Unified Automation for Software and Configuration:

Achieving Speed, Control and Scalability in IT Operations

r Path®

Achieving Speed, Control and Scalability in IT Operations

Introduction

Software systems and their associated configuration files are inextricably related system artifacts. Software changes often result in changes to configuration settings, and changes to configuration settings often drive changes in software.

So, why are software systems and configuration settings managed separately, as if they were unrelated? Because, traditionally, tools and approaches for managing software stacks and their configuration settings have been very different animals.

This white paper sets out to describe a unified approach for managing software systems and configuration files as part of a single model-driven and version controlled approach. It will explore how rPath can automate provisioning and maintenance of software systems alongside script-based configuration tools; and it will discuss rPath's plans for native management of software systems and configuration settings as part of a single model-driven, version controlled platform.

In both cases, the result is a massive improvement in the efficiency and scalability of processes for provisioning and maintaining software systems and the business services they enable. In the end, it addresses what IT really cares about: Dramatic improvements in responsiveness and a way to deal with scale without adding cost.

The Automation Imperative

Today's software systems are complex. According to the IT Process Institute, "these days, systems have tens of thousands of files, and hundreds, or even thousands, of configuration options, and countless file versions."[1] All of these components are interdependent and constantly changing.

What's more, system scale is compounding like never before, fueled in part by the rise of virtualization and cloud computing.

As a result, provisioning and maintaining software systems, and ensuring that all system artifacts are consistently synchronized, continues to be a major challenge—particularly against the stark backdrop of eroding budgets, reduced headcount and "lean," do-more-with-less mandates.

This makes process improvement and automation an imperative—not simply using custom scripts to automate manual tasks, but using intelligent automation to make release management and change processes predictable and scalable.

---

[1] The Visible Ops Handbook: Implementing ITIL in 4 Practical and Auditable Steps. IT Process Institute, June 2005.

rPath—Intelligent System Automation

Recognized as a leader and innovator in next-generation system automation, rPath automates the packaging, provisioning and maintenance of software systems across physical, virtual and cloud-based environments. This helps IT organizations to:

- Accelerate deployment cycles from months or weeks to minutes
- Reduce costs and improve system-to-administrator ratios by 6-10X
- Eliminate the pain of change with conflict-free and fully reversible updates

There are two key aspects of rPath that make it unique.

Deep System Modeling

The system model is about creating a blueprint for how systems should look and using that as the basis for constructing and maintaining systems over time. rPath automatically analyzes and deeply models entire software stacks—from the application through the OS—and every layer in between.

The result is the industry's first deeply modeled system inventory, which describes the desired state of every file, binary, application component, and software stack on every production system, including information about policies that must be adhered to, the entire dependency chain—including OS components, middleware and libraries—and the impact of change.

These models are stored as version-controlled system manifests—they provide an unprecedented degree of system transparency, and serve as a basis for conflict-free deployments, updates and managing the complete lifecycle of deployed systems.
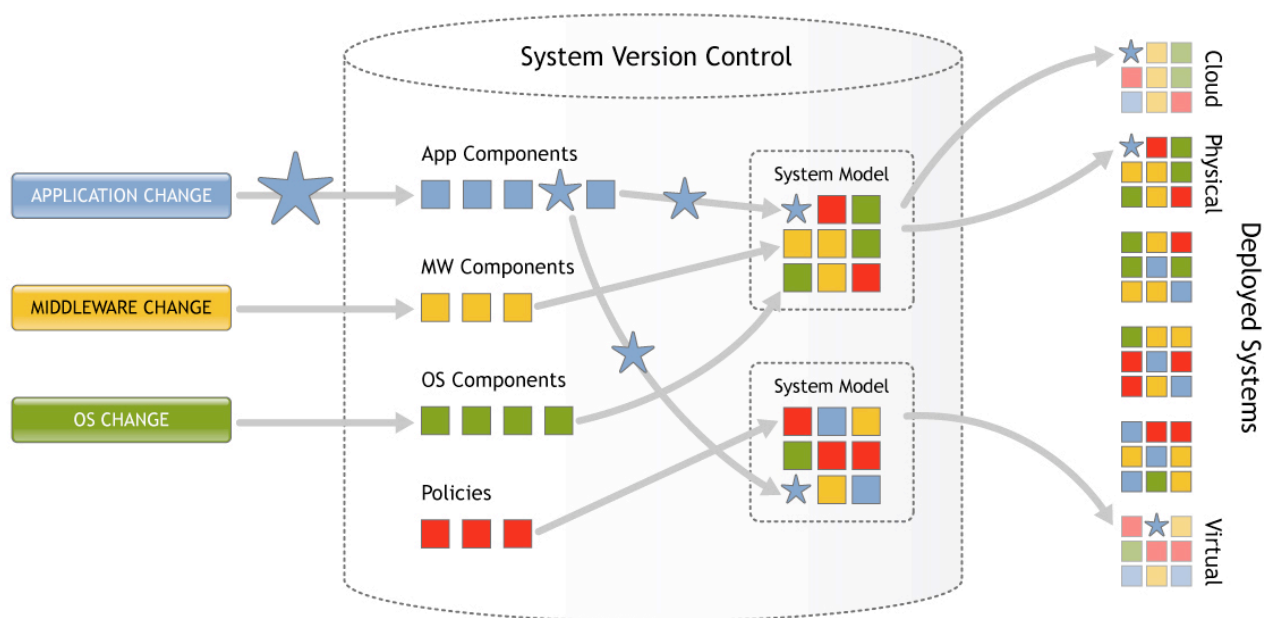
Figure 1: The version-controlled system model provides deep system transparency and serves as the basis for managing change.

System Version Control

System version control is about describing systems over time—creating a persistent and immutable system blueprint. rPath provides the industry's only commercial version control platform for deployed software systems.

rPath is not a source code management system—it's an operational management platform that applies the principles and disciplines of source code control to the management of deployable software systems—system manifests, packages, binaries, policies and configurations.

System version control is the technological foundation that allows a complex system to be defined by a single version number, ensuring systems can be quickly reproduced, patched and updated, rolled back and reported on.
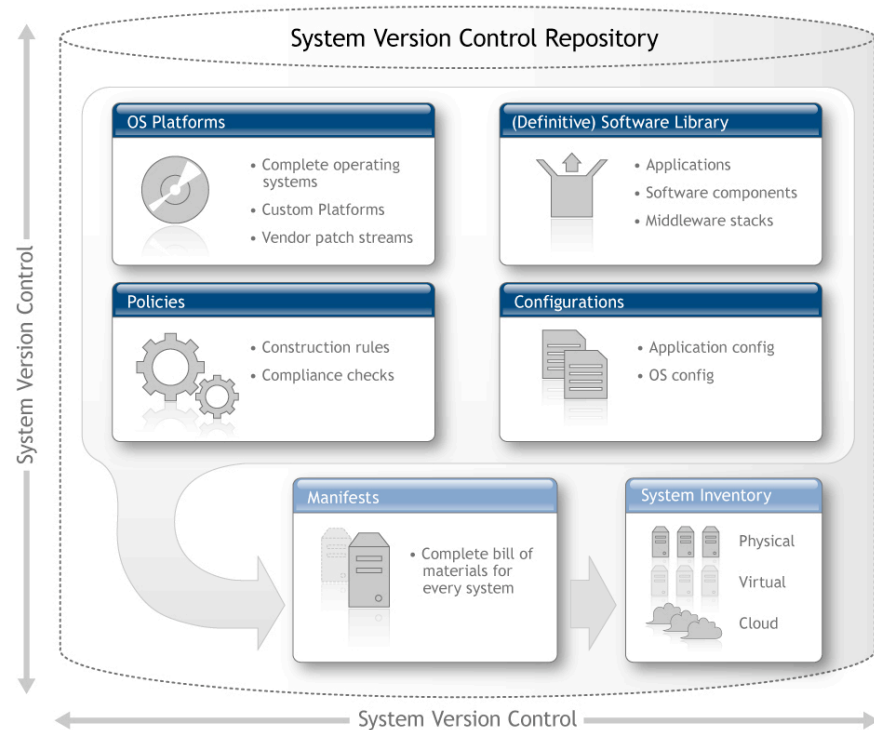
Figure 2: rPath provides a platform for modeling and versioning entire software stacks, including policies, dependencies, configurations, etc.

Applying Intelligent System Automation to Configuration Files

The combination of a system model and version control makes rPath the best solution available for automating and managing software. The same principles of deep modeling and system version control can and should be applied to the process of creating, provisioning and maintaining configuration file data—the information associated with the host environment, which needs to be created or edited to optimize the system for its local operating environment.

Today, configuration data is often edited by hand or with configuration management tools designed for automating this specific task. Some IT organizations are using open source tools like Reductive Lab's Puppet, Cfengine or Opscode's Chef for this purpose.

These tools provide value for managing configuration files, and less value for managing software stacks. For this reason, combining tools like Puppet, Cfengine and Chef with rPath can form a powerful solution for managing entire systems, including their supporting configurations.

Today, rPath supports script-based tools like Puppet, Cfengine or Chef in two ways.

Using rPath and configuration tool side-by-side

In this scenario, Puppet, Cfengine or Chef can be used to manage configuration files, while rPath is added to manage the operating system, middleware and application software. No changes or integrations are required in either system.

Some of today's customers who use this approach add a lightweight integration such as using rPath to install the Puppet software itself, or using Puppet's change scheduling features to initiate rPath software updates.

Using rPath to deploy and manage configuration tool scripts

Since these tools rely on programming custom scripts that necessarily change over time, the scripts themselves need to be managed under version control. rPath provides the perfect foundation for managing configuration scripts alongside system manifests. Managing and versioning these scripts with rPath means that software systems and their associated scripts march through the lifecycle together—testing, deployment, rollbacks, auditing are controlled and coordinated.
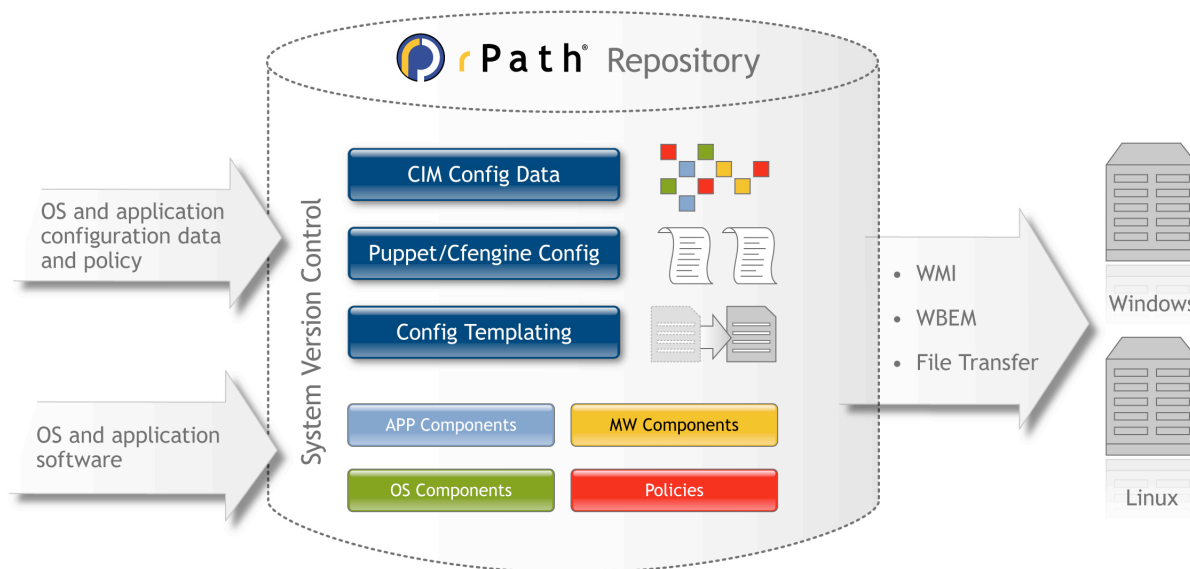

Figure 3: rPath provides a unified automation platform for managing configuration scripts alongside other system artifacts.

Using rPath with Puppet, Cfengine, Chef and other configuration tools enables IT
organizations to:

- **Reduce costs and improve system-to-administrator ratios** by
  automating provisioning and maintenance of complete software stacks and
  associated configuration data across physical, virtual or cloud-based
  environments;
- **Mitigate compliance and operational risks** by eliminating the pain
  of change and adding transparency, consistency and control to system
  provisioning and change processes;
- **Accelerate IT** by automating manual and ad hoc processes for improved
  business responsiveness and to support elastic computing models.

Strategic Direction

Today, rPath automates the creation, provisioning and maintenance of software
systems across physical, virtual and cloud environments. It does this based on a
unique approach to model-driven and version-controlled automation.

rPath's 2010 roadmap invests heavily in model-driven and version-controlled
configuration management—extending its core capabilities to native configuration
management.

Since configuration requirements are highly diverse across enterprises, rPath is
investing in and supporting two complementary approaches to configuration:

- **Version Controlled CIM (Common Information Model)** — rPath will
  be the first vendor to bring full CIM-based configuration management under
  system version control. CIM is an expanding industry standard for system
  configuration modeling. By storing CIM data under the same version control
  umbrella as software, rPath will make it easy to deploy and update
  consistent systems by simultaneously deploying applications and their
  supporting configurations, and will have the first complete system model.
- **Direct Configuration Templating** — Another future approach is to
  templatize and parameterize traditional configuration files under system
  version control. This approach does not provide the powerful central model
  of CIM, but it is easier to adopt for applications with complex configuration
  files that require small degrees of change in deployment. A good way to
  think of this is as a lighter-weight and ad hoc "backdoor" to fully model-
  driven configuration, which allows quick changes to be easily made on the
  fly.

Conclusion

Why shouldn't software systems and configurations be managed together? Today, they're not because of a limitation in today's tools, not a limitation of imagination or need. With rPath, configuration data and software systems can be managed together as part of a unified automation platform.

Today, this means managing your configuration scripts alongside system manifests managed with rPath—and benefiting from better control and coordination between these artifacts. Tomorrow, it will mean a complete, model-driven platform for automation of systems—software and configuration settings.

This also means complete control and transparency around the system lifecycle. Configuration and software are mutually coordinated and controlled: They're tested together, deployed together, changed together, and rolled back together. They can even be audited and remediated together; out of band configuration changes can be automatically detected and corrected. "Configuration drift" is eliminated.

The result: Dramatic improvements in speed, control and scalability for IT operations.

About rPath

rPath automates system provisioning and maintenance across physical, virtual and cloud environments. rPath's innovative release automation platform is based on the industry's only commercial version control repository for managing deployed software systems. The result is an easy-to-deploy and cost-effective automation solution for rapid, low-risk and low-overhead deployment and maintenance of complex software systems. rPath dramatically improves responsiveness to business lines, reduces compliance risks, and allows resource-constrained IT organizations to significantly reduce operating costs and "do more with less." Headquartered in Raleigh, NC, rPath has 80+ customers including some of the world's largest enterprises and ISVs. Visit www.rpath.com.