

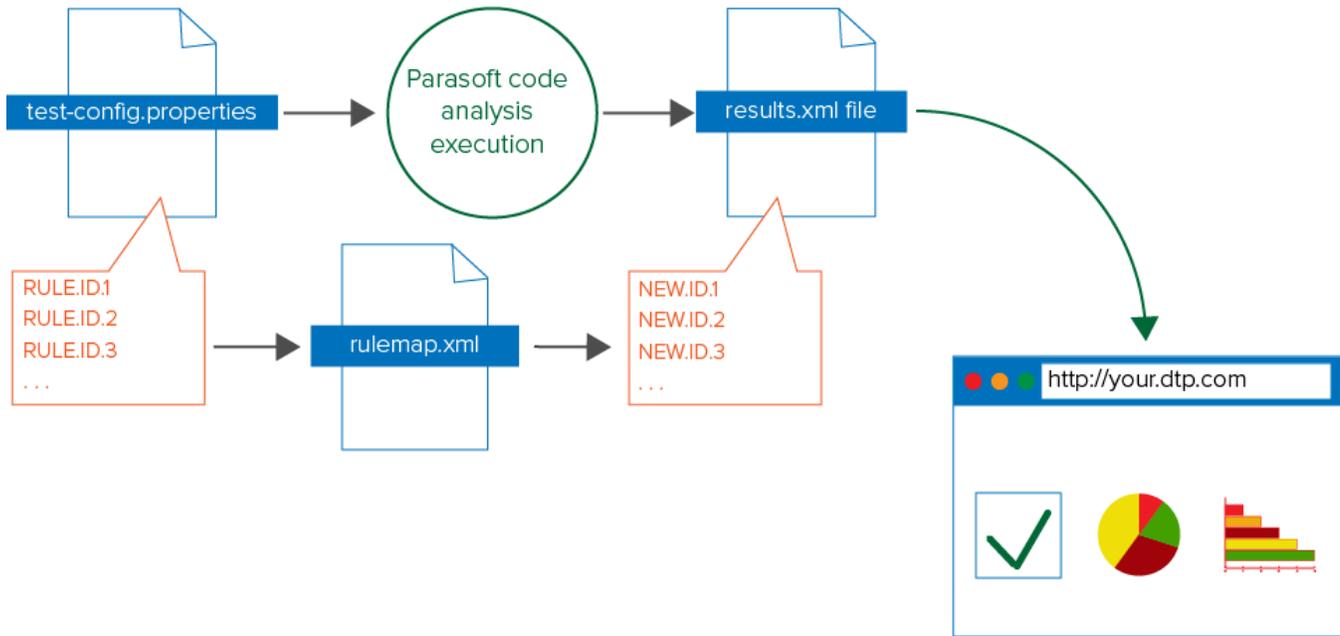
Custom Code Analysis Overview

In this section:

- [Introduction](#)
- [Rules](#)
- [Test Configurations](#)
- [Rule Maps](#)

Introduction

Parasoft's out-of-the-box code analysis tools check for programming patterns known to result in software defects. Code checkers are based on industry guidelines, academic research, and other sources to help you conform to coding best practices and meet compliance objectives. You can customize how Parasoft tools analyze code and report violations, though, to meet the unique challenges associated with your development processes. Custom rules and reporting involve the following components. The following diagram is a simplified visualization showing the relationships between test configurations and rule maps associated with code analysis execution in Parasoft.



Rules

Rule files (also called "checkers") encapsulate the logic for analyzing the code. Rules have a .rule file extension and represent the fundamental unit of a code analysis run. Parasoft tools ship with hundreds of rules, but you can create custom rules using Parasoft RuleWizard. Refer to the [RuleWizard documentation](#) for details about creating custom rules.

Test Configurations

A test configuration specifies which code analysis rules are active during code analysis execution. Parasoft C/C++, dotTEST, Jtest, and SOAtest ship with test configurations that you can modify using the editor included with the tool. Refer to the documentation for your tool for details.

You can also import test configurations for 10.x series of Parasoft tools into DTP and customize them using the dedicated UI. This enables you to centralize and manage code analysis across teams. See the [Test Configurations](#) section for details.

Parasoft compliance packs ship with special test configurations for analyzing code against industry standards. Compliance pack test configurations are automatically imported into DTP when the pack is installed. See the following documentation:

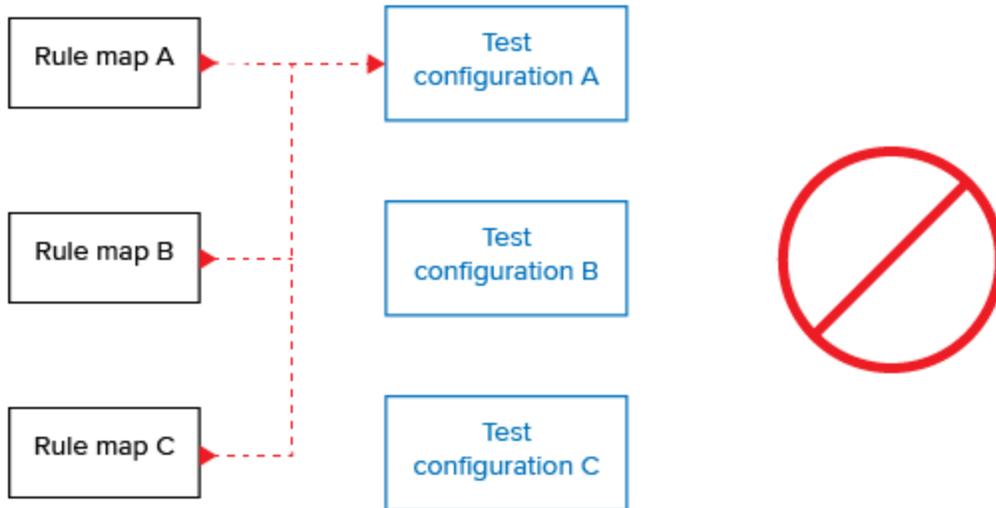
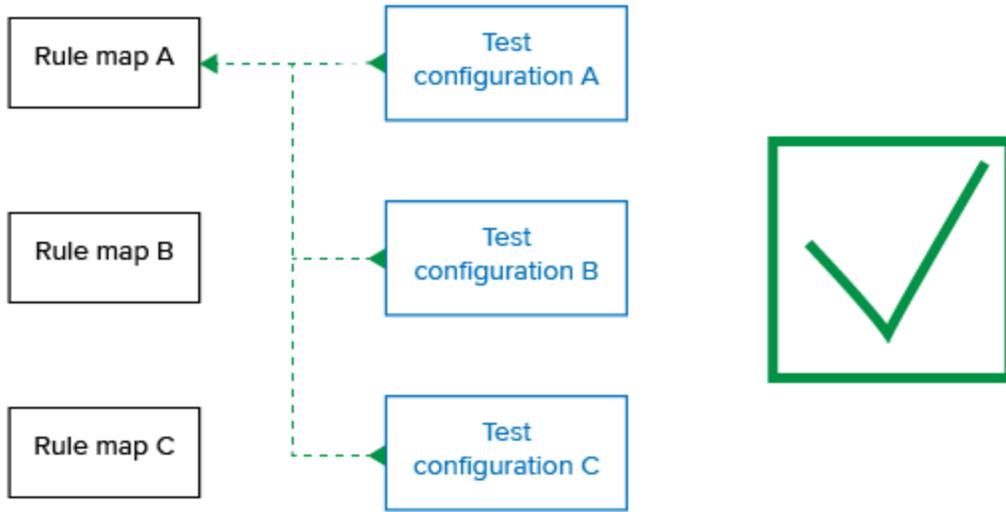
- [Automotive Compliance Pack for DTP 5.4.2](#)
- [Security Compliance Pack for DTP 5.4.2](#)

Compliance packs require an additional license.

Rule Maps

Rule maps contain the metadata about rules used in the test configuration. The rule map specifies the severity, rule ID, and categories for each rule. You can edit the rule map to reuse Parasoft code analysis rules to meet your needs.

For a rule map to take effect, it has to be associated with a test configuration. Several test configurations can use a single rule map, but a rule map can only be associated one test configuration.



Parasoft C/C++, dotTEST, Jtest, and SOAtest ship with rule maps that you can modify using the editor included with the tool. Refer to the documentation for your tool for details.

You can also use the rule map editor built into DTP to create and modify rule maps and associate them with test configurations. See the [Rule Maps](#) section for details.