

Static Tab Settings - Defining How Static Analysis is Performed

Performed

During a test, C++test will perform static analysis based on the parameters defined in the Test Configuration used for that test.

The main Static tab has the following settings:

- **Enable Static Analysis:** Determines whether C++test performs static analysis, which involves checking whether the selected resources follow the rules that are enabled for this Test Configuration.
- **Limit maximum number of tasks reported per rule to:** Determines whether C++test limits the number of violations (tasks) reported for each rule, and—if so—the maximum number of violations per rule that should be reported during a single test. For instance, if you want to see no more than five violations of each rule, set this parameter to 5. The default setting is 1,000.
- **Do not apply suppressions:** Determines whether C++test applies the specified suppressions. If suppressions are not applied, C++test will report all violations found.
- **Code parsing problems:** Determines how C++test should analyze files with parse errors. The following options are available:
 - **Ignore:** This option is enabled by default. C++test will ignore code parsing problems and report Static Analysis findings. Note that in some cases, analyzing files with parse errors may produce inaccurate results.
 - **Warning:** C++test will report Static Analysis findings, and code parsing problems will be reported as set up problems and warnings.
 - **Error:** C++test will not report Static Analysis findings if code parsing problems occur. Set up problems and warnings will be reported.

Rules Tree tab

- Determines which rules are checked during static analysis. Use the rules tree and related con-trols to indicate which rules and rule categories you want checked during static analysis.
 - To view a description of a rule, right-click the node that represents that rule, then choose **View Rule Documentation** from the shortcut menu.
 - To enable or disable all rules in a specific rule category or certain types of rules within a specific rule category, right-click the category node, then choose **Enable Rules>[desired option]** or **Disable Rules> [desired option]**.
 - To search for a rule, use the **Filter** field at the top of the rules tree. By default, this search covers only rule names. If you want to search throughout rule descriptions, click the **Search also in rule descriptions** button to the right of this filter.
 - To hide the rules that are not enabled, click the **Hide Disabled** button. If you later want all rules displayed, click **Show All**.

Tips

- The number next to each rule ID indicates the rule's severity level. The severity level indicates the chance that a violation of the rule will cause a serious construction defect (a coding construct that causes application problems such as slow performance, security vulnerabilities, and so on). Possible severity levels (listed from most severe to least severe) are:
 - Highest - Level 1
 - High - Level 2
 - Medium - Level 3
 - Low - Level 4
 - Lowest - Level 5
- To learn about the rules that are included with C++test, choose **Help> Help Contents**, then open the C++test **Static Analysis Rules** book, then browse the available rule description files.
- To generate a printable list of all rules that a given Test Configuration is configured to check:
 1. Open the Test Configurations panel by choosing **Parasoft> Test Configurations**.
 2. Select the Test Configurations category that represents the user-defined Test Configuration you want to modify.
 3. Open the **Static** tab.
 4. Click the **Printable Docs** button.

Flow Analysis Advanced Settings tab

This tab allows you to control specific Flow Analysis options, such as analysis depth, support for multithreading APIs, and violation reporting verbosity. For details, see [Configuring Flow Analysis Analysis Options](#).