

補助機能

コーディングガイドラインへの適合状況を可視化

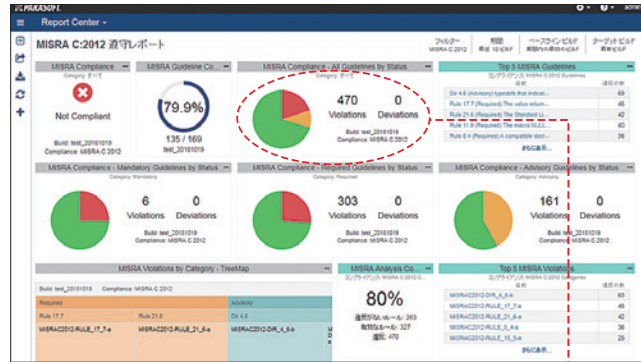
ソフトウェアの品質状況をリアルタイムに表示、レポートを自動生成

コンプライアンスパック

- MISRA C:2012、AUTOSAR C++ 14、CERT C コーディングスタンダードなどの遵守状況をリアルタイムに表示
- コーディングガイドラインに則った遵守サマリーレポートや逸脱のレポートを提供

C++test のコンプライアンスパックは、MISRA C:2012、AUTOSAR C++14、CERT C コーディングスタンダードなどの遵守状況をリアルタイムに表示するダッシュボード画面の提供と、コーディングガイドラインに則った遵守サマリーレポートや逸脱のレポートを提供しています。コーディングガイドラインの遵守状況の説明責任を果たすことが容易になるだけでなく、未遵守箇所を早期に特定し必要な措置を講ずることにより、欠陥のあるソフトウェアに関連するビジネスリスクを排除することが可能になります。

● MISRA C:2012 遵守用ダッシュボード画面

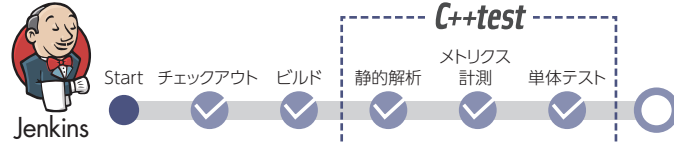


● MISRA Compliance:2016のレポート画面



CI (継続的インテグレーション) ツール連携

● 静的解析および単体テスト実行プロセスを自動化
Jenkins(CIツール)を活用することにより、静的解析および単体テスト実行プロセスを定期的に自動実行することができます。ビルドやテストの実施漏れを防ぎ、エラーや欠陥のフィードバックサイクルを早めることで、開発者に負担をかけず、品質の高いソフトウェアをより高速にリリースすることができます。



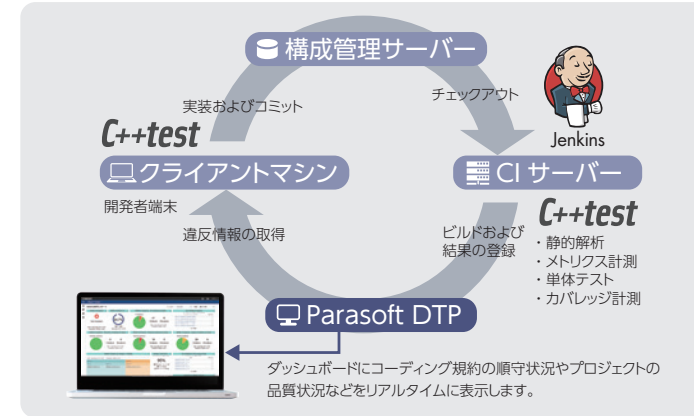
レポート生成

- テスト結果をレポート出力(HTML、PDF、XML、CSV形式)
- テスト実行時の追加情報レポート機能

テスト結果をレポートとして出力できます。コーディング規約が、守られていることを証明する場合などに利用できます。テスト実行に関する詳細な追加情報を出力することも可能です。



● コンプライアンスパックの構成例



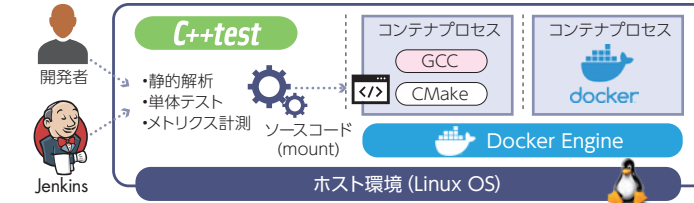
MISRA Compliance:2016のレポートを自動生成

*MISRA Compliance:2016で規定された4種類のレポートを自動生成します。MISRA準拠を提示するための時間と労力を大幅に削減することができます。

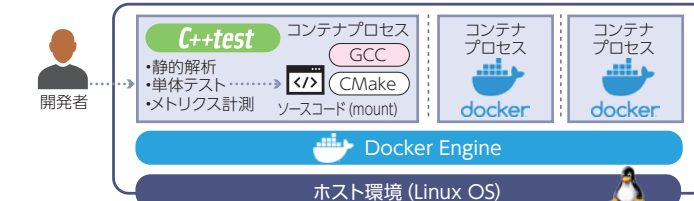
- ガイドライン施行計画書(Guideline enforcement plan)
- ガイドラインの再分類計画書(Guideline re-categorization plan)
- 逸脱報告書(Deviation record)
- ガイドラインコンプライアンスサマリー(Guideline compliance summary)

Docker連携

● Dockerコンテナ上のビルド環境を利用したテストが可能
ホスト環境上にインストールされたC++testから、Dockerコンテナ上に存在するビルド環境を利用し、静的解析および単体テストを実行することができます。Jenkinsからも実行可能なため、CI環境にも組み込むことができます。



● Dockerイメージの配布で、テスト環境構築作業が不要に
C++testは仮想環境上でも動作可能であるため、C++testを組み込んだ「Dockerイメージ」を開発者に配布することができます。これにより、ビルド環境だけでなく、C++testのテスト環境の構築作業もゼロにすることができます。



C++test

動作環境

- システム構成 バージョン: C++test 10.4.3 ※10.4.3で対応したのが緑字です。

CPU	2.0GHz以上、マルチCPU構成を推奨
メモリ	4GB以上(推奨:8GB以上)
Windows	OS Windows 7 (32bit/64bit) Windows 8.x (32bit/64bit) Windows 10 (32bit/64bit) Windows Server 2008 (64bit) Windows Server 2012 (64bit) Windows Server 2016 (64bit)
Linux	OS Linuxカーネル2.4以上、glibc 2.12以上、x86互換プロセッサ
Linux (32bit)	OS Linuxカーネル2.4以上、glibc 2.12以上、x86互換プロセッサ
Linux (64bit)	OS Linuxカーネル2.6以上、glibc 2.12以上、x86_64互換プロセッサ
ホストコンパイラ	Visual C++ 6.0 (Visual Basic 6.0), 9.0 (Visual Studio 2008), 10.0 (Visual Studio 2010), 11.0 (Visual Studio 2012), 12.0 (Visual Studio 2013), 14.0 (Visual Studio 2015), 14.1x (Visual Studio 2017), 14.2(Visual Studio 2019) GNU gcc/g++ x86:4.0.x, 4.1.x, 4.2.x, 4.3.x, 4.4.x, 4.5.x, 4.6.x, 4.7.x, 4.8.x, 4.9.x, 5.x, 6.x, 7.x, 8.x, 9.x GNU gcc/g++ x86-64:4.8.x, 4.9.x, 5.x, 6.x, 7.x, 8.x, 9.x National Instruments LabWindows/CVI 2013 Clang C/C++ Compiler v2.9 for Win32 (静的解析のみ) National Instruments LabWindows/CVI 2015 Clang C/C++ Compiler v3.1 for Win32 (静的解析のみ) National Instruments LabWindows/CVI 9.0 (Static Analysis Only)
ホストコンパイラ	GNU gcc/g++ 4.0.x, 4.1.x, 4.2.x, 4.3.x, 4.4.x, 4.5.x, 4.6.x, 4.7.x, 4.8.x, 4.9.x, 5.x, 6.x, 7.x, 8.x, 9.x Intel(R) C++ Compiler v18.0 Green Hills MULTI for Linux x86 Native v4.2.x
ホストコンパイラ	GNU gcc/g++ 4.0.x, 4.1.x, 4.2.x, 4.3.x, 4.4.x, 4.5.x, 4.6.x, 4.7.x, 4.8.x, 4.9.x, 5.x, 6.x, 7.x, 8.x, 9.x Clang C/C++ Compiler v3.9, v4.0, v5.0, v6.0, v8.0 Intel(R) C++ Compiler v 18.0 Green Hills MULTI for Linux x86 Native v4.2.x

● C++testをプラグインできる統合開発環境

- Eclipse for C/C++ Developers 3.8, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9
- Visual Studio 2008, 2010, 2012, 2013, 2015, 2017, 2019
- Renesas e2studio 5.1
- Wind River Workbench 3.3, 4.0
- Arm Development Studio 5 (DS-5) v5.18以上
- QNX Software Development Platform 7.0
- Texas Instruments Code Composer Studio IDE 5.x, v6.0, v7.4, v8.0

【開発元】



【総販売代理店】



テクマトリックス株式会社

ソフトウェアエンジニアリング事業部
〒108-8588 東京都港区三田3-11-24 国際興業三田第2ビル
TEL : 03-4405-7853 FAX : 03-6436-3553
URL : https://www.techmatrix.co.jp/
E-MAIL : parasoft-info@techmatrix.co.jp

● 共通項目: クロスコンパイラ

Renesas	Windows	SuperH RISC engine C/C++ Compiler V5.1 (静的解析のみ)、V9.03 (C++については限定サポート)、V9.04 (C++については限定サポート) M16C/R8C C Compiler 5.4x (静的解析のみ) RX C/C++ Compiler 2.2x, 2.5x ※下記のコンパイラについてはお問合せ先までご連絡ください。 R32C/100 Series C Compiler RH850 Family Compiler R178 (78K0R) C Compiler H8S, H8/300 SERIES C/C++ Compiler (静的解析のみ) V850 Optimizing C Compiler (静的解析のみ) M32R Family C/C++ Compiler (静的解析のみ) 78K0 C Compiler (静的解析のみ)
Cypress Semiconductor	Windows	FR Family SOFTUNE C/C++ Compiler V6
Green Hills Software	Windows	Green Hills Software Compiler for PPC v3.5, v4.0.x, v4.2.x, v5.0.x, v2013.1.x, v2017.1.x Green Hills Software Compiler for V850 v5.1.x, v2013.5.x, v2014.1.x, v2017.5.x Green Hills Software Compiler for RH850 v2013.5.x, v2014.1.x, v2017.5.x Green Hills Software Compiler for ARM v2014.1.x, v2017.5.x (静的解析のみ) Green Hills Software Compiler for ARM64 v2014.1.x, v2017.5.x (静的解析のみ)
	Linux	Green Hills Software Compiler for V850 v2013.5.x, v2014.1.x, v2017.5.x Green Hills Software Compiler for RH850 v2013.5.x, v2014.1.x, v2017.5.x Green Hills Software Compiler for ARM v2014.1.x, v2017.5.x (静的解析のみ) Green Hills Software Compiler for ARM64 v2014.1.x, v2017.5.x (静的解析のみ) Green Hills Software Compiler Native v4.2.x
Arm	Windows	Arm RealView 4.1 Arm Compiler 5.0, 6.6, 6.9 Arm GNU GCC 4.5.x
	Linux	Arm Compiler 5.0, 6.6, 6.9
Texas Instruments	Windows	TI TMS320C6x C/C++ Compiler v7.3, v7.4, v8.2 TI TMS320C2000 C/C++ Compiler 6.2, v16.9, v18.1 TI MSP430 C/C++ Compiler v4.0, v18.1 TI MSP430 C/C++ Compiler GNU GCC 6.x TI ARM C/C++ Compiler v5.1.x, v18.1 TI ARM C/C++ Compiler GNU GCC 7.x
	Linux	TI TMS320C6x C/C++ Compiler v7.3, v8.2 TI TMS320C2000 C/C++ Compiler v16.9, v18.1 TI MSP430 C/C++ Compiler v18.1 TI MSP430 C/C++ Compiler GNU GCC 6.x TI ARM C/C++ Compiler v5.1.x, v18.1 TI ARM C/C++ Compiler GNU GCC 7.x
Wind River	Windows	Wind River GCC 3.3.x, 3.4.x, 4.1.x, 4.3.x, 4.8.x Wind River DiAB 5.7x - 5.9.x
	Linux	Wind River GCC 3.4.x, 4.1.x, 4.3.x, 4.8.x Wind River DiAB 5.7x - 5.9.x
Cosmic Software	Windows	Cosmic Software 68HC08 C Cross Compiler v4.6.x (静的解析のみ)
Intel	Linux	Intel® C++ Compiler v18.0
HighTec	Windows	GCC for Tricore 4.9.x
GNU	Windows	GNU gcc/g++ 4.0 - 8.x GNU gcc/g++ (x86_64) 4.0 - 8.x ※GCCベースのクロスコンパイラについてはお問合せ先までご連絡ください。
Clang	Linux	Clang C/C++ Compiler v3.9, v4.0, v5.0, v6.0
IAR	Windows	IAR Compiler for ARM v6.1x (C言語のみ), v6.3x (C言語のみ), v6.6x, v7.4x, v7.8x, v8.11.x, v8.22, v8.40 IAR Compiler for MSP430 v5.4x (C++言語は静的解析のみ), v6.1 (静的解析のみ) IAR Compiler for RX v2.5x, v2.6, v3.10.x IAR Compiler for STM8 v1.4x (静的解析のみ) IAR Compiler for RL78 v3.1x IAR Compiler for M16C & R8C v3.5x (静的解析のみ)
Keil	Windows	Arm C/C++ Compiler RealView 4.1 for uVision Arm C/C++ Compiler 5.0 for uVision Keil C51 Compiler v8.x (静的解析のみ) Keil C166 7.0 (静的解析のみ)
QNX	Windows	QNX GCC x86 4.2.x, 4.4.x, 5.x QNX GCC 5.x QNX GCC (ARM) 5.x QNX GCC (ARM64) 5.x
	Linux	QNX GCC x86 5.x QNX GCC 5.x QNX GCC (ARM) 5.x QNX GCC (ARM64) 5.x
Altium TASKING	Windows	Altium TASKING VX-toolset for TriCore C/C++ Compiler 4.0, 6.0, 6.2 Altium TASKING 80C196 C Compiler v6.0.1 (静的解析のみ) Altium TASKING classic compiler for C166/ST10 v6.0 (静的解析のみ)
Freescale	Windows	Freescale CodeWarrior ANSI-C/C++ Compiler 5.0.x for HCl2 (静的解析のみ) Freescale CodeWarrior C/C++ Compiler v6.0 for ColdFire (静的解析のみ) Freescale C/C++ Compiler v5.1 for Embedded ARM (静的解析のみ)
Microchip	Windows	MPLAB C Compiler for dsPIC v3.2x (静的解析のみ) MPLAB C32 Compiler for PIC32 v2.0x (静的解析のみ)
National Instruments	Windows	LabWindows/CVI 2013 Clang C/C++ Compiler v2.9 for Win32 (静的解析のみ) LabWindows/CVI 2015 Clang C/C++ Compiler v3.1 for Win32 (静的解析のみ) LabWindows/CVI 9.0 (静的解析のみ)
Mentor Graphics	Windows / Linux	CodeSourcery Sourcery G++ Lite 2009q1-203 (静的解析のみ)
Embarcadero	Windows	Embarcadero C++ Compiler 6.2x, 6.9x for Win32 (静的解析のみ)

C++testの情報は www.techmatrix.co.jp/product/ctest/

●掲載されているあらゆる製品名は、各社の商標あるいは登録商標です。



for IEC 61508
for ISO 26262
for IEC 62304

ソフトウェアの品質向上と 効率的な開発の実現をサポート

- コーディング規約チェック
- フロー解析
- 単体テスト
- カバレッジ計測
- アプリケーションモニタリング
- 組み込みソフトウェア開発での利用
- 規格コンプライアンス
- コンプライアンスパック
- CIツール連携
- レポート生成

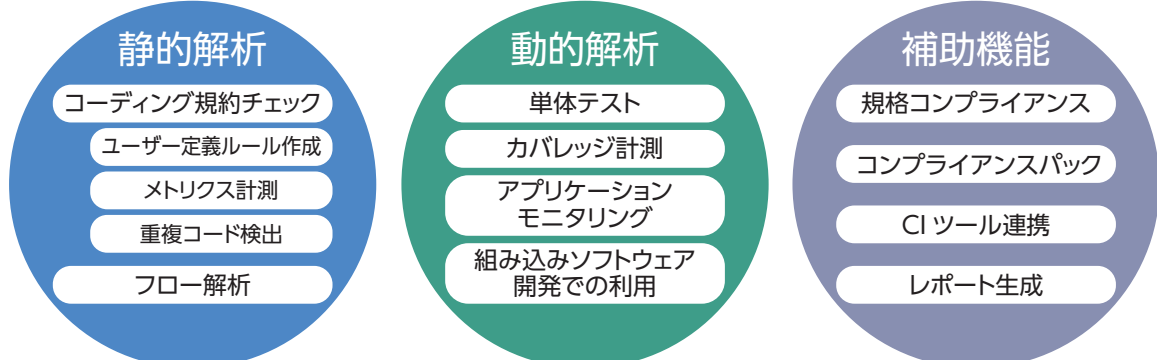
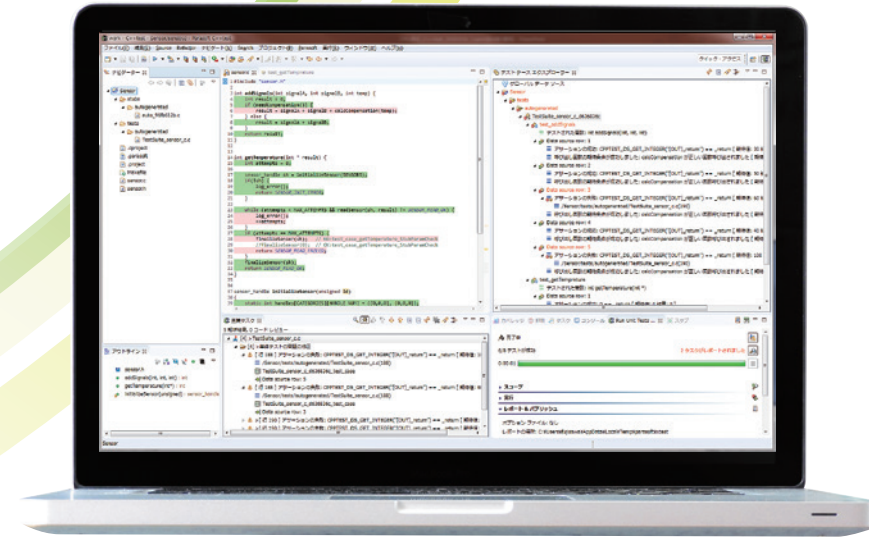


このカタログの印刷には、環境に優しい植物性インクを使用しています。

C言語/C++言語対応 静的解析・単体テストツール



C++testは、静的解析（コーディング規約チェック/フロー解析）、単体テスト、カバレッジの計測、実行時メモリエラー検出、効率的な運用や規格遵守を補助する機能などを搭載したC言語/C++言語対応のオールインワンテストツールです。MISRA C/C++, AUTOSAR C++14コーディングガイドライン、CERT C/C++コーディングスタンダードなどで定められた規約に基づくコーディングの支援や、単体テストやアプリケーション実行時に自動的にカバレッジを計測するなど、さまざまな要件に対応し、ソフトウェアの品質向上とテスト工数の大幅削減をサポートします。



機能安全認証取得

IEC 61508/ISO 26262/IEC 62304準拠
第三者認証機関による認証取得済み

規格コンプライアンス

C++testは、第三者認証機関であるTÜV SÜD社よりIEC 61508およびISO 26262、IEC 62304に準拠したテストツールとして認証を取得済みです。

- 《機能安全規格準拠に役立つルールセット》
- HISソースコードメトリクス チェックルール
HIS (Herstellereinitiative Software) はドイツの自動車メーカーによる団体に、車載システムの設計や品質保証に関する標準を策定しています。
 - MISRA C:1998, MISRA C:2004, MISRA C++:2008, MISRA C:2012規約チェックルール
MISRA (Motor Industry Software Reliability Association) が作成した「自動車産業におけるCプログラミングガイドライン」をもとに作成されています。
- 《医療機器ソフトウェア安全規格対応ルールセット》
- FDA C/C++ (米国食品医薬品局)に関するルール
FDA (米国食品医薬品局) が定めた、医療機器ソフトウェアならびに医療機器製造に使用するソフトウェアの検証原則に基づくルールです。



for IEC 61508
for ISO 26262
for IEC 62304

静的解析

静的解析で「致命的なバグ」を見逃さない!

バグの作り込みを抑制、ソースコードの可読性と保守性を強化

コーディング規約チェック

- MISRA, AUTOSAR, CERTなどさまざまなコーディング規約に対応
 - 独自のルールセットの作成やコーディングルールの追加・編集が可能
- MISRA C/C++, AUTOSAR C++14コーディングガイドライン、CERT C/C++コーディングスタンダードなどの規約に対応しています。静的解析のルールは、4,000ルール以上搭載。バグの作り込みを抑制し、ソースコードの可読性、保守性、移植性に優れた高品質で寿命の長いソースコードの実装を支援します。

● MISRA C:2012に完全対応

- Automotive Pack
 - AUTOSAR C++14 Coding Guidelines
 - HIS Source Code Metrics
 - High Integrity C++
 - ISO26262 ASIL A Unit Testing
 - ISO26262 ASIL B and C Unit Testing
 - ISO26262 ASIL D Unit Testing
 - MISRA C 1998
 - MISRA C 2004
 - MISRA C 2012
 - MISRA C++ 2008

● CERTなどセキュリティルールを搭載

- Security Pack
 - CWE SANS Top 25 Most Dangerous Software Errors
 - OWASP Top 10 2017
 - Payment Card Industry Data Security Standard
 - SEI CERT C Guidelines
 - SEI CERT C++ Rules
 - Security Rules
 - UL 2900

- コーディングルールセット (抜粋)
- AUTOSAR C++14コーディングガイドライン
 - MISRA C:1998, MISRA C:2004, MISRA C++:2008, MISRA C:2012 規約チェックルール
 - CERT C, CERT C++ チェックルール
 - HIS ソースコードメトリクス チェックルール
 - FDA C/C++ 推奨ルール
 - OWASP TOP10, PCIDSS, CWE/SANS 最も危険なプログラミングエラーTOP25など、セキュリティに関するルール
 - IPA/SEC コーディング作法ガイド チェックルール

ユーザー定義ルール作成

ユーザー定義コーディングルールを作成する「RuleWizard」を搭載。GUIを使用して、視覚的にコーディングルールを作成できます。また、違反となるソースコードを読み込ませることで、ユーザー定義ルールを自動生成することが可能です。

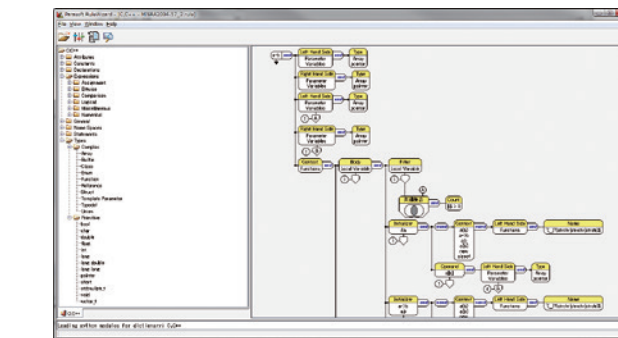
メトリクス計測

メトリクスの自動計測で、ソフトウェアの品質を定量的に評価します。バグの温床となる複雑なソースコードを検出します。早期にリファクタリングすることで、バグの防止、テストしやすいソースコードの実装が可能です。

重複コード検出

保守性を低下させる要因であるコードクローンを検出します。クローンと見なす条件の柔軟な設定が可能であり、実用的なチェックが可能です。

● 独自のコーディングルールの作成



プログラムのあらゆるパスをシミュレートし、バグを早期に発見

フロー解析

- ソースコードを解析し、関数・ファイルをまたがったバグに至るまでの処理の流れをレポート
- フロー解析機能のルール数は114種類

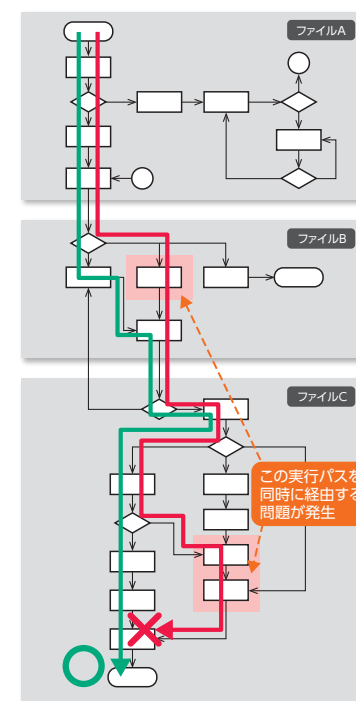
フロー解析は、ソースコードの処理フローを解析します。プログラムを実際には実行せず、論理的に実行して、メモリ破壊などの問題があるパスを指摘します。複雑なアプリケーションでも、複数のファイル、メソッドにまたがるパスを自動的にトレースし、NULLポインターの間接参照やバッファオーバーフローなどプログラムの動作に致命的な影響をもたらすバグを早期に発見します。また、Parasoft DTPと連携することにより、フロー解析におけるデータフローのシミュレート結果をより詳しく表示できます。

検出可能な項目 (抜粋)

- メモリリーク/リソースリーク
- バッファオーバーフロー
- NULLポインターの参照
- 未初期化変数の参照
- 整数オーバーフロー
- ゼロ除算
- 配列の境界外アクセス
- イテレーター範囲外アクセス
- セキュリティ脆弱性
- デッドロック
- 不適切な排他制御



● ファイルをまたがる問題を検出 (イメージ図)



動的解析

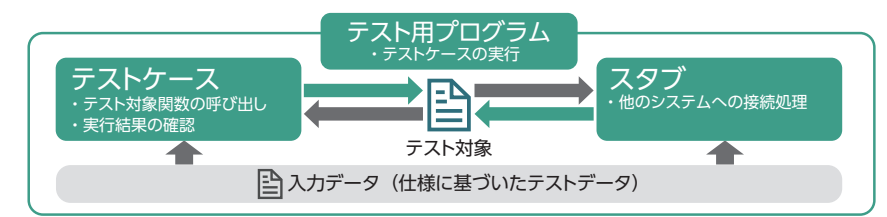
単体テスト工数を大幅削減

GUI操作でテストケースとスタブを生成。テストの実行と回帰テストを自動化

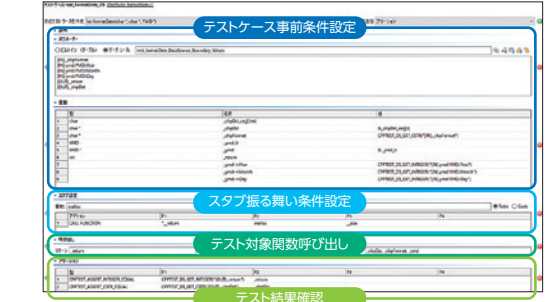
単体テスト

- テストドライバー、スタブ、テストケースを生成し、ソフトウェアの単体テストを自動化
- Excelで管理しているテストデータ、CppUnitのテストケースを活用

単体テスト時の課題であった「テストのためのコーディング」を行うことなく、単体テストを行うことができます。テストケースエディターで、GUIの操作のみでテストケースの作成やスタブの自動生成、また、スタブコールバックフレームワークにより作成済みのスタブの流用や、より高度なスタブの振る舞いを簡単に実装できます。また、外部テストデータの取り込みや、既存のテスト資産の再利用が可能です。



● テストケース・スタブを1画面でコントロール



9種類のカバレッジを計測。単体テストの網羅性を視覚的にレポート

カバレッジ計測

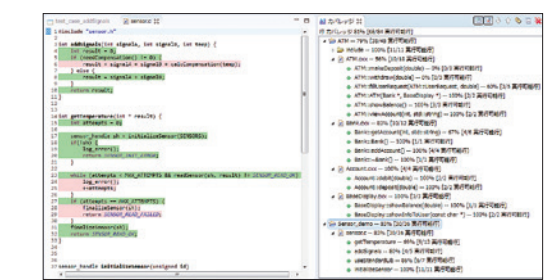
- プロジェクト、ファイル、関数単位でカバレッジの計測が可能
- 実行/未実行の箇所をハイライト表示

単体テスト実行時に自動的に9種類のカバレッジを計測します。複数のカバレッジを同時に計測することもできます。画面上で実行/未実行の箇所を分かりやすくハイライト表示するため、視覚的に確認することができます。

C++test がレポートするカバレッジ

- ステートメントカバレッジ (C0:命令網羅率)
- 判断文カバレッジ (C1:分岐網羅率)
- 単純条件カバレッジ (C2:条件網羅率)
- MC/DC (Modified Condition/Decision Coverage)
- 関数カバレッジ
- コールカバレッジ
- 行カバレッジ
- 基本ブロックカバレッジ
- パスカバレッジ

● 9種類のカバレッジを自動的に計測



行カバレッジの計測結果の例

アプリケーション実行時に、メモリ関連エラーの検出とカバレッジを計測

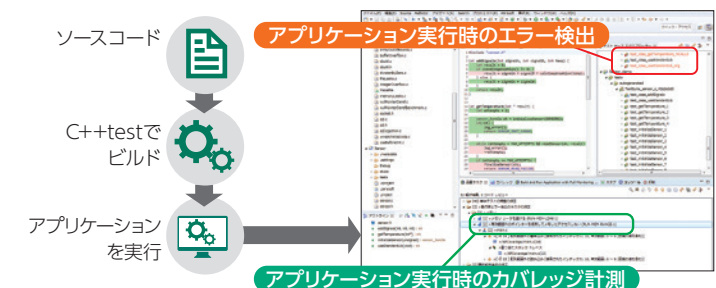
アプリケーションモニタリング

● アプリケーション実行時のカバレッジを計測

システムテストを実施しながらカバレッジを計測することで、テストの抜けや漏れを効率的に確認できます。

● アプリケーション実行時に発生したエラーを自動検出

不正メモリアクセス・メモリ破壊・メモリーリーク・未初期化メモリの参照・NULLポインター参照などを検出し、スタックトレースと併せて問題をレポートします。



実機やシミュレーターで、単体テスト・カバレッジ計測が可能

組み込みソフトウェア開発での利用

● ホスト、シミュレーター、ターゲット環境で実行可能

C++testをインストールしたホストマシンだけでなく、実機(ターゲット機)や開発環境などに付属するシミュレーター上でも単体テスト、カバレッジ計測(単体テスト時とアプリケーション実行時)および実行時メモリエラー検出を実行できます。

● 組み込みソフトウェアでの単体テスト実行イメージ

