenText™ support, COBOL Analyzer was quickly installed and all the COBOL code was verified. For the first time, the code was visualized in a diagram view of the application which gave instant insight. COBOL Analyzer started generating reports immediately to show the exact relations between code components. An interactive tool displayed exactly what programs touch which files and how data flows within the system. Duchacek comments: “The information COBOL Analyzer provides can make or break a business. Within half an hour of installation I had a general synopsis

of what I was dealing with and I could dive in very quickly without having complete business knowledge.”

Duchacek needed to make a change to the date schema within the system. Using COBOL Analyzer he had an immediate overview of what would be impacted by this change, before drilling into the code.

### Results

COBOL Analyzer ensured smooth knowledge transfer from a long-established developer to one unfamiliar with the business and the system. David Kopf Instruments’ new developer was productive in hours instead of months by using COBOL Analyzer rather than a home-developed analysis solution or a reverse-engineering effort.

Simmons was pleased with the outcome: “Of course it helped that Eric is an experienced COBOL developer, but COBOL Analyzer definitely got us up and running much quicker than we had anticipated.”

She concludes: “We enjoyed the opportunity to take part in the COBOL Analyzer Beta program. Micro Focus (now part of OpenText™) made the whole process very smooth and seamless, with no disruption to our business at all. We feel COBOL Analyzer came along just at the right time to set us up perfectly for the future.”

Learn more at  
[www.microfocus.com/opentext](http://www.microfocus.com/opentext)