**S101 Summary**

**KEY BENEFITS**

- Limit and control the complexity of code bases
- Plan a move away from spaghetti code
- Impose your own code model
- Report measures representing tangled, interconnected code and overly complex structures
- Create and edit architectural diagrams that represent how your code should be structured
- Observe architectural violations through a layering diagram
- Create architectural rules and constraints
- Monitor structural changes from baseline to current code versions

**KEY FACTS**

- Structural analysis by module composition or slice through the structure
- Detailed collaboration mode for dependency tracing
- Dependency display as graph or dependency structure matrix (DSM)
- Charting and analysis of structure metrics
- Architectural diagrams generated for portions of, or entire, code trees
- Architectural rule constraints and concessions
- Comparison of baseline code structure with newer versions

---

**Structure 101**

for QA·C and QA·C++

Using PRQA’s advanced language parsing engines for C and C++ to extract and deliver high-fidelity data to Structure 101, PRQA is helping developers to better understand and manage C and C++ software architecture. Its partner in this product collaboration is Headway Software, who pioneered large-scale code structure visualization and architectural constraint mechanisms in the Java domain.

Structure 101 examines the structure of large code projects, and exposes interdependence and associated complexity between modules. It does this by ascertaining and graphically displaying dependencies right through the project codebase using sophisticated algorithms to detect and display the structural layout. QA·C and QA·C++ are essential in this examination by providing the most granular and highest fidelity structural and dependency output. The combined solution gives developers and architects key warnings and advice about excessively tangled code, bypassing of defined APIs, and source projects that do not meet modular design requirements.

---

**S101 Key Elements**

**Analyse code structure** – view the dependencies across your project from top-level modules down to individual functions, classes and files; inspect each dependency and view its source; display structural data in a series of graphs and dependency matrices.

**Perform impact analysis** – posit structural changes without touching the code; examine inappropriate library usage; test for a clean API before switching to a newer version of a module.

**Descramble unstructured code** – impose your own model for code structure using sophisticated auto-partitioning options or user-guided transformations.

**Limit structural complexity** – explore measures representing tangled, interconnected code (Tangles) and overly complex structures (Fat).

**Manage code architecture** – create and edit architectural diagrams that represent how your code should be structured; observe architectural violations through a layering diagram; create architectural rules and constraints; publish violations to these constraints and rules; monitor structural changes from baseline to current code versions.

---

www.programmingresearch.com
About PRQA

PRQA, founded in 1986, is the world’s leading developer of software source code analysis technologies. With over 3,000 customers worldwide, PRQA is a global provider of solutions that enforce coding standards to ensure software quality.

PRQA has representation on and close affiliation with the following standard bodies:
- ISO C Committee
- ISO C++ Voting Member
- MISRA C Founding Member
- MISRA C++ Founding Member

Provides products and services to industries including:
- Aerospace
- Automotive
- Defense
- Chip Manufacturing
- Finance
- Industrial Applications
- Medical Device
- Telecommunications

Contact Us

For further information regarding PRQA products and consulting services, please contact PRQA via your local sales representative, or directly at: info@programmingresearch.com

© Programming Research Ltd 2012