

Eliminate Build Failures and Slowdowns

Build systems are often a critical point of failure in software development organizations. A slow or error-prone build can cause costly delays in bringing software applications to market. This is especially true in today's development environments in which a reliable, efficient build process is absolutely necessary to ensure the fast delivery of high-integrity software products.

Until now, teams have lacked the visibility into their build systems that can allow them to determine with certainty the quality and security of assembled software components and the exact reason a build failed.

Coverity's build analysis capability, an essential part of the Coverity Integrity Center, allows development teams to ensure that every component of their software meets business requirements and integrity standards. It identifies problems that lead to delays in software delivery by providing deep insight into the software build process so you can pinpoint the exact cause of slow or failed builds.

Benefits of Build Analysis

- Unlock developer productivity
- Effectively manage change
- Ensure security and compliance
- Improve Software Integrity

Productivity

With this essential capability, development teams can now begin to accurately assess the overall integrity of the build and answer key questions—what components were assembled? Were they the right ones? Were all the right files checked in? What processes were run and how often? Additionally, build analysis can be used to set rules to ensure proper compilation flags and enforce inclusion or exclusion of directories and libraries that should not be in the build.

Change

Build analysis helps engineers control the effects of change by allowing them to create a map of all processes that are executed during the build. For each process, this map contains a complete set of all the input files to that process and all the output files that are generated by that process. Teams can now have a dependency tree for the entire software production process with file and process granularity.

Compliance

With the ability to generate a comprehensive bill of materials, build analysis provides critical compliance and reporting data by showing that the files that have been checked out are actually the files contained in a specific build, and that the list generated by the source code management (SCM) system is accurate for a specific build.

Software Integrity

Development teams can now accurately determine if an application is ready to move to the deployment phase in the development cycle or whether additional steps need to be taken to improve software integrity. With Coverity's build analysis, faster build times for your software can mean faster time to market of your products.

Improve Software Integrity with Coverity Build Analysis

- Eliminate defects due to the improper or accidental use of wrong object files
- Find file access violations and leaked files
- Uncover unnecessary duplicate process invocations
- Pinpoint and eliminate the root-cause of slow builds
- Map all the processes executed during a build
- Gain comprehensive visibility into dependencies
- Verify that changes are actually being delivered
- Generate comprehensive 'bills of materials' to confirm proper component use
- Verify and validate the use of internal and open source code
- Produce essential data for meeting compliance requirements
- Align packages with components (including third party and open source)
- Speed production and reduce unnecessary retesting of components

Supported Platforms

Build Server: Linux Redhat Enterprise 5, IA, Solaris 8,9 and 10

Desktop client: Linux Redhat Enterprise 5, Windows XP, 2003