Microsoft

Microsoft® Visual Studio® Team System 2008 Development Edition

What's New in Visual Studio Team System 2008 Development Edition

Code Analysis

The code analysis tools in Microsoft® Visual Studio® Team System 2008 Development Edition perform extensive checks for code defects, which are presented as warnings in the error window. Code analysis has been enhanced with:

Rules Extension and Enhancement

Several rules have been enhanced by providing greater accuracy, particularly around naming rules.

Spelling Checker with Custom Dictionary Support

Use the spelling checker for resource strings as well as class, method, and property names.

Better Control Over Supression from the Error List

Suppress code analysis issues from the error window at either the project level or in-source.

Auto-Suppress Generated Code Option

Automatically suppress error messages from generated code. This is useful for designer-generated code.

Code Analysis Policy Improvements

When copying the settings from the server to your project, you now have the option to replace your local selection, or merge the policy rules with your local project rules. Also, you now have more complete information about policy violations to determine source.

Code Metrics

Code metrics are a set of software measures that give developers better insight into the code they are developing. By taking advantage of code metrics, developers understand which types and/or methods should be reworked or more thoroughly tested. In addition, development teams identify potential risks, understand the current state of a project, and track progress during software development.

Profiling Tools

Profiling tools in Visual Studio Team System 2008 Development Edition enable developers to measure, evaluate, and target performance-related issues in their code.

The following features have been added to the Profiling Tools:

64-Bit Support

The Profiler now includes support for both the 64-bit applications that run on 64-bit operating systems and hardware and the 32-bit applications executed on 64-bit operating systems and hardware.

Full Allocation Stacks

The Profiler has full call stacks for allocation. This is useful for allocation that occurs in non-user code, but is indirectly caused by user actions. By using the full stack, you can see exactly which parts of your code are indirectly causing the allocation.

You can collect allocation data by configuring settings in the performance session property page. Use the allocation view in the performance report to see your results.

Line-Level Sampling Data

Profiling tools now include an instruction pointer and line views in performance reports. Also, the modules view now includes line information.

Report Noise Reduction

You can configure performance reports for noise reduction. This limits the amount of data in the Call Tree view and the Allocation view. By using noise reduction, performance problems are more prominent. This is helpful when you analyze performance reports.

Runtime Control

Profiling tools includes a runtime control. The runtime control starts automatically with the profiler. It can be paused and resumed for performance data logging. In addition, you can use the runtime control to start the application with logging paused. This enables you to skip data collection on application startup. When you use the runtime control, you can manually insert annotations in the performance data when events of interest occur in the application lifetime. You can filter the data on your annotations later.

Filtered Analysis

You can now filter performance reports on timestamp, process, thread, and marks. You can use the show query button to get the filtered analysis. Also, you can use the /summaryfile option from the command line.

Compare Reports

The Profiler now supports the comparison of reports. You can compare a report either by using the Performance Explorer or the /diff on options from the command line.

Improved Chip Counter Support

Profiling tools provide new friendlier chip-counter names (For example: "L2 Misses," "ITLB Misses," "Mispredicted Branches"). You can modify XML files to further configure counters for a specific architecture.

(over)

Profiling Tools (continued)

Windows Counter Support

The Profiler now collects Windows counters (for example, "% Processor Time," "% Disk Time," "Disk Bytes/sec," "Page Faults/sec"). You can use either the Windows counters node in the performance sessions properties page or the /wincounter option from the command line. The marks view displays the counters. You can use counters as filtering endpoints.

Compressed Repost Files

Profiling tools enable you to generate small compressed report files that open quickly. This is because these files, which are created from full reports, are analyzed already. You can either right-click the report in the Performance Explorer and choose Save Analyzed or use the /summaryfile option from the command line.

Hot Path

The Profiler now has the ability to automatically expand the most expensive code path in the Call Tree and Allocation views of the performance report.

Copy Report View Data to HTML

The Profiler includes support for rich reports in the clipboard. You can copy and paste rich data (tables with headers and values) from the performance reports.

Windows Communication Foundation Support

Profiling tools now support Windows Communication Foundation (WCF).

Load and Web Test Integration in Visual Studio Team System 2008 Team Suite

You can create performance sessions for Web and Load tests from Test View and Test Results.

Note: This feature applies only to Visual Studio Team System 2008 Team Suite.

Part No. 098-108999