bmc

Solution White Paper

Operational Instrumentation Even Developers Can Love

Amplify your CI/CD delivery pipeline by leveraging a Jobs-as-Code approach

Table of Contents

Executive Summary

The ideal state for automated delivery pipelines is that all manual or creative work is done at inception with subsequent stages fully automated. This paper takes a simple position: application automation that runs and monitors business applications (whether you call it operational instrumentation, workflows, or jobs) is part of the application and should also be "developed" or coded as early in the software development lifecycle (SDLC) as possible, just like all other application components. In the past, it was difficult to take this approach if you used traditional enterprise solutions because they were built with operations in mind, not developers or DevOps engineers.

By taking a Jobs-as-Code approach, developers can build in operational automation at the same time they are developing business logic. Jobs-as-Code lets developers build automation the same way and with the same tools they build Java or Python code. Control-M Workbench and Control-M Automation API give developers a sandbox to build, test, and debug jobs (operational instrumentation) using their favorite tools on their own laptops or development environments.

This white paper provides:

- An overview of the Jobs-as-Code approach, plus analyst insight on why it is valuable
- Examples of enterprises that have used it successfully
- An introduction to Control-M Workbench, a complete development environment that makes it easy for developers to embed workflow automation into their jobs
- A look at how Control-M Workbench can accelerate application delivery by up to 20 percent, run in production with up to 25 percent fewer incidents, require up to 50 percent fewer FTE resources, and result in up to a 20 percent lower MTTR

INTRODUCTION

There's an argument that developers don't know operations so they can't create operational instrumentation. That argument is half true. Developers are NOT operators.

But they are also not shoppers or investment bankers or chess players. Yet it is developers who build systems and applications that do all these things.

You may argue that developers would rather build chess programs or Grand Theft Auto, rather than operational stuff, and you may have a point. Developers would rather spend their time working on cool stuff, which is exactly why they should use Control-M to build in automation by creating Jobs-as-Code, instead of writing every. single. script. from. scratch.—every single time!

The reality is that developers build at least some, perhaps much, of the automation today. They write scripts, embed operational functions in their code, and choose tools like Jenkins, Cron, or others to do it. Why?

Some of the factors that contribute to this behavior include:

- **Ignorance of alternatives** – It's what they know.
- **Positive environmental influence** – It's what their co-workers and buddies do.
- **Negative environmental influence** – Ops/IT won't let them play with their environment or make it really hard to do so.
- **Delusions of grandeur** – Developers believe they know how to do it best, and no one is going to tell them anything better.
- **Practicality** – Developers need to build their stuff fast. They're on the hook and there's no one they can really trust to either do it well or not slow them down.

The development community is proving there is a better way to write apps. It's called continuous integration and continuous deployment (CI/CD) or DevOps. Why wouldn't these approaches work for automation too? The answer is they absolutely can, and you just need a few bits and bobs.

WHAT IS JOBS-AS-CODE?

Jobs-as-Code builds many of the components needed to run a workflow into the workflow code itself. It gives developers the ability to create workflow artifacts the same way they develop code, using the same tools. Jobs-as-Code is a shift-left technique that pushes workflow automation and execution earlier in the development cycle.

Benefits to the Jobs-as-Code approach include significant time savings during workflow creation, coding, testing, debugging, and promotion to production. In short, it increases code quality while shortening delivery time.

Julie Craig, application management research director at Enterprise Management Associates, defines Jobs-as-Code this way in the report Jobs-as-Code Approach Infuses DevOps Focus into Job Scheduling, Workload Automation:

> **"The Jobs-as-Code concept allows Development to encapsulate granular knowledge about applications and services into the programming environments they utilize as part of their day-to-day work. This "shift left" in terms of work stream specification means that the steps required to execute a given job become part of the code itself."**

Jobs-as-Code is valuable for any development environment, but is especially valuable for organizations that utilize a DevOps approach to application development.

For more information about the Jobs-as-Code approach, see BMC's white paper How to Accelerate DevOps Delivery Cycles.

WHAT'S NEEDED?

Four elements are essential to make it easy for developers to build automation into their code:

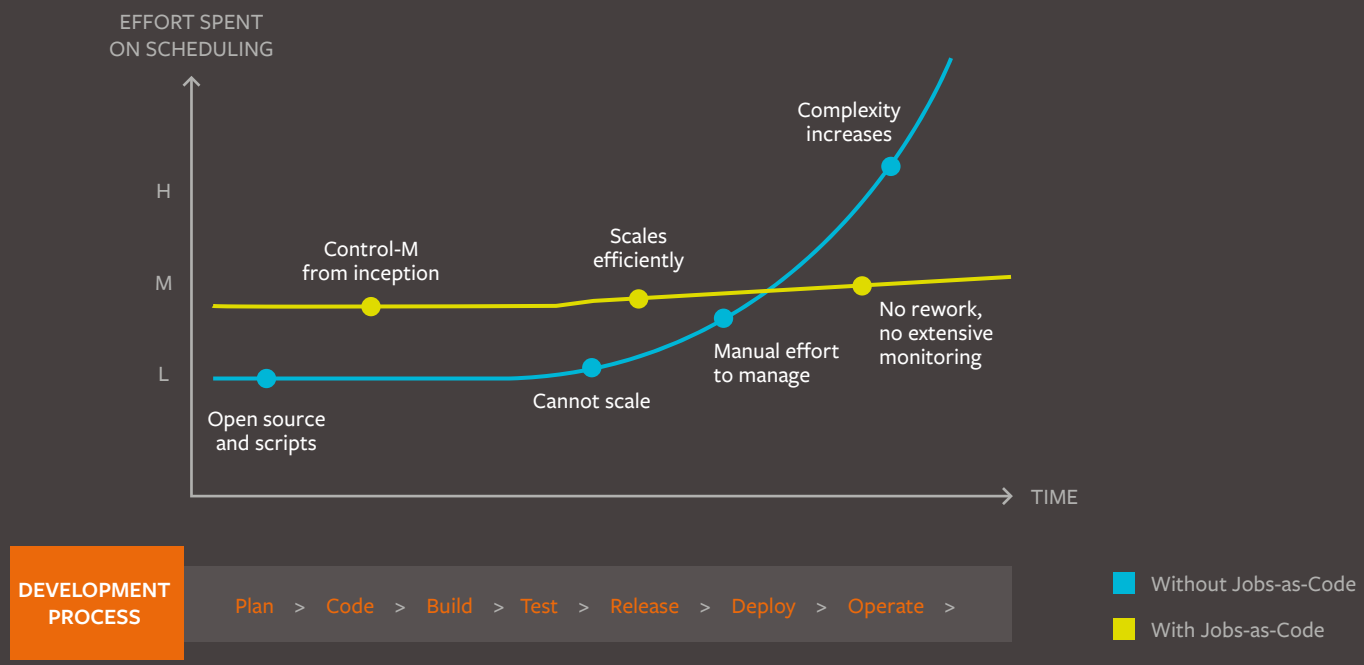
1. A "language" for building automation that is similar to the language used for building business logic
2. The ability to store and manage automation code together with all other application artifacts
3. Support for whatever toolchain is used to build, test, and promote other application artifacts
4. Developer tools for automation "coding" similar to those used for business logic coding

The first three elements can be satisfied with packaged interfaces, such as Control-M and its Automation API, which is a set of programmatic interfaces that let developers and DevOps engineers use Control-M within the agile application release process. **What's been missing are familiar, easy-to-use tools that make it as easy for developers to create automation as it is for them to create code. Control-M Workbench fills that void**.

WHAT IS CONTROL-M WORKBENCH?

Control-M Workbench is a compact virtual appliance available as a free download from github (controlm.github.io). It is a complete, standalone development environment that lets users code, debug, and test job flows without requiring any additional services. It provides developers a ready-to-use Control-M sandbox as a virtual appliance that runs in Apple® macOS®, Microsoft® Windows®, and Linux® environments. It can be downloaded and running in less than ten minutes and provides a nearly complete Control-M stack that supports all of the same artifacts as the full version of Control-M. Using Control-M Workbench removes the tooling barrier between development and operations teams, which helps prevent bugs and delays and enables closer collaboration.

Shift Left with a Job-as-Code Approach with Control-M



Control-M Workbench makes it easy for developers to embrace Jobs-as-Code because it allows them to work in their familiar environment using the same tools (including Git, Jenkins, Chef, Puppet, Mocha, Robot, etc.) they use to build and test applications. Because Control-M Workbench is a publicly available sandbox they can run on their laptops, developers and engineers are freed from any concerns about access, ownership, or licensing.

Control-M Workbench provides JSON notation for coding all the definitions required to create application jobs and exposes Control-M functionality as a RESTful endpoint. This approach enables all application artifacts, including the operational automation that will run applications in production, to be stored together in an SCM like Git. A comprehensive set of services are accessed via REST web services providing validation, testing, configuration, and deployment of workflow scheduling

artifacts within the enterprise's automated application release and deployment process. Developers can do syntax verification, agent provisioning, configuration management, code deployment, and other functions. The environment supports file transfer, relational databases, the Hadoop ecosystem and all other types of enterprise workloads.

Because Control-M Workbench is easy to access, developers are freed from any reliance on either the IT or operations organizations to develop, test and debug automated workflows. With Control-M Workbench, developers and engineers who have no previous knowledge of Control-M can run jobs within ten minutes of completing the download. And once their workflows are built, they are completely compatible with and can be deployed into any enterprise Control-M environment.

"With its latest Jobs-as-Code functions, BMC has introduced significant new features to the Control-M solution that benefit both current customers and prospective buyers. A product formerly aimed primarily at operations can now be used—essentially free of charge—by development as well. These new capabilities provide a basis for achieving the true purpose of DevOps, which should be cross-functional collaboration, wherever it is needed and at any stage of the application lifecycle."

EMA Associates
Jobs-as-Code Approach Infuses DevOps Focus Into Job Scheduling, Workload Automation

See **Control-M Automation API, Docker, and Microservices** for more technical information and additional Jobs-as-Code customer profiles and use cases.

PROOF POINTS

Carfax, which provides information services to millions of used car buyers and dealers, switched to the Jobs-as-Code approach to support its DevOps efforts. Carfax processes more than 129,000 workloads daily across 350 batch nodes, manages data from 34,000 sources and has more than 13 billion records in its database. Its environment is decidedly large scale and high speed. It has also been a proving ground for the Jobs-as-Code concept. Here's what the company says:

"A Jobs-as-Code approach is paramount for anyone doing agile development and DevOps. We have been using Control-M for years in operations, and now the product gives our developers full ownership and control of their jobs in a coding environment that is familiar to them, so they can define the business processes they want to automate in production."

Robert Stinnett, Automation Analyst, IT Operations, Carfax

A Fortune 500 company took the Jobs-as-Code approach for jobs that run in Control-M. Its code is being written in JSON and existing flows are constructed in XML. Changes that previously took hours or days are now implemented in seconds or minutes. The team even packaged the Control-M Automation API command line interface into a Docker image so that new users can be onboarded almost instantly without having to install or configure anything before starting with Control-M. The result? "The tool has empowered developers to own their own Control-M work without engaging another team. They can make changes that used to take weeks in a matter of minutes. It's easy to record, and there's no more hand holding by the Ops team," said one of the

project leaders. “This is appealing to leadership because it fits in with CI/CD goals and the ability to represent everything as code. We were blown away.”

CONCLUSION

DevOps is delivering many benefits but is still plagued by cultural, collaboration, and tooling problems. Tooling challenges, in particular, leads to delivery delays because of the time required to debug jobs executing in widely diverse development, test, and production environments. Now organizations can avoid these delays by giving developers the tools and access they need to make their code production ready from the time it is created, with JSON, Git, Jenkins, Docker, and other tools and technologies they already use. Because Control-M Workbench delivers these capabilities without requiring developers to learn a new environment or do a lot of scripting, it reduces the overall time required to get new services into production. Control-M Workbench is a natural complement to DevOps, and an asset to any organization that wants to take time, errors, and effort out of its workflow development and execution processes.

FOR MORE INFORMATION

To learn more about how Control-M Workbench can support your DevOps efforts and streamline your development and workflow management, visit **bmc.com/it-solutions/control-m-devops.htmlAbout BMC**. Download Control-M Workbench at **controlm.github.io** and try it for free.

498629