

IBM proactively helping insurance companies to comply with the NAIC Model Audit Rule

Highlights

- **NAIC revises Model Audit Rule to improve surveillance of the financial condition of insurers**
- **Affects insurers with USD \$500M or more in annual direct and assumed premiums**
- **First management report of internal control over financial reporting due in 2011 for the 2010 reporting period**
- **IBM Rational Build Forge provides a solution to help insurers support the Model Audit Rule, Sarbanes-Oxley and other compliance requirements**

What is the Model Audit Rule?

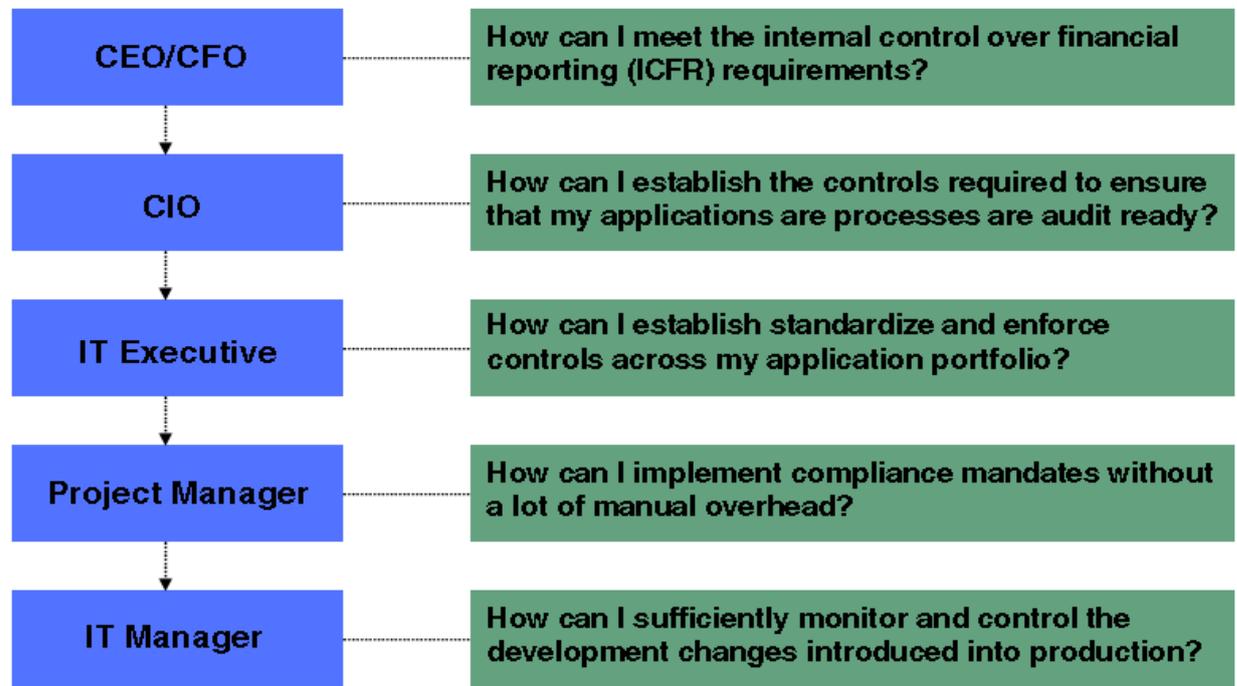
The purpose of the Annual Financial Reporting Model Regulation, commonly referred to as the Model Audit Rule, is to improve surveillance of the financial condition of insurers.

The National Association of Insurance Commissioners (NAIC) revised the regulation to incorporate certain best practices related to auditor independence, corporate governance, internal control over financial reporting. Once adopted by a State, a Management's Report of Internal Control over Financial Reporting (ICFR), similar to the report required by Section 404 of Sarbanes-Oxley (SOX), will be due in 2011 for the 2010 reporting period. According to the NAIC's Implementation guide for the Annual Financial Reporting Model Regulation, the areas of potential material internal control risk include: investments (including capital expenditures), policy and claim reserves, benefit payments, premiums and agent's balances, reinsurance, related party (affiliate) transactions and operating expenses/taxes.

SOX Section 404 requires that "Changes to business processes, including computer systems, are identified and the documentation is updated promptly. Changes to key controls are identified and assessed promptly".

An insurer that is a Sarbanes-Oxley compliant entity, or a direct or indirect wholly-owned subsidiary of a Sarbanes-Oxley compliant entity, will be able to file its or its parent's Sarbanes-Oxley Section 404 report in satisfaction of the Model Audit Rule ICFR reporting requirement. However, there is also an addendum that needs to be filed with the Sarbanes-Oxley Section 404 report that would include a positive statement that there were no material processes related to the preparation of the audited statutory financial statements that were excluded from the Section 404 report.

How does a CEO/CFO problem become a software development problem?



What types of insurance companies does this affect?

Any public or non-public company, including insurance companies, captive insurance companies, nonprofit insurers or health plans that file an annual statement with their domiciliary State regulator is affected. Insurers with USD \$500M or more in direct and assumed premium annually are required to prepare and sign an ICFR.

Meeting the compliance requirement

It's necessary to have controls in place that sufficiently monitor changes introduced into production. Most complex development environments involve a patchwork of technologies (such as version control, test automation, defect tracking and

deployment) in addition to proprietary tools. While the best of these products may have compliance information held within its database, it can be difficult to extract this information with any regularity, and even harder to reconcile the data with other products to form a true picture of what was delivered into production. Most organizations have a source repository "place of record" where different versions of software and baselines are stored. *But where is the place of record to reproduce an entire release?* A solution to this, is to automate the software build and release process.

The build and release process involves a comprehensive progression that connects each stage of the software assembly process, including linkage to source control, test automation, defect tracking, packaging and deployment systems to produce the final executable that is delivered to internal and external customers. This anchors compliance data to the definitive artifact of any software project, the executable. It also provides a reliable record of what really happened in development and what was delivered in production. When this process is automated, it becomes repeatable, reliable and standardized.

IBM Rational Build Forge:

- Consolidates build-related information from all your tools in the software delivery process, including source code managers, defect tracking and test systems
- Automates the build management and software delivery processes
- Improves product quality, speeds development cycles and increases staff productivity
- Supports geographically distributed development teams
- Provides an adaptive solution to help development teams standardize and automate repetitive tasks, manage compliance mandates and share information

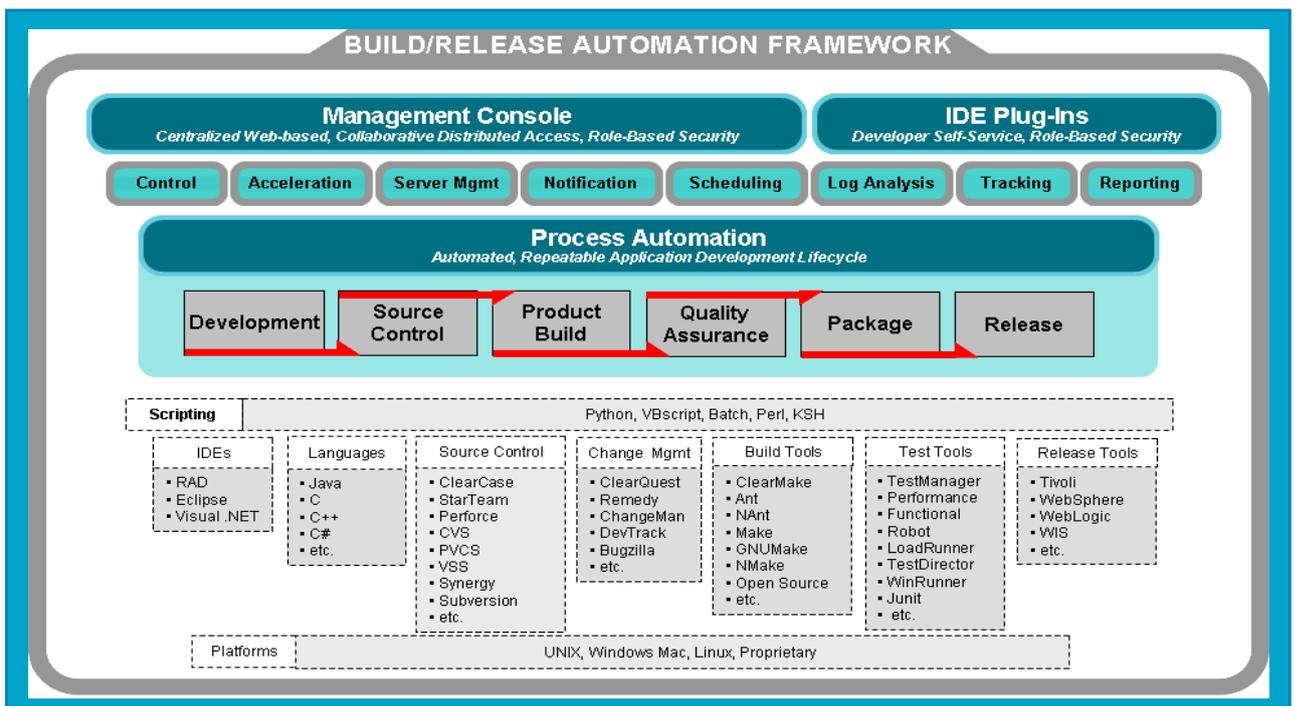
What is Rational Build Forge?

IBM Rational Build Forge® is a build and release automation solution that bridges the gap between coding and deployment by allowing teams to create standardized software build environments, and automate the steps in the process of delivering software releases from source code management systems through test or production deployment. **IBM Rational Build Forge provides a complete history and audit trail for every build, test, and deploy activity. This powerful and critical feature supports the Model Audit Rule, Sarbanes-Oxley and other compliance requirements.**

In essence, Rational Build Forge provides a solution to automate the entire process end-to-end; not just automating individual tasks, but also the hand-offs among the various steps in the process. Rational Build Forge was designed to integrate existing scripts and tools, so there is no need to

replace those pre-existing assets. Rational Build Forge offers a comprehensive application development process management solution that provides complete management and control of the application development lifecycle. Rational Build Forge automates, standardizes and optimizes complex processes, integrating diverse toolsets to deliver a repeatable and reliable application development lifecycle process.

Automation is really just the first piece of what Rational Build Forge does; it also provides value-added capabilities such as build acceleration and automatic notifications that are accessed through a secure, web-based interface that allows managed delegation of tasks to other groups - even offshore teams. The result – faster development cycles, higher product quality, increased staff productivity, reduced costs and shorter time-to-market.



Five “best” practices for development compliance management

Best practices for development compliance management:	Rational Build Forge can help meet this by:
<p>Creating automated, cross-functional development processes. Standardize, retain and communicate development processes across the application development lifecycle.</p>	<p>Rational Build Forge software streamlines complex processes by allowing tasks to be grouped into a container called a “project.” Projects can involve a comprehensive, multiphase development process, extending from source checkout all the way through the build, test and deployment phases.</p> <p>Each development process is documented, retained and executed consistently each time, whether the process occurs within a single location or across the globe.</p>
<p>Enabling the computer to document your processes. Store and track critical process data automatically in a secure knowledge base rather than relying on manual data collection.</p>	<p>Rational Build Forge captures critical data and provides traceability to use as evidence for compliance management and auditing tasks by:</p> <ul style="list-style-type: none"> • Automatically tracking your processes from initial coding through deployment, and recording the results in the system's database. • Versioning your processes for each product iteration so you can tell what has changed, who changed it and why it was changed. • Consistently documenting your processes by generating a complete Bill of Materials with each completed process run of the release.
<p>Prioritizing reproducibility of executables. Capture comprehensive configuration information to allow teams to re-create accurate reproductions of any build, executable or release from the past or present.</p>	<p>Almost every executable is the direct result of more than just compiling code. Important information about the environment, operating system, patch level, Java™ Developers Kit, etc., is required to completely and accurately reproduce the executable.</p> <p>Within Rational Build Forge, an executable is created by running a set of processes. Whenever a build or other process run occurs, the Rational Build Forge system tags it with a 128-bit fingerprint or digital signature that can be used to trace information throughout the development and implementation process back to a unique process run or executable.</p>
<p>Establishing ongoing governance through consistent adherence and enforcement. Institute IT controls that manage access for different roles, and perform constant checks and balances on positive and negative compliance indicators.</p>	<p>Rational Build Forge provides audit information about every command, in every environment, on every server.</p> <p>With Rational Build Forge, data collection for compliance management purposes is a systematic, unattended and sustainable process that continually requires team members to follow prescribed corporate practices.</p> <p>Compliance evidence can be produced by the Rational Build Forge system in four ways.</p> <ul style="list-style-type: none"> • Generation of detailed audit trails and documentation of who did what, when and why. Rational Build Forge automatically logs every action it takes for any project, storing all logs in its database. • Segregation of duties with customizable security settings and privileges. • Communication of real-time status and potential compliance risks through notifications and alerts. • Reduction of audit preparation time by leveraging a consolidated source of compliance data.
<p>Continuously monitoring and reviewing. Extract management information and provide ongoing review to identify potential compliance risks and assess organizational rewards.</p>	<p>Rational Build Forge software continuously monitors the development and implementation processes and can pinpoint policy violations and identify areas requiring additional process controls through detailed team member activity reports and quality trend reports that show improvements and gaps over time.</p> <p>Out-of-the-box reports can help reduce the time spent on compliance reporting, and ad hoc reporting capabilities allow teams to customize reports to meet their specific organizational needs.</p>



Summary

NAIC revised the Model Audit Rule, also known as the Annual Financial Reporting Model Regulation, to incorporate certain best practices related to author independence, corporate governance and internal control over financial reporting. The purpose of this regulation is to improve surveillance of the financial condition of insurers. Any public or non-public company, including insurance companies, captive insurance companies, non-profit insurers or health plan that files an annual statement with its domiciliary State regulator is affected. Non-US based insurance companies doing business in the United States are included, provided they file an annual statement.

Most states have already adopted the Model Audit Rule by incorporating the provisions into their regulations. The first Management's Report of Internal Control over Financial Reporting will be due in 2011, for the 2010 reporting period; therefore, it's imperative that companies put provisions in place now.

Rational Build Forge provides a solution to automate the ENTIRE process end-to-end -- not just automating individual tasks, but also the hand-offs among the various steps in the process. Rational Build Forge offers a comprehensive application development process management solution that provides complete management and control of the application development lifecycle.

Rational Build Forge automates, standardizes and optimizes complex processes, integrating diverse toolsets to deliver a repeatable and reliable application development lifecycle process. Rational Build Forge:

- *Provides automated, cross-functional development processes.*
- *Enables the computer to document the processes.*
- *Captures comprehensive configuration information to allow teams to re-create accurate reproductions of any build, executable or release from the past or present.*
- *Establishes ongoing governance through consistent adherence and enforcement.*
- *Provides for continuous monitoring and reviewing.*

...and thus help an insurer meet the requirements for the Model Audit Rule provision of Internal Control over Financial Reporting (ICFR).

© Copyright IBM Corporation 2010

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
January 2010
All Rights Reserved

IBM, the IBM logo, ibm.com, and Rational are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.