

WHAT WILL YOUR MAINFRAME LEGACY BE?



If you [Google](#) the word legacy, a mere noun since time began, you'll find it now has an additional part of speech, an adjective with this definition: "denoting or relating to software or hardware that has been superseded but is difficult to replace because of its wide use."

Ho! Someone was very clever there, but only clever enough to memorialize the pejorative mainframers have suffered for nearly two decades. I recall the standard response when disparaged thus was something like, "Oh, yeah, legacy: a system that works." And we smugly went about our business, assured we had righteously trounced the impertinent offender.

However, scoffing at naysayers never solved the actual problem of misinformation around the mainframe, which continues to persist today despite the platform being the most modern system of record available to large organizations that require a highly available, reliable and secure environment through which to run their mission-critical enterprise applications and data.

The True Legacy of the Mainframe

Having had our laugh, mainframers must now focus on promoting and justifying the real meanings of the word legacy:

- Something of value left to someone
- A thing handed down by a predecessor

I especially like some of the synonyms for these definitions when referring to something of value, particularly:

- Heritage

- Endowment
- Gift

These synonyms really make me think about the mainframe as more than just a platform and instead as an ecosystem of continuously evolving value, experiencing a massive positive transformation to its culture, processes and tools, all of which will soon be left to a new generation of stewards with new talents and ideas. That's why these synonyms also make me think of my career as a mainframe infrastructure technician and manager.

What Will Your Legacy Be?

I've observed there are two sorts of people on final approach in their careers:

1. The kind who are watching the clock, coasting out, doing "good enough for government" work; for these, retirement cannot come soon enough.
2. The kind who still have fire in their bellies, are still trying to top their personal best, are still fascinated with the technology, are wanting to start something new and want to see if they can make the latest elephant dance.

The second group of people are the blessed who have found something they love and have hardly worked a day in their lives because of it. Many, if not most, started their careers this way, I believe, but somehow lost the gift. And who can blame them with all the shenanigans and indignities that some businesses have foisted upon "their most valuable resources"?

But does any mainframer really want to go out like the first group of people: apathetic, discontented and perhaps bitter? I doubt it. I know I surely do not!

How do you want to be remembered? What do you want to see when you look back on your career? What about those you leave behind? In the remaining time you will work, what about your goals and aspirations? Don't let your career as a mainframe systems programmer, performance tuner, capacity planner, hardware specialist, job scheduler, or operator simply become a job.

The business we're in has literally advanced humankind and drives us, as a people, into the future. We who work with it all share a common fascination for the technology, and we revel in how we take part in that advancement. In so doing, it makes us part of something larger than ourselves.

In my case, I'm proud of [my participation and my contribution](#) to this advancement. And I want to hand down a heritage to those who come behind me. It's incumbent upon us older mainframers to endow our successors, and those with whom they work, with all the knowledge and guidance possible to prepare them to carry forward.

What we do now, or don't do, will have consequences for the future. I, for one, don't want a lack of action or failure on my part to have any negative repercussions. And I surely do not want my legacy to be described with any words like "aftermath," another synonym attached to "legacy."

As I approach my career end, I think a lot about those taking up the mantle I will lay down. How can I help future mainframers succeed? One way would be to automate the seemingly last bastion of manual operation: JES2, where IBM provides a strong spooling facility, but leaves it to the customer's brain trust to exploit as well as they can.

Leaving a Proper Mainframe Legacy

By and large, our successors don't have the fundamentals most of us have. They may have never experienced a wait state during an IPL (thank goodness). They may not have learned assembler or troubleshoot a dump, nor juggled initiators or fetched and mounted tapes. Their roles begin in the second act.

Fortunately, IBM has done a good job making the base stable. So, the new mainframers can jump into the middle of the technology and be successful. But there is plenty we can do to help further position them for success.

Automation is a big one, particularly [automating batch processing](#). Our beloved spooler JES2 is a real workhorse, but it is still, at the end of the day, a manually operated system. Somebody must configure it, evolve strategies for its best use, and monitor and manage it continuously, especially during peak batch processing windows. Why batch has not been automated by IBM is a mystery.

But there is good news. [BMC AMI Ops Automation for Batch ThruPut](#) takes this challenge head-on and solves the problem from soup to nuts, endowing the complete automation of the batch workload upon the harried operations area. Our rules-based, policy-driven control system automatically manages all facets of batch processing, from job entry through execution.

Using our rules-based, Job Action Language (JAL), shop standards are easily enforced. And the supporting code can be quickly adapted to meet the installation's changing needs. No more inertia, passing up opportunities to evolve and improve because it is too difficult to write or modify static system exits. BMC AMI DevX has brought agility to batch management.

Our policy-driven components replace manual initiator management, negating the need for continuous operator attention, monitoring and manipulation of initiators and job classes. Batch just runs—optimally—with constancy and predictability. This frees operators from these and other mundane tasks allowing them to focus on serving customer needs such as exception handling, demand processing and one-off requests.

I've worked with ThruPut Manager for over 25 years as a mainframe technician, to an operations director, to a field engineer purveying the product for BMC AMI DevX. Using it at a previous employer, we went from out-of-control to full control, from piecework to assembly line, from clunky to streamlined, and from reaction to proaction with this systems software. There, we no longer simply had only the responsibility for batch, but we had the means to manage and control it as well.

When you finally step down from your role of consummate expert, how will you be remembered? There is no better heritage to bequeath, no finer gift to give to future mainframers, than automation of your batch workload.