# AuthXDC®

## Unmatched Visibility Across Authorized Environments – Only With AuthXDC

AuthXDC is an optional extension to Z/XDC, purpose-built to address the unique challenges of writing software that executes in authorized or constrained environments. It enables software development teams and SYSPROGS to efficiently diagnose and resolve issues in systems where security or system restrictions normally prevent interactive debugging.

### Why You Need AuthXDC

AuthXDC eliminates these barriers by extending Z/XDC's interactive debugging capabilities into authorized and constrained environments—giving you complete visibility without compromising security or stability.

**APF Authorized Programs:** APF-authorized programs can perform privileged operations that problem-state programs cannot. Without AuthXDC, debugging stops when your code becomes authorized - leaving critical functionality invisible.

**Programs with Execution Keys 0-7:** Execution keys 0-7 provide access to protected system storage. Debugging code with these keys requires specialized tools that understand and respect key-based protection mechanisms.

**Supervisory State Routines:** Supervisor state code can execute privileged instructions and access protected system areas. No other debugger can operate in this mode - AuthXDC can.

**Cross memory Environments:** Modern system software often operates across multiple address spaces. AuthXDC lets you follow execution flow seamlessly as your code switches between Primary, Secondary, and Home address spaces.

### With AuthXDC, there are no debugging dead zones.

Follow your program's execution path wherever it leads - from problem state through authorized code, across address spaces, into supervisor state, and back - all within a single, continuous debugging session.

# AuthXDC®

**Step 1: Start with Z/XDC**

Begin debugging your program with standard Z/XDC capabilities. Set breakpoints, examine variables, and step through code as usual.

**Step 2: Automatic Authorization Detection**

As your program transitions to authorized code, AuthXDC automatically activates, maintaining your debugging session without interruption.

**Step 3: Full Authorized Debugging**

Continue using the same Z/XDC commands and interface while debugging authorized code. All Z/XDC features remain available - breakpoints, tracing, variable inspection, control block formatting.

**Step 4: Seamless Transitions**

Follow execution flow naturally as your program moves between:

- Problem state ←→ Authorized state
- Different execution keys
- Cross-memory environments
- Supervisor state and back
- SVC/PC/SRB routines and their callers

**Step 5: Complete Context**

AuthXDC maintains full debugging context across all transitions, so you never lose sight of where you are, how you got there, or what's happening at each step.

AuthXDC respects all security boundaries while providing complete visibility. Your system's security posture remains unchanged - AuthXDC simply makes authorized code debuggable without compromising protection.

## Why Choose AuthXDC?

AuthXDC extends Z/XDC's capabilities for application development into system-level programming where no other debugging tool can operate.

Leverage AuthXDC for end-to-end visibility to reduce risk, to understand code behavior before making changes, test modifications interactively and verify accuracy immediately.

Prevent production incidents by thoroughly testing authorized code in realistic environments before deployment.

## Ready to Debug Authorized Code Effectively?

Stop struggling with blind spots in your most critical system-level code. AuthXDC extends Z/XDC's powerful debugging into authorized environments, giving you complete visibility without compromising security.



See AuthXDC debug authorized code seamlessly; schedule a live demo today!

**sales@colesoft.com**