bmc helix Control-M

White Paper

How to Orchestrate Data Workflows with the BMC Helix Control-M Python Client

Table of Contents

Executive Summary

Data science and engineering teams use many tools today and spend a lot of time wrangling data and managing data pipelines. In fact, data scientists spend more of their time on data preparation (22%) than any other task, plus another 17% on data cleansing according to the Anaconda 2021 State of DataOps report.[1] After these front-end activities are completed and data teams finally get to develop, deployment takes up another 11% of their time. **That adds up to data scientists and engineers spending half their time on data and development pipeline activities**. Imagine what these teams could do if they could focus on delivering data-driven insights one hundred percent of their time.

To make the processes more convenient (and hopefully save time), organizations give these teams wide latitude to choose tools they are familiar with. And because more data sources and related tools are available than ever before, there is more diversity (and more silos) across the enterprise development environment. Unfortunately, that means over-burdened operations staff need to learn each tool and build documentation, standards, and complete other activities for each.

Now enterprises that use BMC Helix Control-M for application and data workflow orchestration can use Python as the bridge to faster workflow promotion, smoother execution in production, and for enabling data development scalability. BMC's new Helix Control-M Python client enables data engineers and developers to easily build automated execution into their workflows as they develop them with Python and their other preferred development tools and environments.

Python has clearly emerged as the favorite development language – according to Anaconda's 2021 State of Data Science report, 34% of respondents said they always used Python in 2021.[2] Plus, 63% say they always or frequently use Python, which compares to 27% for R (which ranked second).

Did You Know?

Managing dependencies and environments is the top challenge for 23 percent of respondents to the Anaconda 2021 State of Data Science Report. Among developer respondents, 30% say it is their top challenge.

[1] Anaconda "2021 State of Data Science" accessible at https://www.anaconda.com/state-of-data-science-2021.
[2] Anaconda "2021 State of Data Science" accessible at https://www.anaconda.com/state-of-data-science-2021.

The Challenge: Going from Data Wrangling to Insights

While the diversity of development and data pipeline tools may be good for data scientists and developers, it presents integration and orchestration challenges for the operations team. The automation and management features available for some data workflows can't be applied to all. That hinders orchestration and visibility, and ultimately limits deploy and production scalability. Without automated orchestration, IT Ops needs to find workarounds to view, schedule, and monitor application and data workflows, manage the dependencies, arrange necessary file transfers and ETL operations, issue alerts, and more. That makes it difficult to manage the workflows and their dependencies effectively and to get a complete end-to-end view of workflow status.

The sooner that orchestration is addressed in the development pipeline, the sooner workflows can run in production.

> As Gartner® says: "Data orchestration systems allow composing these complex workflows, scheduling, and executing them reliably at scale in production. A large part of a data engineering team's focus should be on orchestration and automation, a prerequisite to building, running and deploying data pipelines at scale." [3]

About BMC Helix Control-M

Helix Control-M is not as well known among data science professionals and software developers as it is in the IT operations world, where it was recognized by Enterprise Management Associates in 2021 (see the details here). Helix Control-M simplifies application and data workflow orchestration, making it easy to define, schedule, manage and monitor workflows, ensuring visibility and reliability, and improving SLAs. It has unrivaled deployment speed, being delivered as a service.

It integrates, automates, and orchestrates workflows on-premises, and in public and private clouds, so jobs and business services are delivered on time, every time. With a single unified view, users can orchestrate all their workflows, including file transfers, applications, data sources and infrastructure with a rich library of plug-ins. Built on AWS, Helix Control-M leverages the ephemeral capabilities of cloud compute services.

Visit our website to learn more about BMC Helix Control-M, including integrations, and how it supports the Jobs-as-Code approach to streamline development and operations.

[3] Gartner "Gartner®, Data Engineering Essentials, Patterns and Best Practices", May 2021. GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.

But the data engineering team does not want to focus on orchestration and automation, it wants to focus on development. The operations team typically takes the lead on orchestration and automation, using a standard enterprise platform. Helix Control-M has already been helping organizations innovate faster by automating and orchestrating enterprise workflows. This functionality extends across the complete data pipeline, including orchestrating across hybrid and multi-cloud operations. The Python client makes this and other functionality accessible to data engineers and developers using the language and tools they are familiar with so they can build automated execution into their workflows as they develop them.

Understanding the BMC Helix Control-M Python client

The new Python client allows data engineers and data scientists to leverage Python programming to seamlessly interact with Helix Control-M. They can use Python to build, test, and promote data workflows through the Helix Control-M Automation API, which is a set of programmatic interfaces that let developers and DataOps engineers use Helix Control-M in a self-service manner within the agile application release process. By connecting both the data and operations teams, organizations can ensure visibility, improve service level agreements (SLAs), and deliver data-driven outcomes faster—at scale—across hybrid and multi-cloud environments.

The Python client is open-source and available under a BSD license on GitHub. Anyone can write and test their workflows using the BMC Helix Control-M sandbox instance.

Helix Control-M provides the ability to embed application and data workflow orchestration (via Python or JSON) in your DevOps tool chain. The Python client provides access to Helix Control-M's production-grade orchestration, helping to eliminate silos in your operational environment, using a programming language that is familiar and intuitive for developers and engineers. Some of the key Helix Control-M features include:

- End-to-end visibility across the entire technology landscape, including multi-cloud and on-premises environments
- Support for data-focused cloud services and traditional technologies and applications
- Predictive SLA management
- Self-service access for engineers, developers, operations, and business users
- Granular security and comprehensive governance

The Python client provides access to key workflow orchestration capabilities in Helix Control-M. Some of these features include:

- Developers can build execution rules into their workflows, by taking a Jobs-as-Code approach with REST APIs and JSON to accelerate application build, test, and validation times. This cuts costs and improves quality by finding defects and bugs earlier in the software development lifecycle.
- Simplify and scale data pipelines by ingesting and processing data from platforms like Hadoop, Spark, Amazon EMR, Snowflake, and Amazon Redshift and get a 360-degree view of data pipelines at every stage—from ingestion, to processing, to analytics. Helix Control-M can automate and orchestrate data flows from all sources including cloud, on-premises, data lakes, etc.
- Intelligently move file transfers from a central interface.
- The ability to visualize and test workflows prior to deployment. The Helix Control-M sandbox instance provide all the functionality needed for experimentation and testing.
- Automate workflow movement across development, test, and production environments.

Helix Control-M has integrations to PaaS services on AWS, Azure, and GCP, SAP (including HANA), Informatica, leading databases, and much more. BMC has committed to increasing the release velocity for cloud services and other integrations. The Python client adds support for all job types, including:

- AWS Glue
- Azure Data Factory
- Databricks for AWS and Azure
- Google Cloud Dataflow
- Google Cloud Functions

The Python client also bridges developers' favored environments and tools to the enterprise application and data workflow orchestration platform.

> "Control-M and Helix Control-M automate every aspect of the big data pipeline from a single point of control, from ingestion and data processing to presenting it to an analytics layer, removing reliance on multiple point solutions in various stages. They offer deep integration with the Hadoop ecosystem, including support for HDFS, Spark, MapReduce, DistCp, Pig, Hive, Sqoop, Tajo, Oozie, etc."

Enterprise Management Associates | 2021 Workload Automation EMA Radar Report

How Does It Work?

Here is an overview of how to set up and use the Python client in Helix Control-M.

1. Install the library one time, using pip
2. Import the functions you plan to use
3. Specify folder attributes
4. Log in to Helix Control-M
5. Create folder and jobs
6. Define dependencies
7. Display dependencies in tabular or graphical form to validate the flow
8. Submit to Helix Control-M

You can see a quick demo here (free registration required). Note that there is only one client for Control-M and Helix Control-M. The same client connects to both if needed.

Summary

The Python client for BMC Helix Control-M was created to make life easier for data science, software development, and IT operations staff. It brings automation to an area that needs it. As the Anaconda 2021 State of Data Science Report notes:

> "Despite themes in the media suggesting automation will 'take over,' it's actually welcomed by data practitioners and isn't seen as a competitor.... Automation is seen as a complement to work. Data scientists aren't worried because they recognize how many aspects of their job still require expert human judgment that technology can't replicate."

Helix Control-M provides automation and orchestration so application and data workflows can be developed and deployed at scale. The Python client removes a key source of friction between the development and operations environments at organizations that are already using Helix Control-M. The client helps organizations get more out of those platforms their data science staff and investments, and their IT operations professionals. The Python client helps extend DataOps principles and Helix Control-M functionality to data operations, giving data science specialists more time to spend on development, and helping operations teams deliver those innovations to the enterprise faster.

You can freely access Control-M Python client documentation here, see our blog for more perspective, and visit the BMC Helix Control-M home page for lots of additional resources.

About BMC

BMC works with 86% of the Forbes Global 50 and customers and partners around the world to create their future. With our history of innovation, industry-leading automation, operations, and service management solutions, combined with unmatched flexibility, we help organizations free up time and space to become an Autonomous Digital Enterprise that conquers the opportunities ahead.

BMC—Run and Reinvent **www.bmc.com**

536358

Partner Logo

bmc | Registered Partner

WHITE PAPER

How to Orchestrate Data Workflows with the BMC Helix Control-M Python Client

ABOUT [PARTNER]

Ut optatius et elit faccus molori veris dioris volore preriasim et es et labores expellupta cus et ut aut voluta temque poribus alit, quidel il iderrum rescipsus rende evendametur minciet hita et aut dolorit, simin perovitat rest dolorro rersperspe min cuptate ditate serae odi debis mollaccus enestrum latinve lignates est, cor suntor Ut optatius et elit faccus molori veris dioris volore preriasim et es et labores expellupta cus et ut aut voluta temque poribus alit, quidel il iderrum rescipsus rende evendametur minciet hita et aut dolorit, simin perovitat rest dolorro rersperspe min cuptate ditate serae odi debis mollaccus enestrum latinve lignates est, cor suntor

PARTNERWEBSITE.COM | XXX-XXX-XXXX