

ebook

The Essential Guide to

Better Release Management and Deployment

Getting Safely to Production with Ease



Introduction

Every application development IT group has some sort of release management processes in place, some effective, some not. Release management begins when code needs to live somewhere beyond a local development machine. The process goes like this. First, you take existing code for a software application or program from somewhere. Then you promote the code into a test environment. Then you test the code. Finally, the code is put into production. While it sounds relatively simple, it's far from that.

Managing an application release and its distribution is actually an incredibly complex piece of the development process. It can involve multiple development environments, multiple IT staff members, many hardware destination points, clusters of servers (HTTP, load and database), and all kinds of different middleware. If there is a glitch at any point, it can send the whole deployment process off course and a total domino effect of trouble can ensue.

In this eBook, we will address what release management challenges you might be up against, how you can go about solving them, and a solution that will get your release management and deployment processes working for you.

The Danger Zone

A release may pass every test in QA, yet when you move it into production, things go wrong. Maybe a component of the release gets left behind, or the wrong version is included in the release package. In other instances, destination targets may be missed or files installed in the wrong locations. You might want to roll back to a previous build, but cannot because the development process has not been tracked in a way that would allow you to recreate the build, or you don't have an appropriate build environment. Or maybe your deployment processes just don't allow for easy redeployment.

If you are facing an IT audit and trying to get compliant, you could find yourself in a situation wondering, "Who authorized this move to production?" The auditors will need traceable records that show the history of deployments regardless of the tools the developers are using. All of these common problems can be avoided.





Things That Make You Go “Hmmm?”

When IT teams really start thinking about what is wrong with their release management and deployment processes, a lot of common questions are asked. Any of these sound familiar?

- ❖ What happens if a server crashes during deployment? Can I rollback to the previous state?
- ❖ Am I distributing the right version to the right targets?
- ❖ If I make a change to a server will it affect other applications?
- ❖ Why is it I am always searching through directories to find what I need?
- ❖ How can I get better visibility into multiple objects so I don't waste so much time?
- ❖ How can I keep track of who's approving the deployments? Doing the testing? The promoting?
- ❖ How can I manage, store, and document all the development information in a single place so I can comply with regulations and audit requirements?
- ❖ How can I easily repeat successful workflow and processes for future releases?

These are just a sample of the many questions that arise but the list goes on and on. With the right release management and deployment solution, all of these problems can be easily resolved. Below is a list of several key ingredients that you should look for in the solution, whether you want to build it yourself or look to an outside vendor. It's not just about avoiding a failure message, it's about enforcing best practices, increasing automation, identifying potential risk, quickly generating compliance reports, and managing multiple targets and multiple platforms.





What Should the Right Release Management and Deployment Solution Include?

- ✓ **Automation of All Movement Along the Path to Production**
From the minute a build is created to final deployment, the promotions/hand-offs should be automated as defined by you or your IT management. This operation should be completely configurable using simple point-and-click functions to meet the individual business requirements that you or your organization have established. You start by defining and setting up the deployment profile which identifies where the build items are sent at each stage (automatically or on-demand).
- ✓ **Total Visibility of Objects and Package Contents**
From the minute a build is created, you should have the ability to trace all information associated with a build package from a revision number to a tag or label, as it moves through the entire development lifecycle. This includes who approved it, who touched it, what happened to it, where it went next, etc. With a simple import command, you should be able to collect and store all identifying information tying the build package back to the original files. The traveling annotation enables you to see the complete history and audit trail associated with each package, also making it easy to meet compliance.
- ✓ **Point & Click Distribution and Deployment**
For companies managing multi-platform, distributed environments, you should have a solution that automatically deploys all necessary components to the appropriate target locations throughout the development lifecycle. It should gather, package, distribute and install application components at each stage of the lifecycle. You should be able to track and manage the distributed packages from a single management console. It will enable pain-free deployments.





✓ Deployment Rollback Features

You absolutely will need rollback recovery options. When it comes to application development, deployment is often the stage where disaster will strike. Rollback features allow users to simply label the last state of the application pre-deployment before moving into production. If something goes wrong, you should have the ability to quickly and easily roll back the machine to the previous state.

✓ Parallel Development and Conflict Resolution

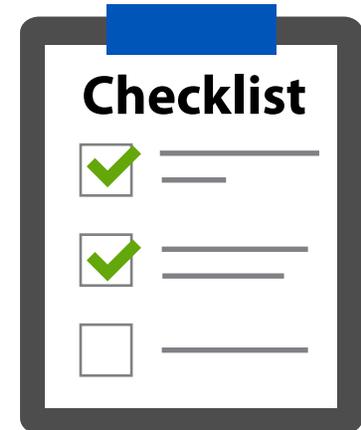
The simultaneous development of multiple versions of software applications can be complex and difficult to manage. All too often, development teams are faced with conflicts between versions, which can confuse and disrupt even the best-managed project teams. The right solution will allow you to quickly identify version conflicts using descriptive and structured status tags (active, pending, cleared, etc.) so that developers can clearly see and resolve existing conflicts at the appropriate time. Developers can take advantage of the productivity benefits that result from concurrent development without worrying about the potential chaos that results from more traditional change management approaches.

✓ Comprehensive Deployment History for Managers and Compliance Auditors

The right solution must ensure that development managers, team members, and other participants can see into the deployment process and easily review the details of a specific distribution to identify failures that require further diagnosis or repair. Development managers and auditors can now review distribution history, analyzing project productivity, failure recovery and compliance with technology change regulations.

✓ Failure Pattern Identification

Stop a problem before it starts. Make sure your solution pinpoints failure patterns among machines, such as those with similar operating systems or databases, for further investigation. You will find that you minimize the down time that developers spend waiting for procedures to execute (or for broken builds to be fixed!), because it's all automated. You will also identify "failures" faster, and be able to find and correct defects earlier.



✓ Secure and Consistent Distribution Packages

The right solution should eliminate the security concerns and inconsistencies of manual file transfers, coordinate arrival regardless of location or system type (Windows, Web, UNIX, Linux, mainframe or IBM i), log all code and content transfers for audit purposes and limit access to objects by user. That way you know you are running exactly what was tested.

✓ Workflow and Change Management

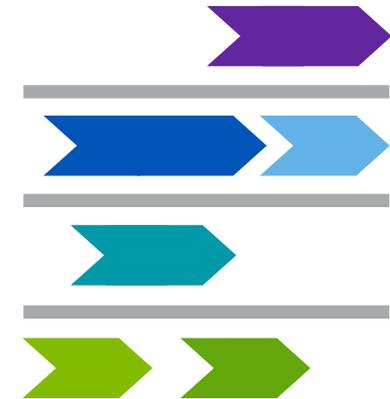
You need a solution that automates, accelerates and enforces workflow and change management processes. These functions should provide a way to verify, approve and archive change requests, requirements, and development projects. The solution should provide real-time status updates to requestors and approvers that ensure the oversight, enforceability and traceability needed for internal and external compliance.

✓ Software Configuration Management

Your solution should include functions that will automatically build, organize and maintain a central inventory of all application components. This allows you to easily access application components by business function and at various stages, across multiple environments and in conjunction with multiple project teams. Management can obtain real-time information about the status of any application, project or object across the inventory.

✓ Release Merging Capabilities

The right solution should allow for a seamless merging of releases without losing version and historical information in the merged release. The companies who will really benefit from this 'merge to parent' functionality are those with scheduled releases and tight controls, software houses who want to consolidate releases without losing the object's identifying mark (the version number), and financial institutions with strict auditing procedures.



✓ Flexibility

The solution you select should be both open and flexible in nature with integration efforts, tool options, and adoption of process methodologies. This is essential in enforcing process across release management.. It should plug in to a variety of tools so users can pursue a best-of-breed strategy. It should support numerous technical standards (e.g., Eclipse, SCCI, etc.) to give you the flexibility to choose the right tool for your IT organization.

✓ Simple Version Control Integration

Your release management processes also need to support simple version controls tools like Subversion, CVS, Microsoft Visual SourceSafe, Perforce and Microsoft Team Foundation. If you are like the majority of IT shops, your development team is using some type of version control. Most are great for keeping track of all the various versions of files; they work fast, developers love them, and some are free but they don't help project tracking or compliance. The right solution allows your software teams to keep using their own version control tools while it handles moving the build package through a pre-defined software development lifecycle (that you set up from the get-go). Through the project tracking functions, management can easily track any changes and provide the necessary compliance reports.





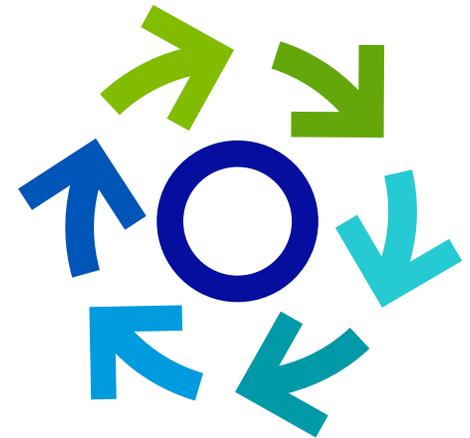
The Rocket® Aldon Release Management and Deployment Solution

Automate and streamline the entire software development process:

- ❖ Simple build integration on import from any version control application
- ❖ Logical, business-oriented view of an application's structure
- ❖ Single-click code promotion
- ❖ Separation of duties
- ❖ Automatic inventory of builds and packaging
- ❖ Role-based and/or individual permission definitions
- ❖ Control and management of all servers from one console

Package and distribute code or content files to designated target locations at each stage of the development lifecycle:

- ❖ Deployment package version tagging/labeling
- ❖ Complete history of activity logs
- ❖ Automatic tracking of objects status as they are being distributed
- ❖ Full archive options
- ❖ Easy re-deploy and rollback



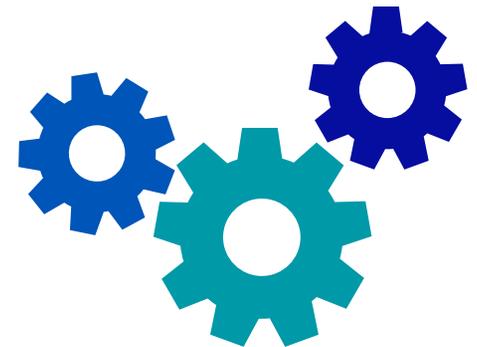
How It Works

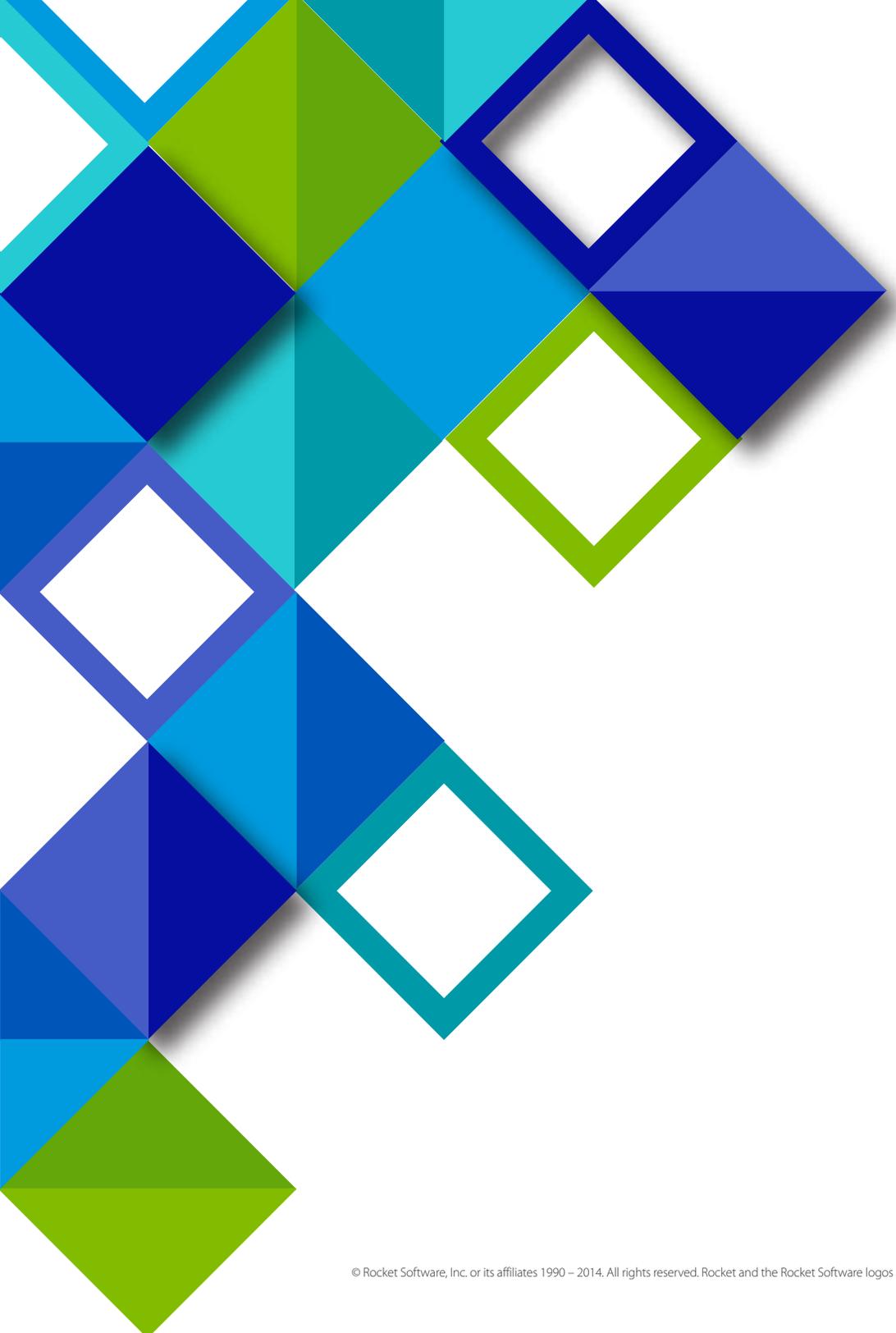
When software developers decide their build package is ready, it is automatically passed over to the Rocket® Aldon Release Management and Deployment solution. The build package then moves through a pre-defined software development lifecycle, complete with all the compliance-savvy functions you need like automated processes, workflows, notifications, and approvals. During the entire process, management can easily track changes and provide the necessary compliance reports. From the time a build is assembled and imported into the system, our solution tracks, manages, and deploys the application through automated workflows and best practices.

Rather than relying on scripts or manual procedures to create compliant move-to-production processes, the Rocket Release Management and Deployment solution provides simple point-and-click setup function for creating and maintaining approved workflows. Once the workflows are defined, the system enforces and automates them, relieving the staff of that administrative burden.

Summary

The Rocket Aldon Release Management and Deployment solution makes it easy for development and operations to coordinate efforts, reducing the costs associated with promoting new code to production. With visibility into exactly what has been changed in each package and enforcement of standard procedures, operations staff can forestall potential risks before pushing code into production.





 www.rocketsoftware.com

 info@rocketsoftware.com

