



# AUTOMATION: THE ROAD OUT OF THE APPLICATION RELEASE SWAMP

By Robert Reeves, Chief Architect for Application Release, BMC Software, and Jody Hunt, Lead Solution Manager for Application Operations, BMC Software

Every day, all over the world, systems administrators are stuck in conference rooms trying to figure out what went wrong with the latest deployment of an important application. After a seemingly routine and minor change, the application failed and nobody knows why or seems to be able to fix it. Someone knocks on the door, and a well-dressed senior manager walks in and wants to know when they'll fix the problem. The answer should be "We're fixing it now," but it rarely is.

That's because most organizations don't believe it is possible to automatically and correctly deploy the many, fast-changing components required of a modern, Web-based, multi-tier application. Throwing more people at the problem is not only too expensive (if you can find that many skilled people) but also can result in more errors. With applications becoming more complex, and changing more rapidly, the future holds more of those morning-after deployment meetings where you are scrambling for answers.

The path forward is to automate application deployment to reduce configuration errors and enable higher application uptime, more consistent deployments, higher compliance rates, and lower administration costs. This article focuses on application release management, which causes many of the biggest headaches in application management, and discusses how automation can help.

## Understanding Application Release Pain

Application release management is the process of moving new application releases from development through the various stages of testing through to deployment. Since these new releases may be critical to meeting business goals, they must be deployed quickly, at low cost, without disrupting other systems, while also ensuring compliance with organizational, industry, or government regulations.

The four steps of application release are as follows:

- » **Packaging** — Creating multiple configuration items that must be deployed at the same time
- » **Deployment** — Using the contents of a package to install applications and configure their operating environments
- » **Promotion** — Delivering a tested package to a higher-criticality environment, such as from development to QA or from QA to production

- » **Compliance** — Documenting that proper deployment processes were followed, and validating the deployed configurations

These seemingly simple steps are often frustrated by the complexity of modern application environments, and of modern IT organizations. Today's service-based applications may contain hundreds, even thousands, of critical configuration items for application servers, databases, Web servers, messaging middleware, and authorization services. Each component must be configured correctly to work with the current versions of every other component making up the application. Over time, configuration drift can cause problems as the mix of components changes, as bugs are found and fixed, or as new functions are added.

This complex mix of "moving pieces" is handed off from development to the operations staff, each of which has different (often manual or informal) systems for tracking and releasing changes. This is a formula for exactly the kind of people and process failures that cause 80 percent of mission-critical outages. According to a 2010 Gartner, Inc. Research report, "through 2015, 80% of mission-critical outages will be caused by people and process issues, and more than 50% of those outages are caused by change/configuration/release integration and hand-off issues."<sup>1</sup>

Here's an example taken from real customer experience. The quality assurance team of a company found and fixed a problem with an outdated middleware component but failed to update the script used to push the application out to dozens of customer environments. As a result, the next day, the IT department was besieged with emails complaining that the application didn't work. This is the sort of unexpected application failure that ambushes corporate IT departments every day, resulting in lost user productivity, extra troubleshooting and diagnostic costs, and reduced user satisfaction with IT.

The "brute force" solution is to throw more highly-paid employees or consultants at the problem. These experts try to write and maintain deployment scripts that attempt to automate application deployment and manage the complexity of the application environment. Another frequent alternative is to deploy every application release across every test and deployment environment manually. Even if a company could find the resources and afford this approach, it would collapse under the sheer complexity of the application environment, the number of changes made every week, and the amount of coordination required between the development and operation staffs.

## The Automation Solution

An automated approach to application release must address all four stages of the release process (packaging, deployment, promotion, and compliance) with easy-to-develop, easy-to-maintain workflows that facilitate the "handoffs" of deployment packages between different operational staffs, where so many balls get dropped.

The key high-level requirements are as follows:

- » **Model-driven configuration management** to reduce complexity and ensure reliable and predictable releases
- » **Parameterized application templates** to ensure consistent deployments across release environments, such as development, QA, staging, and production
- » **Role-based access control** to ensure that only employees with proper authorization can authorize and execute changes, which helps meet security and other compliance requirements
- » **Very granular and precise rollback capabilities** to undo any changes that threaten the integrity of the application environment

A data model, backed by an enterprise-ready Configuration Management Database (CMDB), is a key enabling technology that describes and tracks the various components required for each application. The configuration data model captures a snapshot of an application environment's configuration, including configuration item details and interdependencies. The model is an abstract representation of the application environment that can be used to compare environments to each other or to a "gold standard" configuration for auditing purposes. In addition, the model can also be safely edited or pushed out as a new configuration, eliminating the need for scripts entirely. This helps ensure reliable and predictable releases, especially given the many, frequently changing "moving parts" in modern applications.



Today's service-based applications may contain hundreds, even thousands, of critical configuration items for application servers, databases, Web servers, messaging middleware, and authorization services.

Change and release management should go beyond change approval. Your solution should enable you to perform task automation and have tight controls over authorizations and the production of consistent application release packages and deployment environments. It is this consistency over time (as the number of changes increases and the complexity of application environments rises) that is critical to avoiding unplanned downtime and exploding management costs. Such scalability is essential given the number of people and organizations, such as business partners and out-sourcers, now involved in application release cycles, the pace of application releases, and ever-increasing requirements for regulatory compliance. This approach also reduces the need to hire more staff to meet a sudden application rollout deadline.

An automated application release solution should also support virtual and physical infrastructures, to allow organizations to cost-effectively deploy changes across both environments, or migrate applications as their business needs change.

Automated workflows and the tools to create them easily can help provide consistent and aligned processes — even as responsibility for changes moves among multiple groups over time. Having consistent, automated processes on which to fall back is important, especially as companies reshuffle support functions to match changing organizational structures or outsource some responsibilities to outside vendors.

Support for granular configuration information allows administrators to control changes at a very low level. This is more efficient than replacing entire configuration files — and reduces the possibility of errors and associated downtime.

A data model, backed by an enterprise-ready configuration management database (CMDB), is a key enabling technology that describes and tracks the various components required for each application.



Another key requirement is the need for discovery capabilities that model the existing infrastructure environment and capture known good configurations as models for future deployments. Automated discovery drastically reduces the cost and time required to create deployment packages. This also reduces the time to discover configurations that have drifted out of compliance and must be changed to ensure a successful deployment.

Snapshots are another useful tool for tracking configuration changes over time, capturing differences among environments, and reporting on such variances

## Application Release Automation at Work

Application release automation is already at work, helping organizations reduce costs and increase business agility. For example, the following BMC customers achieved these results with application release automation solutions from BMC:

- » **An online trading company** reduced the cycle time for application release from 50 minutes to 50 seconds per month, reduced deployment defect rates, reduced incidents related to configuration changes by 85–95 percent, runs release audits automatically, and supports 25–30 percent more deployments with no additional staff.
- » **A bank** reduced the risks associated with changes from three major service level agreement (SLA) breaches or outages per month to none, and reduced its mean-time-to-repair (MTTR) incidents from 45 minutes to two minutes.
- » **A telecommunications company** reduced unplanned downtime by 60 percent and saved \$11 million in change management costs.
- » **One large bank** estimated it would need 40 to 60 Weblogic administrators to roll out a single new application. With the application release automation solution, the bank found it could get by with two — one to manage the rollouts and one as a backup.

for auditing and configuration drift management. Snapshots can help the IT organization prevent application outages by finding misconfigurations before they cause an outage, as well as greatly speed troubleshooting and remediation should an outage occur.

Out-of-the-box compliance templates based on industry standards, regulations, and controls provide more control over the infrastructure and application resources and help manage vulnerabilities and risks. They also reduce the need for the IT staff to “reinvent the wheel” by allowing them to adopt proven processes instead, thus preventing costly and time-consuming deployment errors.

Platform-transparent packaging that hides the complexity of deploying Java EE, .Net, and other applications allows lower-skilled workers to package and deploy new application versions. This reduces deployment costs and allows organizations to allocate senior staff to more strategic, higher value projects, while preserving their ability to use the application platform that best meets their needs.

A release automation solution must also support commercial application environments, no matter how complex. IBM® WebSphere® Portal, for example, combines a complete content management system with the IBM WebSphere application server. This combination results in the need to manage not only the thousand or so WebSphere configuration items, but also the content associated with Web applications, such as applications, themes, and portlets.

Finally, it is important to remember that application release is only part of the application operations challenge. IT managers must also manage projects and portfolios, application performance, and configuration compliance. A release automation solution should work seamlessly with other IT operations tools, enabling the cross-silo workflows that further reduce costs, prevent errors that can cause application downtime, and speed business responsiveness.

## Moving Ahead

A steady flow of new applications, as well as enhancements to existing applications, is the lifeblood of the modern business. Enterprises require these new capabilities to attract customers, improve employee productivity, forge new links to suppliers and other business partners, and enter (and even create) new markets.

Slow, inconsistent, and unreliable application deployments rob users, business partners, and customers of badly needed agility to respond to changing business conditions. The application downtime caused by misconfigurations saps user productivity and can even rob the organization of sales. They make it harder for the enterprise to cut costs, adapt to new business challenges, and meet strict compliance requirements.

More and more organizations are finding that they do not need to live with the status quo of failed deployments, ballooning support costs, and unexplained outages. Instead, they are using application release automation tools to reduce the pain, delay, and costs now associated with application deployment.

BMC BladeLogic Application Release Automation addresses the challenges discussed in this article. For more information, visit [www.bmc.com/products/product-listing/bmc-bladelogic-application-release-automation.html](http://www.bmc.com/products/product-listing/bmc-bladelogic-application-release-automation.html).

## END NOTES

1. Gartner, Inc. Research, *Best Practices: Executing Release Management With Change and Configuration Management*, May 10, 2010.

## ABOUT THE AUTHORS

Robert Reeves is the Chief Architect for BMC's Application Release products. He was formerly the CTO and one of the original founders of Phurnace Software (acquired by BMC). Reeves was the Chief Architect of the Phurnace engine and the company's first products. He has over ten years of experience in the software development industry, including roles at drkoop.com, NextJet, 360Commerce, CarOrder.com, and Trilogy Software. He has spent his career focused on configuration management and deployment of enterprise applications. Reeves has been awarded a provisional patent for technologies used in the Phurnace products. He has a B.A. in economics with a mathematics minor from the University of Texas.



Jody Hunt is Lead Solution Manager at BMC for Application Operations. His career has spanned diverse roles in the software industry, including control flow analysis algorithm development at Language Technology, Inc., object database technical sales at Object Design, technical marketing at DAZEL Corporation, telecom industry marketing and partner management at IONA technologies, and partner sales at Progress Software. He received a BA in computer science from the University of Texas.



## BUSINESS RUNS ON I.T. I.T. RUNS ON BMC SOFTWARE.

Business thrives when IT runs smarter, faster, and stronger. That's why the most demanding IT organizations in the world rely on BMC Software across distributed, mainframe, virtual, and cloud environments. Recognized as the leader in Business Service Management, BMC offers a comprehensive approach and unified platform that helps IT organizations cut cost, reduce risk, and drive business profit. For the four fiscal quarters ended June 30, 2010, BMC revenue was approximately \$1.92 billion.