HISTORICAL RE-ENACTING WITH THE MAINFRAME GREEN SCREEN

When I see someone staring at a green screen and coding, I get the same sense as when I watch a historical reenactment: a strong sense of how things were. Nostalgia is fine, but that doesn't mean we should continue living in the past.

When I see someone staring at a mainframe green screen and coding, I get the same sense as when I watch a historical reenactment: a strong sense of how things were. Nostalgia is fine, but that doesn't mean we should continue living in the past.

Despite my long career as a mainframe software developer, my college degree was in history. I love history and I love to experience it. Actors in historical re-enactments share my passion for history and go to immense efforts to accurately portray a time. For a day or a weekend, they stay in character and try to live in the past.

The mainframe has progressed in the decades since I first became a mainframer in the '80s, and so have mainframe development interfaces. While we have gained some desk space with a thin monitor instead of a massive 3270 terminal, how we work is, in many ways, unchanged from the '80s, and the continued use of the green screen ISPF is a sign of it.

Developing Without the Mainframe Green Screen

When we ask the millennial generation of developers to join our ranks in the mainframe, think of how the green screen looks to them. We are saying that to work on the mainframe you need to dig into history and experience life in the '80s. No wonder we get the reactions we do.

Instead of using the mainframe green screen, we should fast-forward a few decades and make a modern Eclipse-based development interface—an IDE they already know—the standard on the mainframe. With tools like this, mainframe-inexperienced millennial developers can pick right up and start to work on a piece of software or an application that formerly would have required esoteric knowledge.

When you empower the next generation to be productive on and excited about the mainframe with the modern tools they're used to, you end up with enthusiastic future leaders.

And while you're busy empowering millennials with an interface they're used to and can be productive with, it might be a good time for veteran mainframers to start working with a modern development interface too, saving the green screen for times they feel nostalgic.

Treating the mainframe like a historical reenactment with the green screen might be a good history lesson, but it isn't the way to work today, especially when your organization is trying to attract new talent, move faster with Agile and DevOps and compete in the digital economy against disruptors. With what is for all intents and purposes an ancient interface, the green screen unfairly obscures the advancements mainframe hardware and supporting software have made.

My recommendation is to glance at the past for guidance but develop mainframe software as it should be done in the present, always looking to the future for innovation. If you're looking for an IDE to empower your mainframe developers with, take a few minutes to learn about BMC AMI DevX's Eclipse-based IDE, **Topaz Workbench**.