Todd Burch

Todd Burch uses z/XDC to fix bugs before the customers find them:

I've been writing in Assembler language since 1986, and writing seriously since 1990. I remember the first assembly I ran that received an RC=0. I was so excited. But all the blood left my face when I realized I was not finished with the code. Now that it assembled, it had to run too!

Before I knew z/XDC existed, (it was known as "XDC" at the time), I was writing progress messages, and traces and forcing S0C1 dumps (or, actually, not forcing them!) to look at the progress of my code when it did not perform as I expected. This worked, and still works, but the process is slow and tedious.

"I could now see my code execute, the registers change, the PSW get updated"

When I did learn about z/XDC, it was a true time-saver. Being the visual person I am, I could now see my code execute, the registers change, the PSW get updated and watch data moving around in memory from here to there. Using z/XDC took a lot of the mystery out of programming, especially when trying instructions I'd never used before.

"z/XDC saves real money towards the total cost of a project"

When writing code professionally, z/XDC saves real money towards the total cost of a project. You fix bugs before your customers find them.

I never realized how much I used and depended on z/XDC until I had to work without it. It was like writing blindfolded, or rearranging all the keys on my keyboard. I could not confidently say that the code was working as designed or if it was ready to ship. Hours had to be spent reviewing the code and going through listings. It was very time-consuming.

"While I might consider myself a seasoned user of z/XDC and have been very successful and productive with it, I still only use a subset of its full functionality"

I may not use z/XDC like most people, as I use it more to validate my programming than to debug my programming. I step through every single instruction (but not every time, mind you!). Once I verify that a macro expands properly, or have written a subroutine and I know it works, I move on. For new development, this is a progression of steps. For maintenance work, it is pointed and targeted validation.

www.colesoft.com
(800) XDC-5150 or 928-771-2003
bshimizu@colesoft.com

ColeSoft®

I use z/XDC for debugging too, and while I might consider myself a seasoned user of z/XDC and have been very successful and productive with it, I still only use a subset of its full functionality. It wasn't until recently that I started using Conditional Statements with breakpoints. WOW! The situation was an intermittent bug in an IPCS VERBEXIT I had written. IPCS was altering my access registers, and causing my VERBEXIT to ABEND S0C4. Using a Conditional Statement allowed me to let the program run through extensive printing data to IPCS (700,000+ lines) until AR15 was non-zero.

"To produce quality code, I rely on z/XDC"

I'm looking forward to using the new FRR/SRB support in z/XDC z1.7. Writing code in this type of environment (running APF-authorized in Key zero, Supervisor state, SRB mode with an EUT=YES FRR), you want to be sure you have got it right.

To produce quality code, I rely on z/XDC. When the code assembles, and even when it writes the correct data to the screen or in the report in the right spot, you don't know that you know that it's right until you've stepped through it. Just running the code isn't "good enough".

Todd Burch - z/XDC User

www.colesoft.com
(800) XDC-5150 or 928-771-2003
bshimizu@colesoft.com