



BMC Software says “Without z/XDC, debugging and problem resolution would be far more difficult and time consuming”:

BMC delivers software solutions that help IT transform digital enterprises for the ultimate competitive business advantage. We have worked with thousands of leading companies to create and deliver powerful IT management solutions. From system z mainframe to cloud to mobile, we pair high-speed digital innovation with robust IT industrialization- allowing our customers to provide amazing user experiences with optimized IT performance, cost, compliance, and productivity. We are constantly delivering new, innovative system z solutions that enable IT to provide business with critical high availability it needs in today's digital world. At the same time, our innovations help IT to reduce system z costs, making system z one of the most cost-efficient pillars for running the business. We believe that technology is the heart of every business, and that IT drives business to the digital age.

“z/XDC helps BMC to deliver quality enterprise solutions that exploit the latest features within z/OS in shorter time-to-market.”

“There is nothing like walking through your new code instruction by instruction, branch by branch”

BMC Software is a long-term customer of ColeSoft Marketing, Inc.'s z/XDC debugging product. BMC's engineers use z/XDC every day to refine and enhance BMC's extremely powerful solutions. z/XDC helps BMC to deliver quality enterprise solutions that exploit the latest features within z/OS in shorter time-to-market. Here is what BMC Lead Product Developer Ken Kornblum has to say about his experience with z/XDC.

I have been an assembler language coder since 1980, and have debugged programs in development using everything from WTOs, embedded S0C1s, SLIP traps, various flavors of DUMPs, and the occasional actual debugging tool. I have always been involved in system level code such as JES exits and anything I could get my hands on, but my greatest challenges came when I started working at BMC Software in 1996. During my years at BMC I have been involved primarily in the BMC Utilities for DB2 as a Lead Developer and as an Architect. These are best in breed utilities which we have always been proud to offer as having greater functionality and speed than the utilities which arrive with the database itself. More recently I have moved to the MainView Infrastructure group where I get to write code and APIs which are utilized by the entire suite of products in the Mainview family (and some that are not!).

ColeSoft's z/XDC is used at BMC by nearly all of the assembler developers. Most of us use z/XDC as both a debugging and a testing tool. There is nothing like walking through your new code instruction by instruction, branch by branch, or through complete function calls by the press of a PF key or command. I have seen code which works and would have passed QA reveal flaws by stepping through decision points and having new conditions come to mind which otherwise might not.

“z/XDC's real power is in the great level of control it gives you”

But z/XDC's real power is in the great level of control it gives you over where you want execution to be paused to enable developers to inspect control blocks and program flow without having to abend the program. These methods include Transient break-points which disappear the first time they are hit, Permanent break-points which remain in place until you remove them, Conditional break-points which may be defined with

complex conditions to be met, or even Deferred break-points you can set in programs which have not even been loaded yet.

At any breakpoint the developer has the ability to view any storage available to the program including other address spaces and data spaces using your established ALETs. The developer can bring up a list of tasks (TCBs) and drill down to any level desired, displaying registers, save area chains, and linkage stacks along the way for any task at any level.

Program source and control blocks can be formatted for the developer to debug-over-source and to view variables and their contents along with metadata such as length and data type. Depending upon some of the resources available to you formatting can be simple assembler CSECT and DSECT maps, Symbol Level maps, or full blown Source level maps.

z/XDC can be used in a wide variety of environments. It can of course be used in TCB mode, but also SRB mode, within an ESTAE, exits, FRRs, PC routines including space switching routines, you name it.

“Without z/XDC, debugging and problem resolution would be far more difficult”

Without z/XDC, debugging and problem resolution would be far more difficult and time consuming. I know I share with many of our developers the sentiment that we would be at a huge disadvantage if z/XDC was ever removed from our “tool belt” of debugging wizardry.

I have made a great deal of use in the past few years of z/XDC's REXX implementation. I'm the type who enjoys writing REXX tools for repetitive tasks, and when I read the help screens describing the built-in REXX support I had hobby projects lined up to make use of it. z/XDC REXX gives developers the ability to run REXX EXECs within the XDC environment while at a breakpoint or any time you are in control at the command line. As you might imagine, given REXX's affinity for the ISPF and TSO environments, XDC must establish an environment for REXX programs and supports only a subset of REXX commands. However, from the z/XDC-REXX environment there are XDCSERVICES available for the developer to invoke z/XDC's internal services. The developer can also author his own user functions.

My favorite z/XDC-REXX lists all of our utility control blocks in a tree-like structure along with their names, storage address (31 or 64 bit), some basic attributes and linkage to other control blocks. The developer can issue any valid XDC line command such as MAP or DISPLAY in the command prefix area in front of every object displayed.

BMC Utilities for DB2 and our MainView products make extensive use of 64-bit storage (a.k.a. storage above-the-bar). z/XDC was ready for this and gives us several options for the display of 32 bit and 64 bit registers as well as the 'bang' character we wish to use to designate 64-bit storage addresses in commands and storage displays.

"I would find my job much more difficult to manage if z/XDC was not available to assist me"

As stated earlier, I would find my job much more difficult to manage if z/XDC was not available to assist me with development and debugging tasks. Having used z/XDC so much and for so long I'm pretty sure I could teach this tool to anybody who is willing to learn. It has a very complete HELP system which also serves as the product documentation. I have seen an occasional developer follow the instructions to print the documentation, but to their dismay it is a huge amount of paper. I recommend using it online!

"z/XDC REXX gives developers the ability to run REXX EXECs within the XDC environment"

Kenneth Kornblum

BMC Lead Product Developer, MainView Infrastructure and RTCS Development