Reviewing Static Code Analysis Results
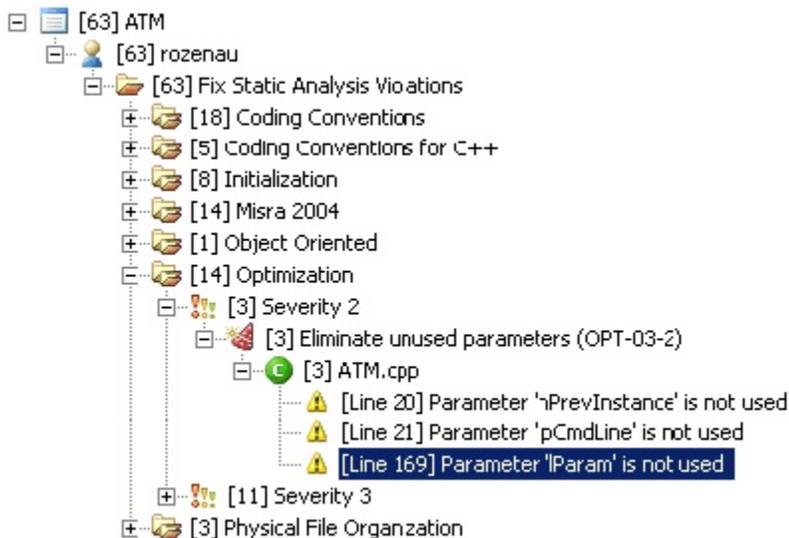
This topic covers how to analyze and correct static code analysis violations. Sections include:

- Accessing Results
- Responding to Results
- Opening Test Configurations that Trigger Violations
- Learning More About Violated Rules
- Reviewing and Correcting the Responsible Source Code
- Suppressing the Reporting of "Allowed" Violations

Accessing Results

For tests run in the GUI, results are reported in the **Fix Static Analysis Violations** category of the Quality Tasks view.

For tests run from the command line interface, results are reported in the **Static Analysis** section of the report. If results were sent to Team Server, results can be imported into the GUI as described in Importing Results into the UI. They will then be available in the Quality Tasks view.

```
[63] ATM
  [63] rozenau
    [63] Fix Static Analysis Violations
      [18] Coding Conventions
      [5] Coding Conventions for C++
      [8] Initialization
      [14] Misra 2004
      [1] Object Oriented
      [14] Optimization
        [3] Severity 2
          [3] Eliminate unused parameters (OPT-03-2)
            [3] ATM.cpp
              [Line 20] Parameter 'hPrevInstance' is not used
              [Line 21] Parameter 'pCmdLine' is not used
              [Line 169] Parameter 'lParam' is not used
        [11] Severity 3
      [3] Physical File Organzation
```

Responding to Results

For each violation reported, we recommend that you and your team review the rule description and the related code, then decide whether:

- The violation is valid and significant (and the violation should be corrected)
- The rule does not apply in that particular context (and the violation should be suppressed)
- The rule is not well-suited to your projects or priorities (and the related rule should be disabled)

Many teams like to review C++test's static analysis violations during code reviews. Developers check their code using the rules selected by the team's architect and/or manager. If a developer thinks that it makes sense to ignore a particular rule violation, that developer discusses this at the code review. The team then decides whether the violation should be suppressed, the rule should be disabled, or the violation should be corrected.

Opening Test Configurations that Trigger Violations

Test configurations that trigger violations can be opened from the Quality Tasks view: Right-click on a violation and choose **View Test Configuration.**
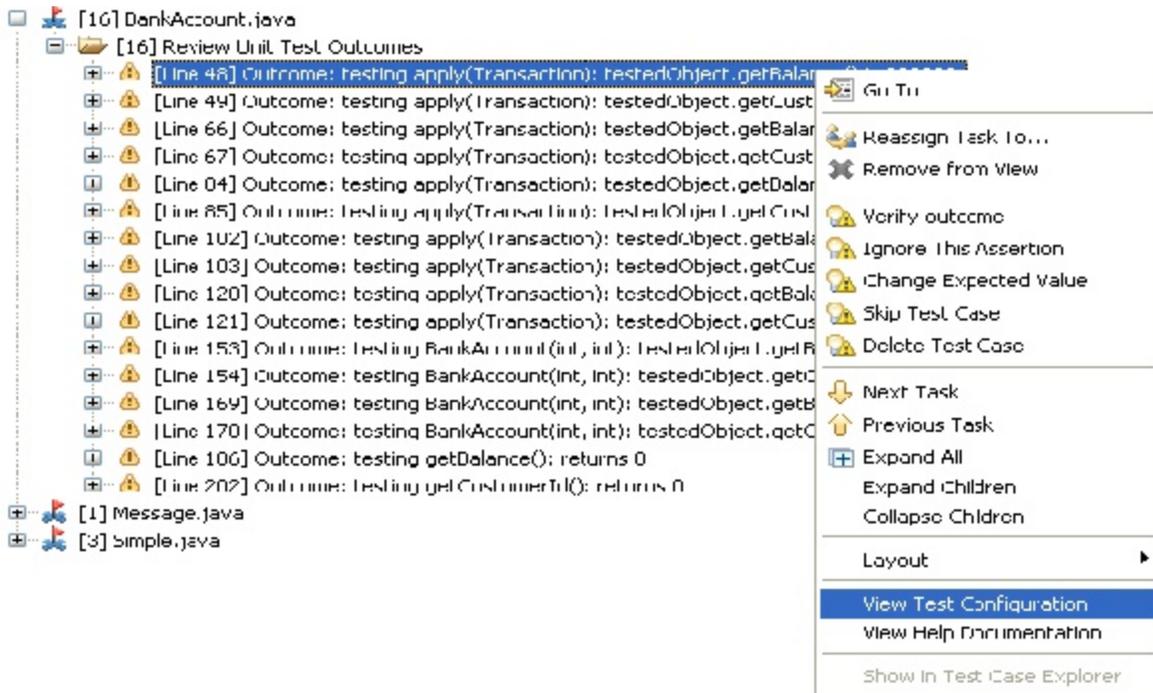
Quickly accessing test configuration from the violation is useful for group architects who are customizing tests and want to quickly disable rules that aren't applicable. Developers importing results from a server-based run may also need to open and review test configurations that trigger violations.

Learning More About Violated Rules

The C++test rule descriptions can help you determine which rules your team wants to follow, understand how reported violations can impact application reliability, security, maintainability, etc., and learn how to correct reported violations.

To view a rule description file, right-click the static analysis violation message in the Quality Tasks view, then choose **View Documentation** from the shortcut menu. A yellow "Yield" sign marks the node that you should right-click.

Reviewing and Correcting the Responsible Source Code

To view the source code responsible for the rule violation, double-click the node that shows the line number, or right-click that node and choose **Go to** from the shortcut menu. The editor will then open and highlight the designated line of code.

You can make the necessary modifications, then save the modified file

Suppressing the Reporting of "Allowed" Violations

See Suppressing the Reporting of Acceptable Violations.