Module Introduction

PURPOSE:

– The purpose of this module is to provide an overview of Metrowerks CodeTEST Software Analysis Tools.

OBJECTIVES:

– Describe CodeTEST

– Identify and describe the features of the Analysis Tools

CONTENT:

– 9 pages

– 2 questions

LEARNING TIME:

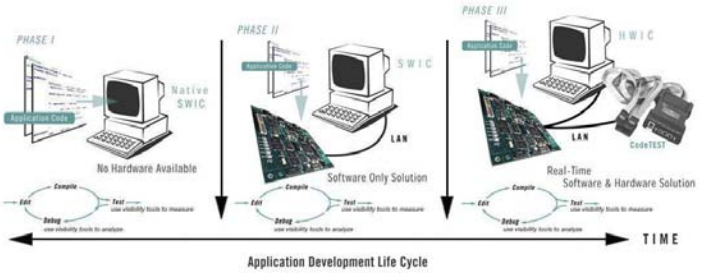
– 20 minutes

Welcome to the module on CodeTEST Software Analysis Tools. This module will describe the features of CodeTEST, explaining how it provides detailed visibility into embedded software and enables precise analysis of software in order to improve quality and reduce development costs.

Upon completion of this module, you'll be able to describe the main features of the CodeTEST Software Analysis Tools, the four main CodeTEST modules, and the benefit that each CodeTEST module provides for embedded developers.

What is CodeTEST?

- Software verification toolset
  - Performance
  - Trace
  - Memory
  - Coverage
- Various connection and data collection methods
- Deliver your code with confidence



CodeTEST is a software analysis tool for C and C++ developers. It provides detailed visibility into how the software in an embedded system is actually executing. This enables developers to identify the root causes of software errors, improve performance, and ensure that all code is tested prior to shipping. The implementation of CodeTEST within developers' established development processes can enable them to shorten their design cycles while improving overall quality.

With CodeTEST, you can quickly identify and resolve software errors, such as crashes or memory leaks, that frustrate and delay development. CodeTEST provides performance, trace, memory, and coverage analysis. CodeTEST Software Analysis Tools are designed to fit into any development environment with a scalable, modular design.

Additionally, CodeTEST allows for a choice of connection and data collection methods including: CodeTEST Native for host based development targeting a simulator; CodeTEST Software In-Circuit (SWIC), which enables connection and data collection via Ethernet; or CodeTEST Hardware In-Circuit (HWIC), which provides the highest level or accuracy and performance by utilizing a dedicated hardware probe for data collection.

CodeTEST also provides RTOS aware analysis for a variety of operating systems such as VxWorks, Linux, OSE, and numerous others. With CodeTEST software analysis tools, developers can save time, money, and have more confidence in their systems at hand-off.

CodeTEST Performance Analysis

- Provides detailed software timing
  - Measure up to 128,000 functions simultaneously
  - Monitor specific sections of code with point-to-point timing measurements
- Enables you to guarantee software performance
  - Identify exactly where the system is spending its time
  - Determine interrupt service routine execution timing
  - Use call-pair data to optimize cache usage

CodeTEST Performance allows you to boost embedded system productivity by precisely identifying embedded software-related bottlenecks. It allows you to measure up to 128,000 real-time function executions simultaneously and measure point-to-point timers to highest possible fidelity.

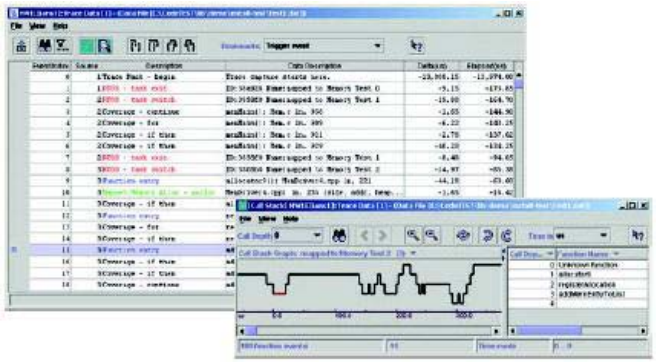
The performance analysis allows you to determine the percentage of the CPU functions that various tasks consume. You can also determine the impact of the execution of interrupt service routines and pinpoint algorithmic errors and call-pair distribution.

You are also able to generate easy-to-view performance reports, and profiles of uncharted parent/child function performance.

CodeTEST Trace

Mouse over each bulleted point for more information.

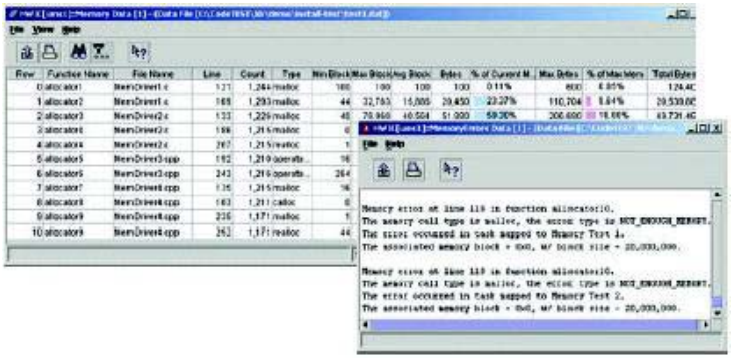
- Validate execution of your design flow and specifications
  - Unmatched accuracy in tracing execution history
  - Locate, view and analyze problem areas quickly and easily
- Trace execution at different levels
  - RTOS level
  - Control-flow level
  - Source level



CodeTEST Trace makes it easy to validate the execution of your design flow and specifications. It provides you with the ability to track in detail the exact execution history of a program at the RTOS, control-flow, and source level. This makes it easier and faster to determine the root and triggering causes of particular issues.

CodeTEST Memory

- Isolate memory leaks and errors
- Optimize memory allocation
- Improve system throughput and efficiency
- Tracks dynamic memory allocations
- Reports memory errors



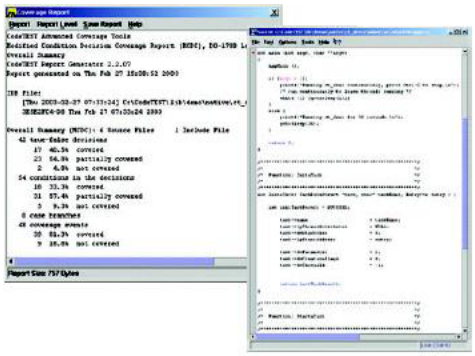
CodeTEST Memory saves you time, resources, and money by quickly pinpointing system bugs caused by memory leaks or errors.

Using CodeTEST Memory you are able to isolate memory leaks and errors, optimize memory allocation, and improve system throughput and efficiency.

CodeTEST Memory also tracks dynamic memory allocations and reports memory errors associated with the original source.

CodeTEST Coverage

- Find out what percentage of code executed in your test
  - Pinpoint untested or dead code
- Get actual coverage measurements
  - Statement coverage (SC)
  - Decision coverage (DC)
  - Modified condition/decision coverage (MC/DC)
- Qualifiable under the safety-critical FAA-recommended guideline (DO-178B)



CodeTEST Code Coverage Analysis allows developers to view exactly which pieces of software have been executed during testing. This allows them to ensure that the product is thoroughly tested prior to shipping. This also potentially reduces downstream costs which could arise from shipping untested code.

CodeTest Statement Coverage (SC) and Decision Coverage (DC) provides comprehensive statement and decision coverage that invokes every point of entry and exit in the program at least once. Equipped with CodeTEST's decision coverage tool, you can also ensure that each decision in the program has been taken on all possible outcomes at least once. CodeTEST Modified Condition/Decision Coverage (MC/DC) provides the most comprehensive code coverage recognized in the industry today by tracking whether or not every variable of every decision in the application has been executed at least once.

CodeTEST Code Coverage is one of the few tools qualifiable under FAA guidelines DO-178B for airworthiness. CodeTEST is capable of producing coverage data for all certification levels including, DO-178B level A, which requires MC/DC coverage.

Question

Is the following statement true or false? Click "Done" when you are finished.

"Implementing CodeTEST can help shorten design cycles while improving overall quality.**"**

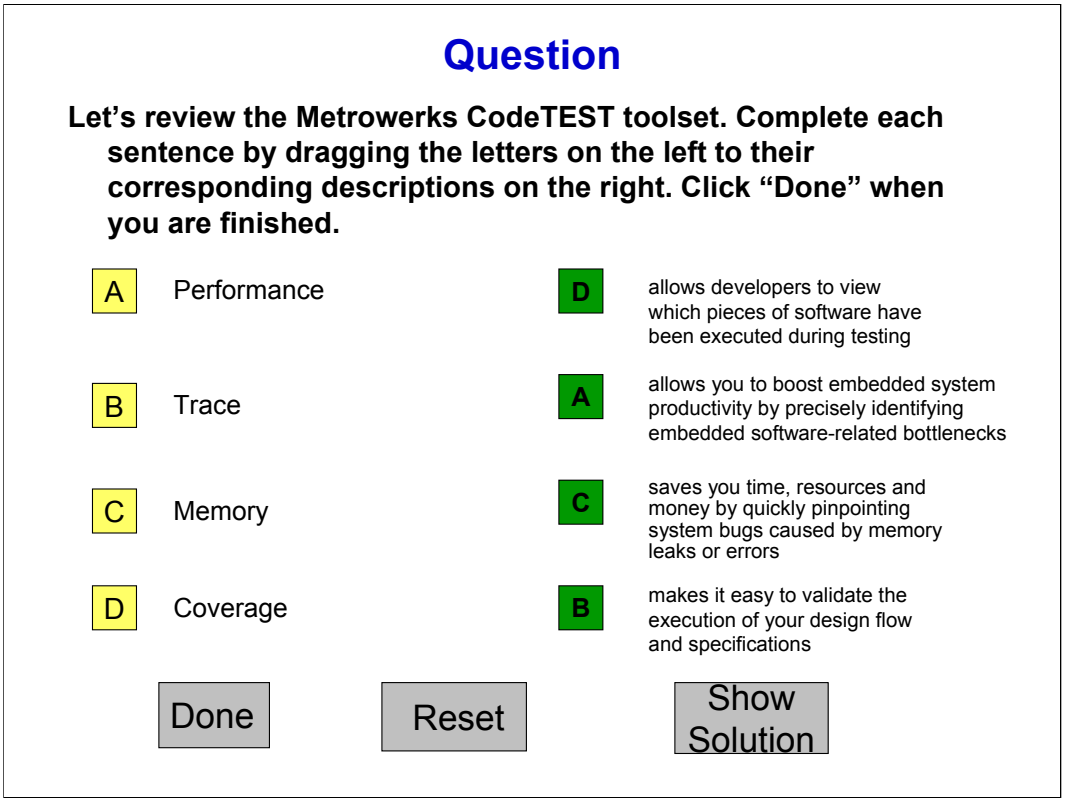
True

False

Let's take a moment to review the CodeTest Analysis Tools.

Answer:

The implementation of CodeTEST within developers' established development processes can enable them to shorten their design cycles while improving overall quality. With CodeTEST, you can quickly identify and resolve software errors, such as crashes or memory leaks, that frustrate and delay development.



Let's take a moment to review what you have learned about the CodeTEST toolset.

Answer:

CodeTEST Performance allows you to boost embedded system productivity by precisely identifying embedded software-related bottlenecks. CodeTEST Trace makes it easy to validate the execution of your design flow and specifications. CodeTEST Memory saves you time, resources and money by quickly pinpointing system bugs caused by memory leaks or errors. CodeTEST Code Coverage Analysis allows developers to view exactly which pieces of software have been executed during testing.

Module Summary

- CodeTEST toolset
  - Performance: identify execution bottlenecks
  - Trace: software execution history
  - Memory: view dynamic memory allocations
  - Coverage: document test completeness

Let's review what we discussed in this module.

CodeTEST Software Analysis Tools overcome the complexities of the embedded software environment from early host-based design and development to final phase test and validation. The CodeTEST toolset comprises of the most comprehensive embedded software visibility, analysis, measurement and verification suite available today. Its tools include CodeTEST Performance, CodeTEST Trace, CodeTEST Memory, and CodeTEST Coverage.