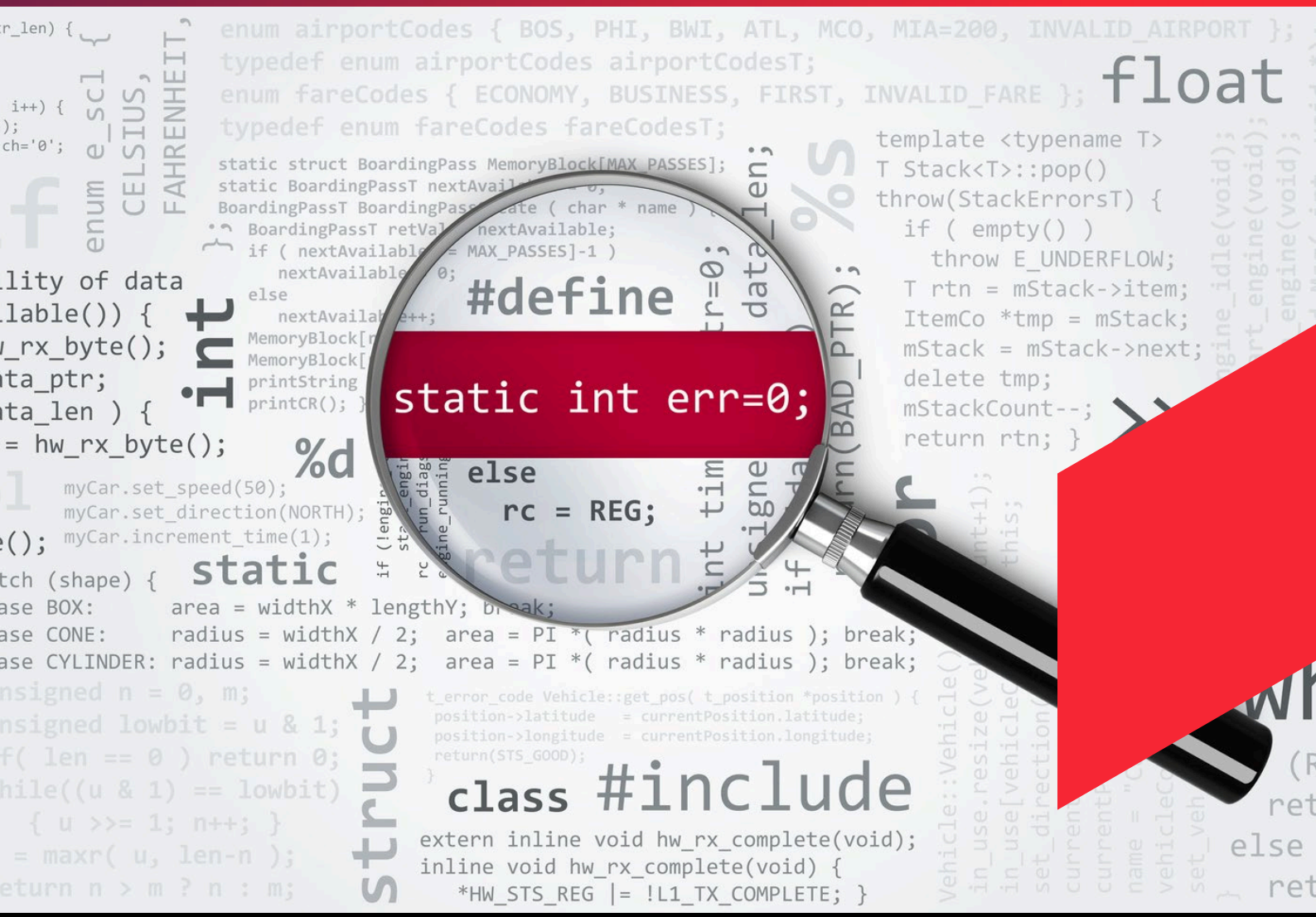
VECTOR >

はじめての PC-lint Plus

静的解析をよりシンプルに



目次

発行元：ベクター・ジャパン株式会社

記述されている内容や製品画面の表示名称は予告なく変更されることがあります。
本稿は、PC-lint Plus バージョン 2025 以降を対象としています。

はじめに

近年、ソフトウェアの規模、複雑さが増し、開発の効率化やコスト削減が求められると同時にソースコードの品質向上やセキュリティ脆弱性への対応も重視されています。そこで重要な役割を果たすのが静的解析ツールです。

本稿では、「静的解析とは何か？」といった静的解析の基礎をはじめ、ベクターが開発、販売する静的解析ツール「PC-lint Plus（ピーシーリントプラス）」についてご紹介いたします。また、評価用ライセンスもご用意しておりますので、本資料をチュートリアルとしてもご活用いただけますと幸いです。評価用ライセンスの入手方法については巻末「8. 評価用ライセンスのご案内とお問い合わせ窓口」をご確認ください。

1. 静的解析とは

本章ではまず、静的解析の基礎知識と重要性について説明します。

1.1 静的解析とは

静的解析とは、ソースコードを実行せずに行なう検証です。ソースコードを解析してソースコード上の欠陥になりえる部分や、保守性・セキュリティ上、問題になりそうな部分などを検出します。また、ソフトウェア開発では「MISRA」や「CERT」、「CWE（共通脆弱性タイプ一覧）」などのコーディングスタンダード（各団体が定めるソースコードの書き方を統一するためのガイドライン）が採用されることも多くあり、静的解析ではこれらの規約に違反していないかのチェックも行います。

1.2 動的テストとの違い

静的解析のような、プログラムを実行せずに行う「静的テスト」とは別に、単体テストや結合テスト、システムテストなどプログラムを実際に実行しその動作を検証する「動的テスト」も存在します。それでは、これらの動的テストと静的テストにはどのような違いがありそれぞれどのような役割があるのでしょうか。

	静的テスト	動的テスト
実施方法	静的解析、レビュー（目視）など	単体テスト、結合テストなど
目的	ソフト品質の評価、問題の早期発見	
検証できること	動的テストでは実行されない、到達しにくいコード内の欠陥検出	実行時のソフトの振る舞いや仕様を満たしているか
	コードの保守性や可読性の評価	パフォーマンスや使用性
メリット	・プログラムが実行できない状態でも早期実施が可能 ・ツールで機械的に実施可能	実行時に発生するエラー検出やパフォーマンスの評価が可能
デメリット	指摘事項が膨大になることがある	・プログラムを動作できる状態にする必要があり、テスト実施までに時間がかかる ・ある程度スキルが必要

図 1 静的テストと動的テストの違い

1.3 静的解析の必要性

上述の通り、静的解析は他のテスト工程と同様に不具合を発見し、品質向上をすることが目的にありますが、中でも特に下記のような役割を持ちます。

・動的テストでは見つからない、潜在的な欠陥を検出できる

・ソースコードの可読性が高まり、修正や変更など保守開発も容易になる

・コーディングスタンダードを意識することで開発者のスキル向上にもつながる

静的解析は、一般的にコーディングが終わった後すぐに実行します。単体テストなどの動的なテストに比べてテストケース設計などの準備が少なく、ソースコードさえあれば実行ができる手軽さがあるため、品質向上の第一歩として有効な手段と言えます。さらに、動的テストも併せて実施することで、両者が相互に補完し合い、より効果的に品質向上を進めることができます。

▶ あるプロダクトで静的解析を実行すると、これだけの指摘が検出されます

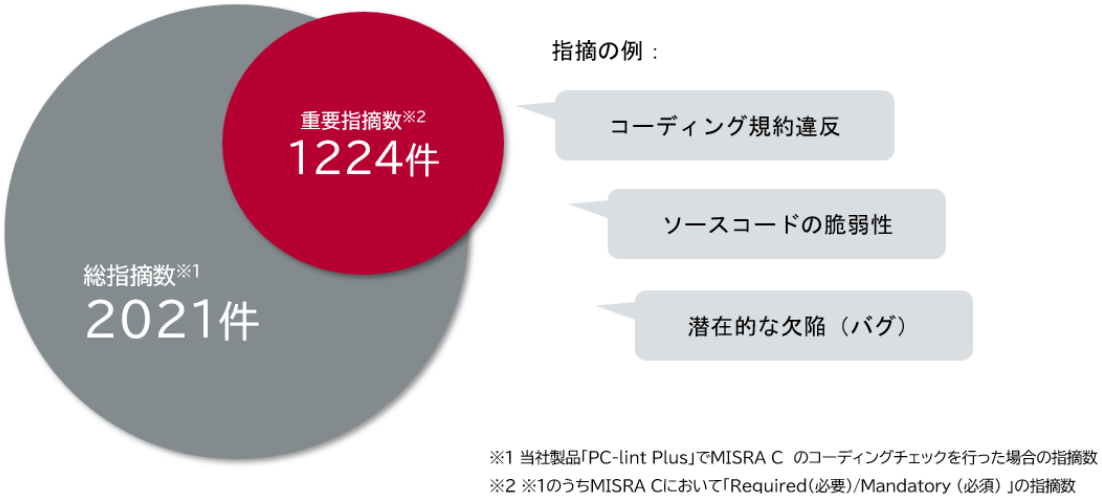


図 2 静的解析の必要性

2. PC-lint Plus とは

本章では静的解析ツール「PC-lint Plus」の機能や特長についてご紹介します。

2.1 PC-lint Plus の概要

PC-lint Plus は、C 言語、C＋＋言語のソースコードを解析してソフトウェアの問題を検出する静的解析ツールです。コマンドラインから実行できるため CI/CT への組み込みも容易で、統合開発環境（IDE）からも実行が可能です。解析結果は簡潔で分かりやすいレポートの形で出力されます。

また PC-lint Plus View を使えば、解析結果を GUI 上で扱えるようになります。結果の確認のみならず、さまざまなデータの集計、変更履歴のトラッキングなどグラフィカルな操作が可能です。
PC-lint Plus View の詳細については「5 PC-lint Plus View」から紹介しております。

ツールの信頼性においても、認証機関である Exida 社から産業（IEC 61508）、医療（IEC 62304）、自動車（ISO 26262）、の機能安全規格に則ったツール認証を取得しており、安心してご利用いただくことができます。

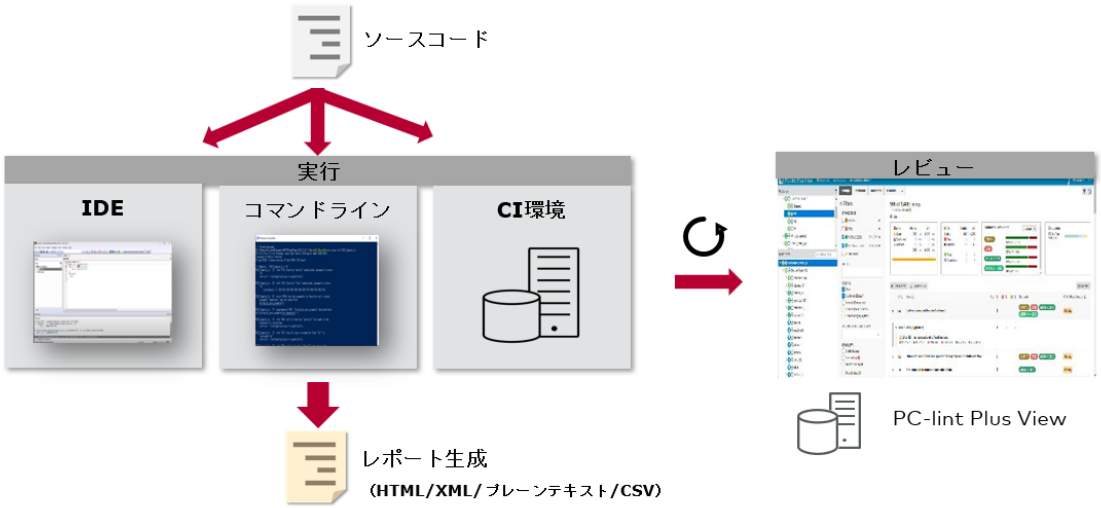


図 3 PC-lint Plus 概要

2.2 PC-lint Plus の仕組み

PC-lint Plus で静的解析を行う手順は以下の通りです。

①設定：ツールのインストールやコンパイラの設定など初期設定を行います

②実行：ソースコードに対してコマンドラインや統合開発環境から解析を実行します

③レポート生成：解析結果はレポートとして出力可能です

2.3 PC-lint Plus の静的解析

PC-lint Plus では主に下記のような解析をすることができます。

潜在的なバグ検出：

文法的なミス、未定義の関数や型の使用、暗黙のキャストによる型変換エラー、無限ループ、Null 等ポインタなどの問題が検出できます。

コーディングスタンダードチェック：

「MISRA C/MISRA C++」、「CERT C」、「AUTOSAR」、「CWE（共通脆弱性タイプ一覧）」のチェック機能を備えています。

スタックメモリ消費分析：

プログラム実行中のスタックメモリの使用量を調べることで、メモリ不足のエラーを防ぐ、またはリソースの節約、最適化に活用できます。

メトリクス計測：

ソースコードの質に着目し、プログラムの複雑さや保守の容易性などを評価します。ソースコードの行数や関数のコール数、循環的複雑度などを含む 100 以上のメトリクス計測が可能です。

2.4 レポート出力

解析結果はコマンドライン上や PC-lint Plus View で表示され、検出された行数や指摘レベル、指摘内容（コーディングスタンダードの規約違反など）が確認できます。また、レポートとして出力することもでき、HTML、XML、プレーンテキスト、CSV の形式に対応しています。

```
--- Module: manager.c (C)
ctypes.h  5  info 755:  global macro 'NUMBER_OF_TABLES' not referenced
#define NUMBER_OF_TABLES    6

manager.c  15  info 765:  external symbol 'Add_Included_Dessert' could be made static
void Add_Included_Dessert(struct order_type* Order)

manager.c  97  info 714:  external symbol 'Add_Party_To_Waiting_List' was defined but not referenced
void Add_Party_To_Waiting_List(char* Name)

manager.c  97  info 765:  external symbol 'Add_Party_To_Waiting_List' could be made static
void Add_Party_To_Waiting_List(char* Name)

ctypes.h  26  info 769:  global enumeration constant 'beverages::BEER' not referenced
enum beverages { NO_BEVERAGE, WINE, BEER, MIXED_DRINK, SODA };

ctypes.h  22  info 769:  global enumeration constant 'soups::CHOWDER' not referenced
enum soups     { NO_SOUP, ONION, CHOWDER };

manager.c  68  info 714:  external symbol 'Clear_Table' was defined but not referenced
int Clear_Table(table_index_type Table)

manager.c  68  info 765:  external symbol 'Clear_Table' could be made static
int Clear_Table(table_index_type Table)

ctypes.h  44  info 768:  global structure member 'table_data_type::Designator' not referenced
  char            Designator;
```

図 4-1 レポート例

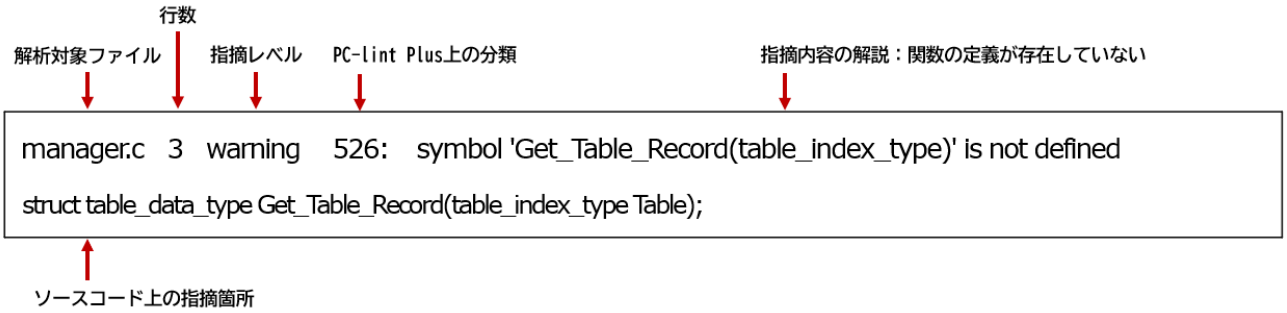


図 4-2 レポートの見方

指摘レベルは以下の 4 種類です。

ERROR:

構文の誤り、セマンティクス違反、処理エラーの可能性など、一般的に修正すべき箇所であることを指摘するメッセージ

WARNING:

プログラムに問題がある可能性があることを示す警告メッセージ

INFORMATIONAL:

一般的でない構造、一般的に良しとされていない慣行、および潜在的な落とし穴を指摘するメッセージ

ELECTIVE NOTES:

必ずしもソースコードの欠陥を表すわけではないが、特定のユーザーには関心点となる箇所を示すメッセージ

2.5 統合開発環境との連携

PC-lint Plus はいくつかの統合開発環境（IDE）と連携し、IDE から静的解析を実行することが可能です。

機能の詳細や使い方などは Online Manual もしくはベクター・ジャパンまでお問い合わせください（「8. 評価用ライセンスのご案内とお問い合わせ窓口」参照）。

対応環境（PC-lint Plus バージョン 2025）:

IAR Embedded Workbench, Keil µVision, MPLAB, Eclipse ベースの IDE, Visual Studio, Visual Studio Code, SEGGER IDE

3. PC-lint Plus を使用する前に

ここからは、実際に PC-lint Plus を動かしながら、その操作方法や機能について説明します。

本章では、PC-lint Plus を使用する際に必要なものについて説明します。巻末「8. 評価用ライセンスのご案内とお問い合わせ窓口」にある、マニュアル一式も併せてご確認ください。

3.1 PC-lint Plus を使用する際に必要なもの

PC-lint Plus を使用する際は、PC-lint Plus 本体の他に以下が必要です。

①コンパイラ:

お客様の開発環境で使用しているコンパイラが使えます。本稿では、チュートリアルとして gcc を使用します。

②Python パッケージ:

PC-lint Plus のパッケージに同梱している Python スクリプトを使うと、実行に必要なコンパイラのコンフィグファイルを自動生成することができます。この Python スクリプトを使用する場合 Python のインストールが必要です。

自動生成に対応していないコンパイラの場合は、手動でコンフィグファイルの作成が必要です。

3.2 解析対象のソースコードについて

本稿では、チュートリアルとして用意したサンプルのソースコードを使用します。ソースコードは巻末の参考資料に掲載しています。

3.3 インストール

インストール方法については、添付のアドバンストマニュアル「PC-lint Plus 使用方法」の「4.インストール」および「5.コンフィギュレーション」を参照ください。

4. 静的解析を実行する

本章では、実際に解析を実行する手順について記載します。

4.1 基本の解析

まずは基本となる解析コマンドを実行します。この基本コマンドにさまざまなオプションを追加することで、レポートへの出力やコーディングスタンダードチェックの機能などが有効になります。

コマンドプロンプトを開き、コンフィグファイルが配置されたディレクトリで下記コマンドを実行してください。

コマンドフォーマット：

pclp64.exe CONFIG.lnt ANALYZE-FILE
①　②

① コンフィグファイル

② 解析対象のソースファイル名

コマンド例：

pclp64.exe co-gcc.lnt manager.c

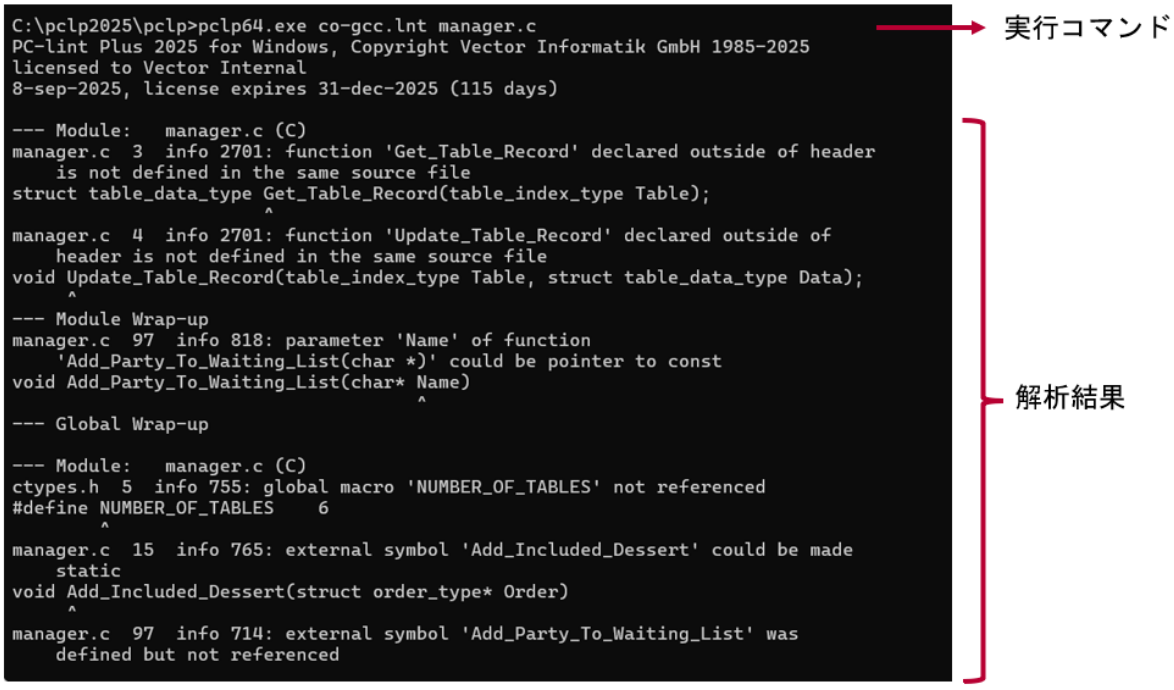


図 5 実行結果（コマンドライン）

4.2 レポートの出力

次に、結果をレポートとして出力するコマンドを追加して実行します。実行すると、指定した形式でのレポートが出力されます。

レポート出力コマンド:

-os(FILE)　　:レポートのファイル名

FORMAT.lnt　:レポートのファイル形式

コマンド例:

pclp64.exe -os(output.html) co-gcc.lnt lnt\env-html.lnt manager.c

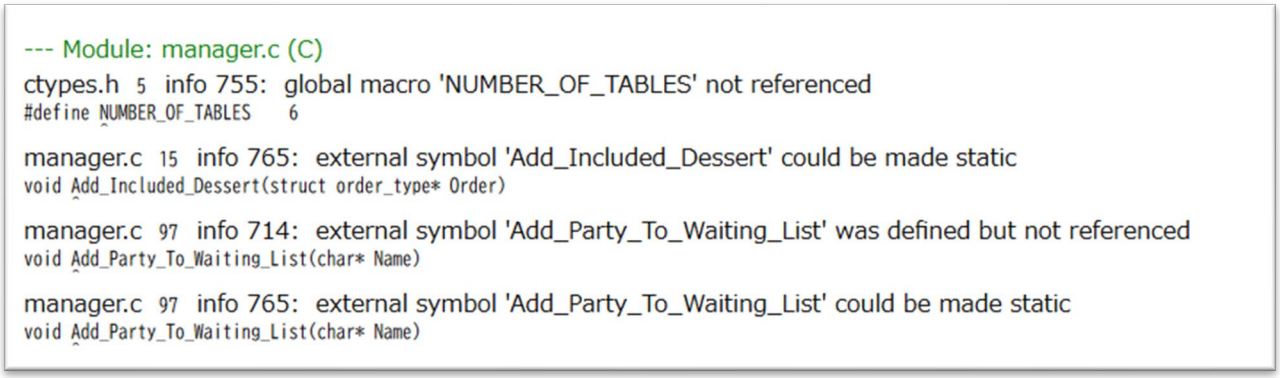


図 6　実行結果（レポート出力）

4.3 コーディングスタンダードチェック

コーディングスタンダードチェックを実行します。

PC-lint Plus 配布パッケージの lnt ディレクトリにある、各規格に対応する.lnt ファイルを実行コマンドに追加することで、コーディングスタンダードチェックをかけることが可能です。

PC-lint Plus では MISRA C/MISRAC++、CERT C、AUTOSAR、CWE（共通脆弱性タイプ一覧）のチェックに対応をしており、各規約の概要とコマンドオプションについて解説します。

4.3.1 MISRA C/MISRAC++

MISRA（Motor Industry Software Reliability Association）は、「安全で信頼性の高いソフトウェア」を目的に C および C++プログラミング・ガイドラインを作成しており、自動車業界のみならず医療、航空宇宙など　さまざまなミッションクリティカル分野の組み込み開発で採用されています。

PC-lint Plus は、以下のバージョンをサポートしています。（Ver. 2025 時点での情報です。最新はベクターまでお問い合わせください）。

MISRA サポートバージョン:

MISRA C :2004

MISRA C :2012（AMD-1、AMD-2、AMD-3、AMD-4 を含む）

MISRA C :2023

MISRA C :2025

MISRA C++

MISRA C++ :2023

MISRA のチェックは以下の.lnt ファイルを実行コマンドに追加します。

MISRA 実行コマンド:

MISRA C 2004 : au-misra2.lnt

MISRA C 2012 : au-misra3.lnt

MISRA C 2023 : au-misra4.lnt

MISRA C++ : au-misra-cpp.lnt

MISRA C++ 2023 : au-misra-cpp2.lnt

※MISRA C 2012 AMD1～4ごとの実行も可能ですが、本稿では省略しています

コマンド例:

pclp64.exe co-gcc.lnt au-misra3.lnt manager.c

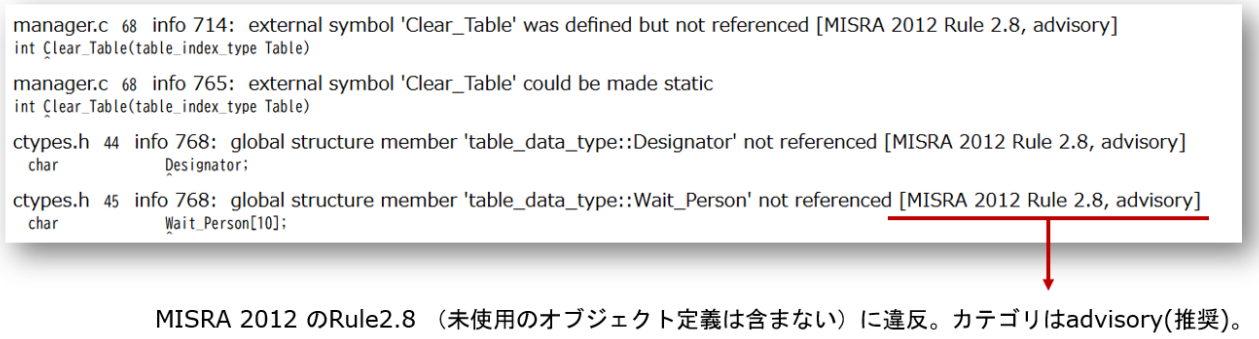


図 7 実行結果（MISRA）

4.3.2CERT C

CERT C はカーネギーメロン大学ソフトウェア工学研究所によって開発されたガイドラインで、プログラムの脆弱性を減らしセキュリティを強化することを目的としています。組み込み開発向けに多く採用され、ハードウェアを意識した規約も多い MISRA に対し、CERT C はサイバーセキュリティ対策など C/C++言語での開発に広く採用されます。

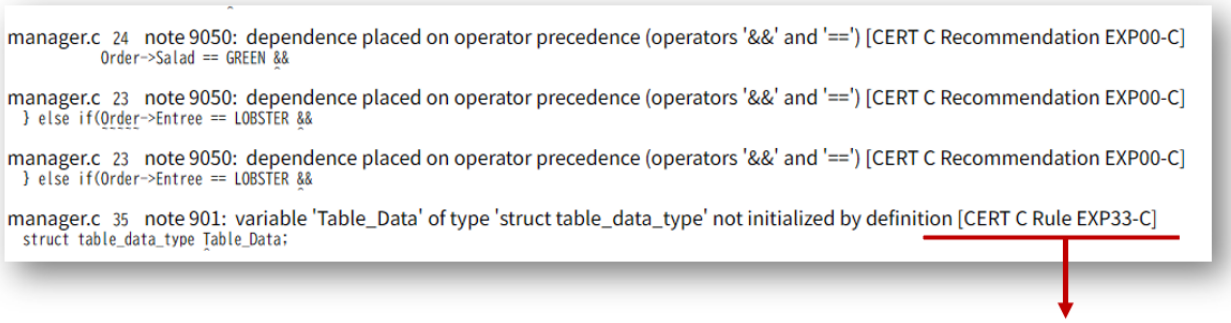
CERT C のチェックは以下の.lnt ファイルを実行コマンドに追加します。

CERT C 実行コマンド:

au-certc.lnt

コマンド例：

pclp64.exe co-gcc.lnt au-certc.lnt manager.c



Cert C のRule EXP33-C （初期化されていないメモリからの読み込みを行わない） を検出

図 8 実行結果（CERT C）

4.3.3CWE（共通脆弱性タイプ一覧）

CWE（共通脆弱性タイプ一覧）は、ソフトウェアのセキュリティ脆弱性や欠陥の種類を体系的に整理したリストです。CERT C と同様にセキュリティ強化を目的としており、実際、CERT C の規約の多くが CWE のリストに含まれます。しかし、CERT C が C/C++言語のコーディングを対象としているのに対し、CWE はソフトウェア開発全般における脆弱性を文書化しており、より汎用的なリストと言えます。

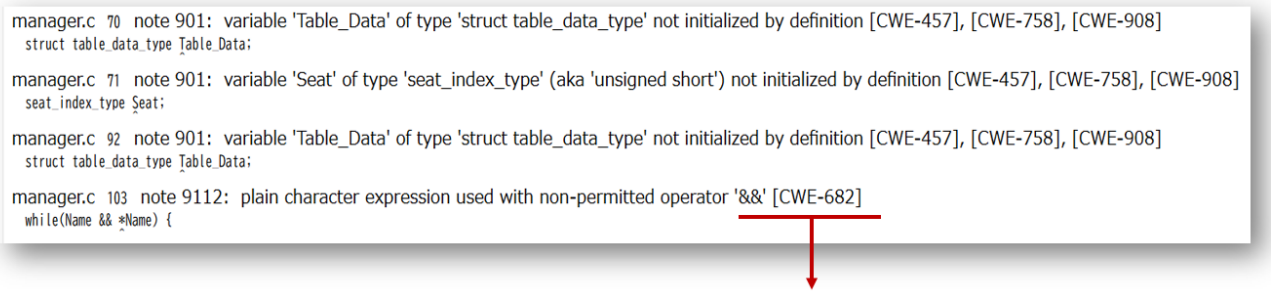
CWE のチェックは以下の.lnt ファイルを実行コマンドに追加します。

CWE 実行コマンド：

au-cwe.lnt

コマンド例：

pclp64.exe co-gcc.lnt au-cwe.lnt manager.c



CWE 682（意図しない計算結果になる可能性がある） を検出

図 9 実行結果（CWE）

4.3.4AUTOSAR C++14 コーディングガイドライン

AUTOSAR C++14 コーディングガイドラインは、安全関連のソフトウェア開発における C++14 の使用を目的とした MISRA C++ 2008 の更新版です。AUTOSAR ガイドラインの目的は、MISRA と同様、安全で信頼性の高いソフトウェアの開発を促進することにあります。

PC-lint Plus は、以下のバージョンをサポートしています。(Ver. 2025 時点での情報です。最新はベクターまでお問い合わせください。)

AUTOSAR サポートバージョン:

AUTOSAR 17-03

AUTOSAR 19-03

AUTOSAR のチェックは以下の.lnt ファイルを実行コマンドに追加します。

AUTOSAR 実行コマンド:

AUTOSAR 17-03 : au-autosar.lnt

AUTOSAR 19-03 : au-autosar19.lnt

コマンド例:

pclp64.exe co-gcc.lnt au-autosar.lnt manager.c

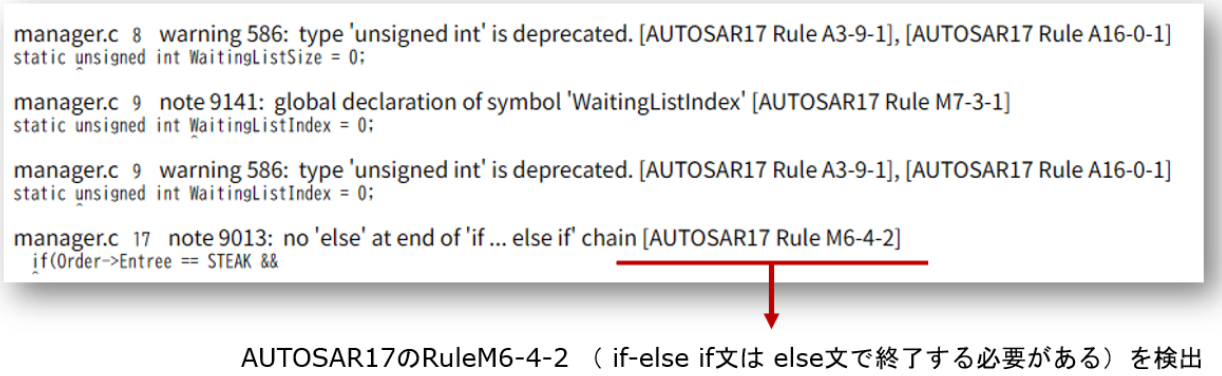


図 10 実行結果(AUTOSAR)

5. PC-lint Plus View

PC-lint Plus View は解析結果を分かりやすく可視化する最新のグラフィカルインターフェイスです（PC-lint Plus バージョン 2025 以降に実装）。GUI 上に解析結果を表示し、データの分析、結果のフィルタリング、変更履歴のトラッキング、ソースコードとの照合などが可能になり、チーム開発においても有効活用できます。本章では、PC-lint Plus View の特長および操作方法について紹介します。

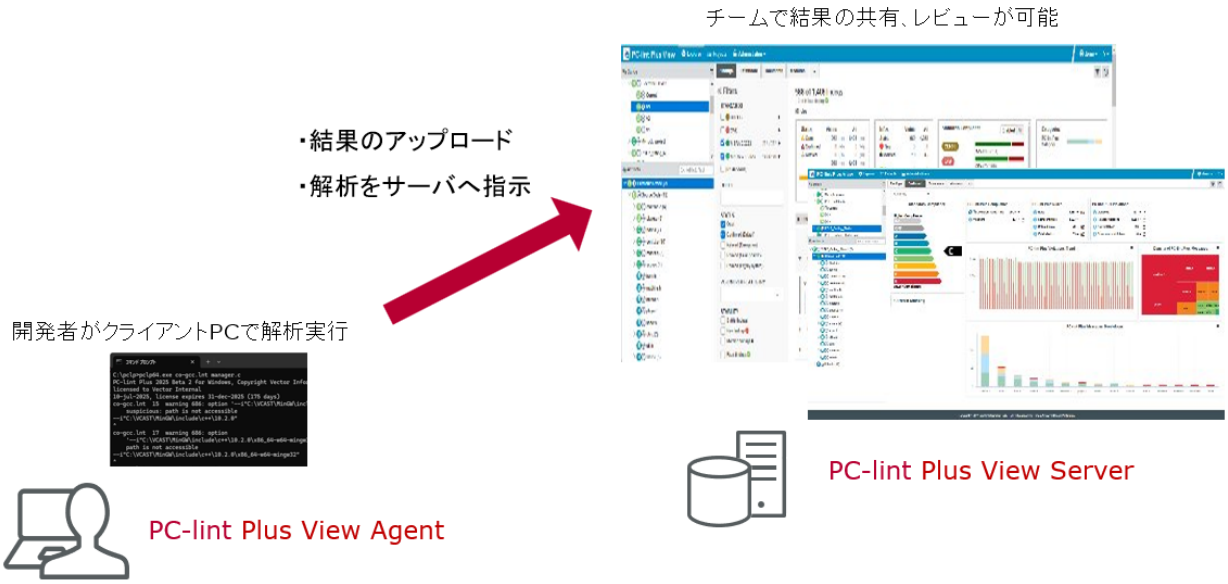


図 11　PC-lint Plus View 概要

5.1 PC-lint Plus View の特長

PC-lint Plus View では、単純な解析実行・結果確認の作業のみならず、さまざまなレベルのユーザーが活用できるような機能が搭載されています。

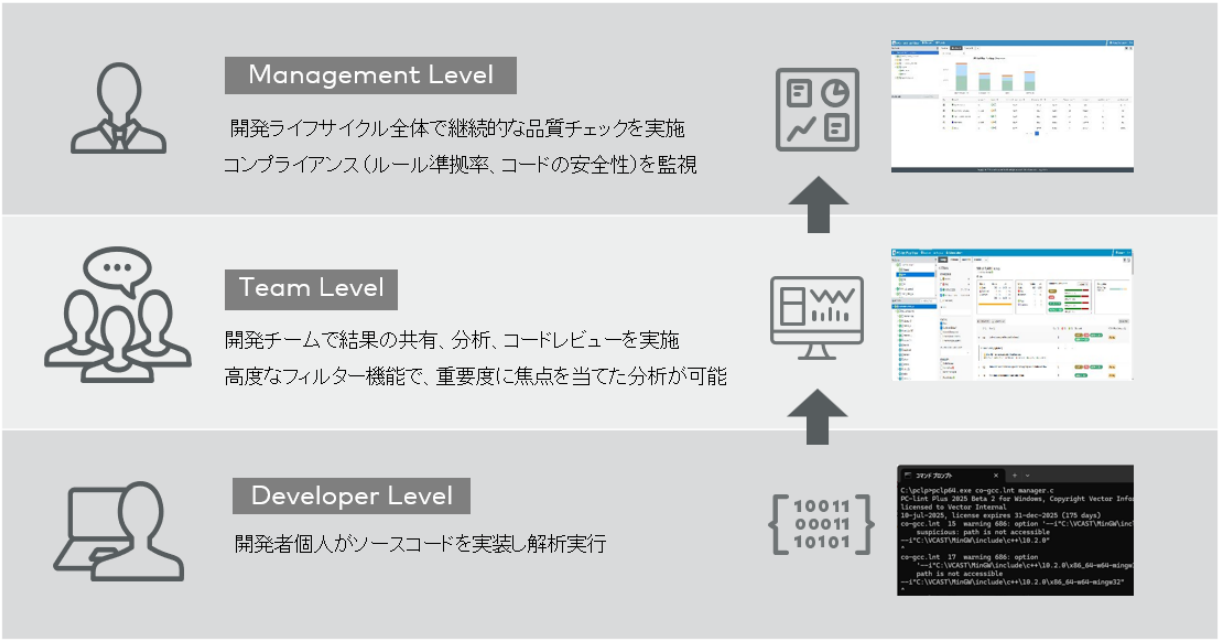


図 12　PC-lint Plus View の活用

・直感的で分かりやすいインターフェイス

PC-lint Plus View では解析結果に関連する情報すべてを 1 つに集約した、分かりやすく直観的なダッシュボードが用意されており、誰でも簡単に扱うことができます。得られた結果は重要度、ルール、メッセージ型、ファイルなどを基にフィルタリング可能です。

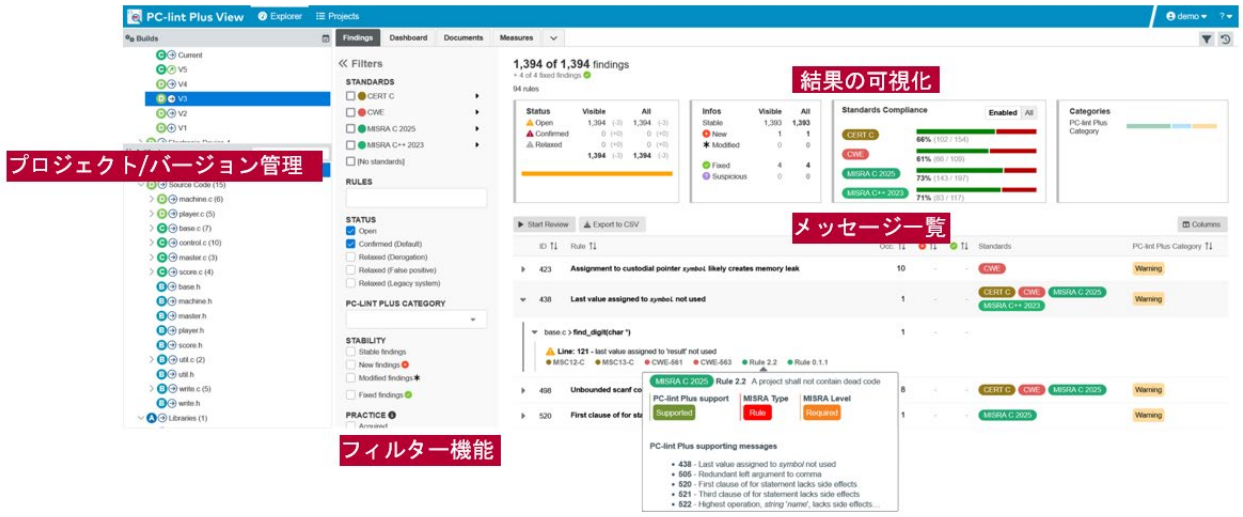


図 13　PC-lint Plus View ダッシュボード

・チーム開発での活用

PC-lint Plus View は、結果をアップする度に自動的にバージョンが管理され、過去の変更履歴の追跡や進捗状況のモニタリングが可能です。既に対応済みのメッセージや抑制されたメッセージも削除されることなく管理されているため、高い透明性を確保した静的解析が可能になり、開発ライフサイクル全体を通してツールが活用できます。

・レポート

PC-lint Plus View からもレポートの出力が可能です。「4.2　レポートの出力」でご紹介したコマンドラインからの実行で出力されるレポートは、メッセージ一覧の表示のみでしたが、PC-lint Plus View では、さまざまな項目でデータが整理され、エビデンスとしても効果的です。また、解析メッセージ毎にリンクも表示され、レポートから PC-lint Plus View 上の該当箇所へ直接アクセスすることが可能です。

出力形式は、PDF、CSV の他、SARIF 形式（Static Analysis Results Interchange Format）も対応しています。SARIF は、静的解析ツールの出力結果を標準化された形式で記述した JSON ベースのフォーマットです。複数の静的解析ツール間での共有や、対応する IDE、CI/CT パイプラインへの統合を容易に行うために有効です。

5.2 PC-lint Plus View を使用する際に必要なもの

ここからは、実際に PC-lint Plus View を動かしながら、その操作方法や機能について説明します。なお、本稿では WindowsPC を使用する場合のチュートリアルを紹介しております。Linux では一部内容が変わりますのでご注意ください。

まず、PC-lint Plus View を使用する際には以下が必要です。

① PC-lint Plus View サーバー
② PC-lint Plus View エージェント
③ Java ランタイム環境　バージョン 11 or 17
④ システム管理者権限を持つユーザーアカウント

5.3 PC-lint Plus View のインストール／設定

インストール／設定方法について説明します。

5.3.1 PC-lint Plus View サーバーのインストール

インストーラーの最新バージョンは、Vector Web サイトのダウンロードセンターからダウンロードできます。

・Vector Download Center

https://www.vector.com/int/en/support-downloads/download-center/

・PC-lint Plus View マニュアル 「Installing PC-lint Plus View」

https://pclpview.pclintplus.com/doc/user-guides/administration/installation.html

5.3.2 サーバーを起動する

以下のコマンドを実行し、サーバーを起動します。

コマンドフォーマット:

<PCLPVIEW_HOME>/bin/start.bat

コマンド例:

C: /pclp2025/pclpview/bin/start.bat

5.3.3 初回ログイン

サーバーが起動できたら、Web ブラウザに次のアドレスを入力してログインページにアクセスします。

PC-lint Plus View:

http://localhost:8280

初回アカウント(Username/Password):

PC-lint Plus View には、以下のユーザーが初期設定されており、ログインに使用できます。

- ✓ admin/admin
  システムを構成するために使用できる管理者アカウント
- ✓ demo/demo
  結果閲覧やレポート作成など一般機能が使用できるユーザーアカウント

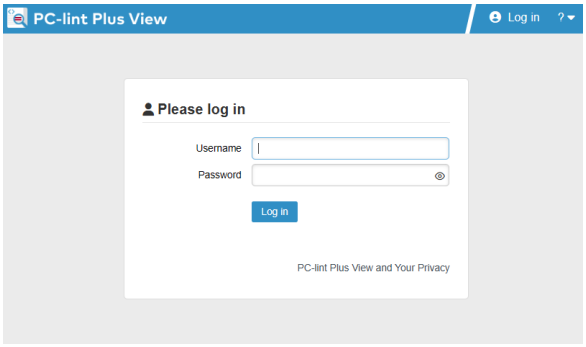


図 14　ログイン画面

5.3.4 アカウントを設定する

新規アカウントを作成する場合は、管理者アカウントでログインし、画面中央上部「Administration」→「Users」→「Add a new user」の順でアカウントの追加が可能です。

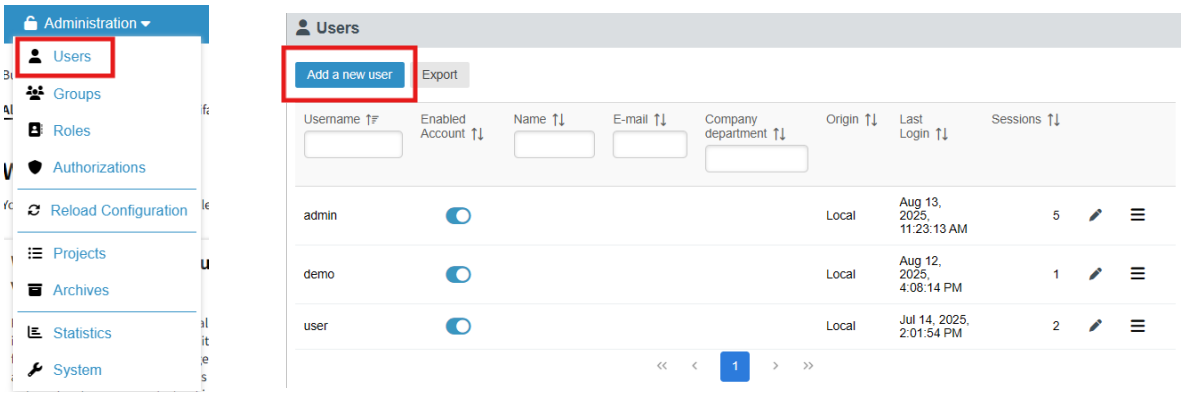


図 15　アカウントの設定

5.3.5 PC-lint Plus View Agentのインストール

PC-lint Plus View Agent は、コード解析を実行するクライアント PC 側にダウンロードするものです。

PC-lint Plus View Agent には主に 2 つの役割があります。

1. クライアント PC 上でコード解析し、その結果をサーバーにアップロードする。
2. コードやその他の入力データの解析を実行するようにサーバーに指示する。

PC-lint Plus View Agent をダウンロードするには、画面右上の「？（ヘルプ）」から「PC-lint Plus View Agent」にアクセスします。

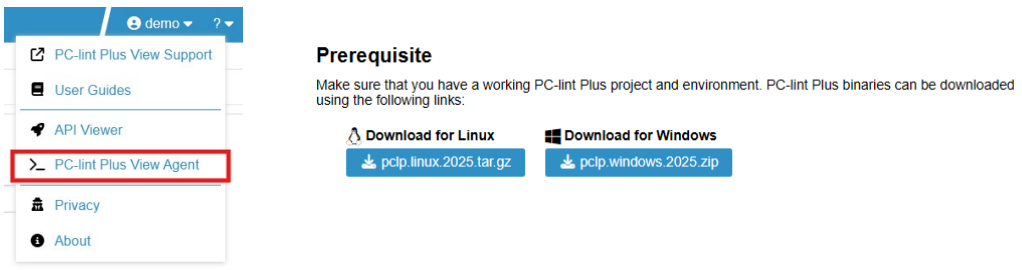


図 16　PC-lint Plus Agent のインストール

5.4 プロジェクトを作成する

5.4.1 作成準備

PC-lint Plus View でプロジェクトを作成するには、demo ユーザーでログインし、「Projects」→「 Create Project」を選択し、プロジェクト名やバージョン名など一般情報の設定をします。

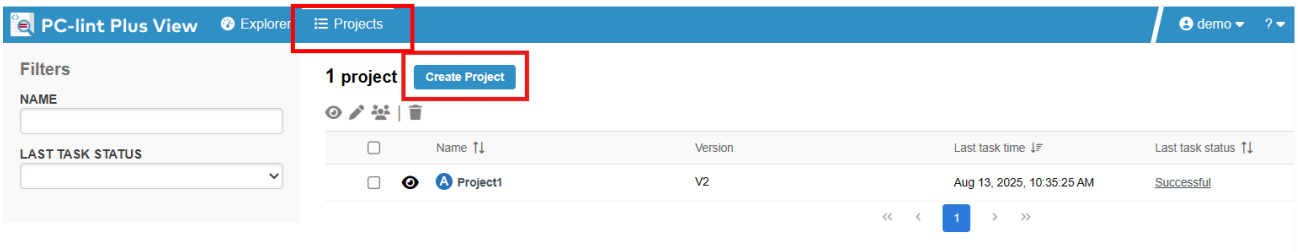


図 17-1 プロジェクトの作成

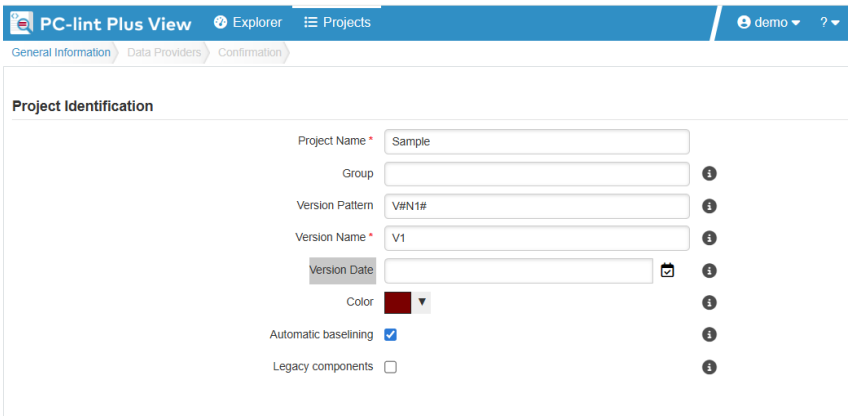


図 17-2 プロジェクトの作成 - 一般情報の設定

✓ Automatic baselining 設定

PC-lint Plus View では、作成されたプロジェクトのすべてのバージョンを保持することが可能ですが、特定のマイルストーンの解析結果のみを永続的に保持したい場合に、ベースラインバージョンとドラフトバージョンの使い分けが可能です。

「Automatic baselining」ボックスをオンにすると、ベースラインバージョンで作成され、以降すべてのバージョンはデフォルトでベースラインバージョンになります。解析にエラーが発生した場合は、ドラフトバージョンが作成されます

「Automatic baselining」ボックスをオフにすると、ドラフトバージョンが作成され、以降すべてのバージョンはデフォルトでドラフトバージョンになります。

次のページでは、ソースコードが格納されているパス、および JSON ファイルが含まれるパスを指定します。

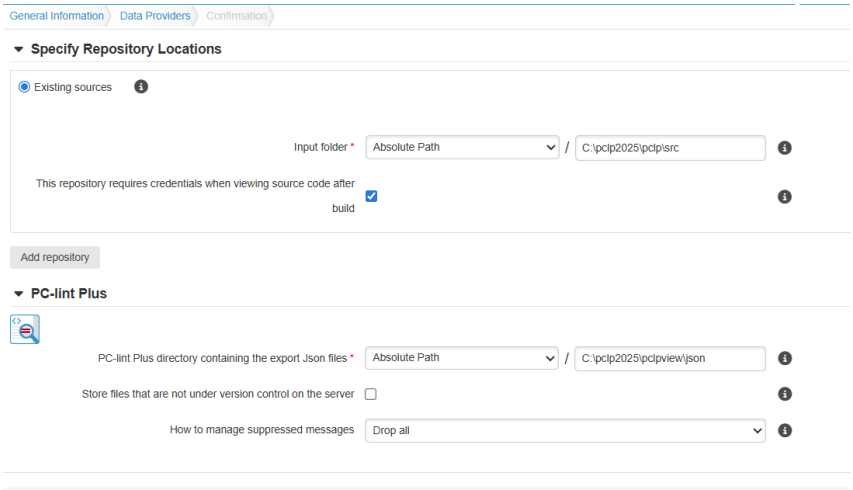


図 17-3 プロジェクトの作成 – パスの指定

設定が完了したら、実行に必要なコマンドが表示されます。

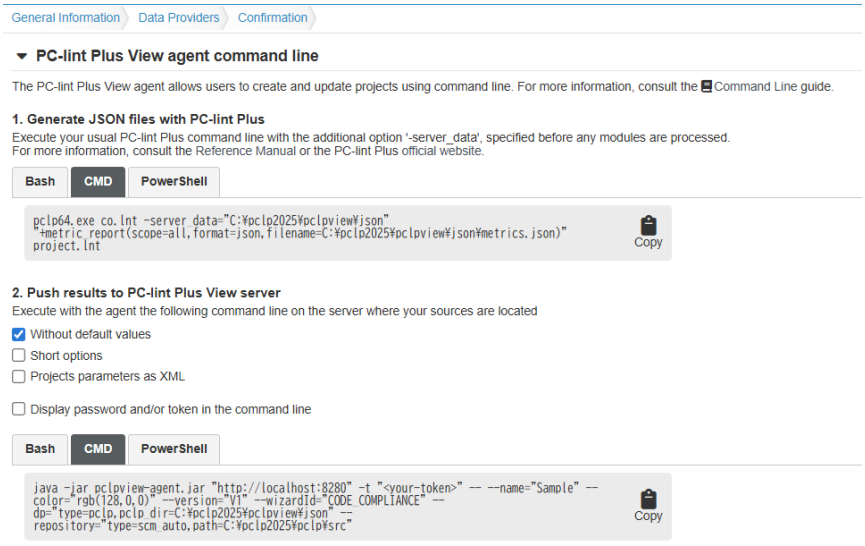


図 17-4 プロジェクトの作成 – 実行コマンドの表示

5.4.2 PC-lint Plus で JSON ファイルを生成する

JSON ファイルには、PC-lint Plus View サーバーに結果をインポートするために必要なデータが含まれます。前項のプロジェクト作成画面で生成された「1. Generate JSON files with PC-lint Plus」コマンドをコマンドラインから実行します。

注意：コマンド末尾「project.lnt」では解析対象となるソースファイルなどを指定します。例「manager.c」

5.4.3 サーバーへ結果をプッシュする

PC-lint Plus の結果は、コマンドラインから PC-lint Plus View エージェントを使用して PC-lint Plus View サーバーにプッシュされ、サーバー上でプロジェクトを作成および更新できるようになります。

前項のプロジェクト作成画面で生成された「2. Push results to PC-lint Plus View server」コマンドをコマンドラインから実行します。

注意：＜your-token＞"には下記方法で取得したトークンを入力する必要があります。

✓ トークンの取得

クライアント PC から PC-lint Plus View のデータにアクセスするときは、トークンにより認証が行われる仕組みになっています。

トークンは「Account Settings」→「Tokens」タブを選択し、任意のトークン名を入力後、「Add」ボタンで取得できます。ここで取得したトークンは有効期限が無いため、永続的に有効になります。無効にしたい場合は、同じ画面から対象のトークンに対して「Revoke」を押下します。

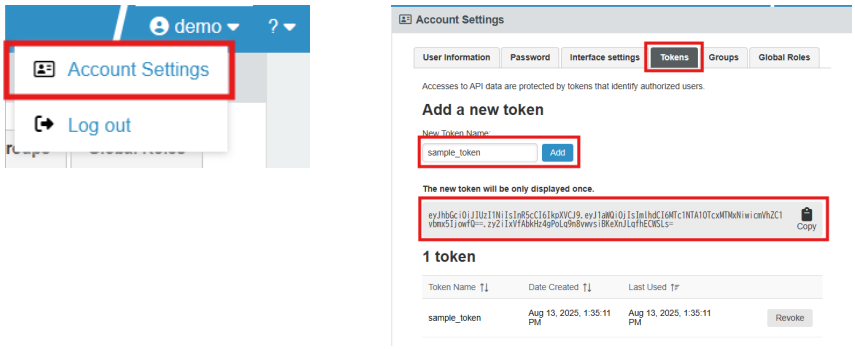


図 18 トークンの取得

一時的に有効なトークンを取得することも可能です。方法はマニュアルの下記ページを参照ください。

・PC-lint Plus View マニュアル 「Authentication and Security」

https://pclpview.pclintplus.com/doc/user-guides/api/security.html#sect_temporary_token

✓ オプション

PC-lint Plus View では解析結果とともにソースコードを表示させることが可能です。以下のコマンドオプションを JSON ファイルパスの後に追加します。

コマンドオプション：

store_files=true

例：

--dp"type=pclp, pclp_dir=<*JSON ファイルパス*>, **store_files=true**"

```
java -jar pclpview-agent.jar -t
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOjIsImlhdCI6MTc1NTA0ODg5MzQ3NSwicm
VhZC1vbmx5IjowfQ==.F6GLY1Y4DKZYfUvkRpJ8HczsEzelS4G2y1l8itOp3WU=
http://localhost:8280 -- --name="Project1" --
dp="type=pclp,pclp_dir=C:¥pclp2025¥pclpview¥json, store_files=true" --
repository="type=scm_auto,path=C:¥pclp2025¥pclp¥src"
```

コマンド実行後、PC-lint Plus View サーバー（http://localhost:8280）に接続し、プロジェクトが作成されていることを確認します。

図 19 プロジェクト作成結果

5.5 PC-lint Plus View の主な機能

作成されたプロジェクトにアクセスすると、解析結果の画面が表示されます。ここでは、画面の見方や主な機能について説明します。

5.5.1 PC-lint Plus View の見方

PC-lint Plus View は合計で6つのタブから構成されており、メインとなる Findings 以外はエクスプローラー設定から表示／非表示のカスタマイズが可能です。

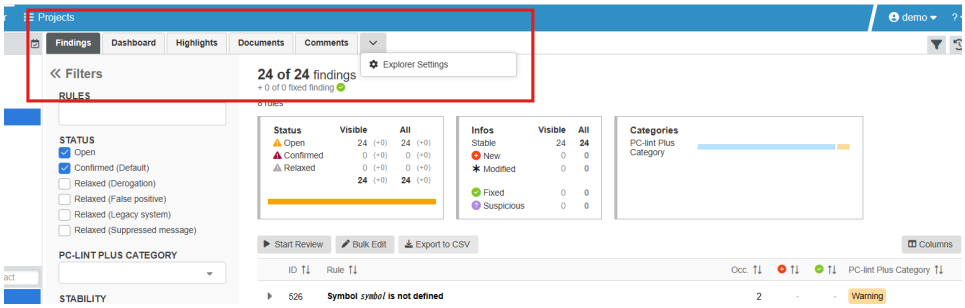


図 20 エクスプローラー設定

Findings：

メッセージ一覧の確認やフィルタリングによる検索などが可能なメイン画面です。

Dashboard：

ルールの遵守率（コンプライアンス）や過去バージョンからの傾向の変化などさまざまな分析データを見ることができます。

Highlights：

ソースファイルごとに集計されたデータが表示されます。

Documents：

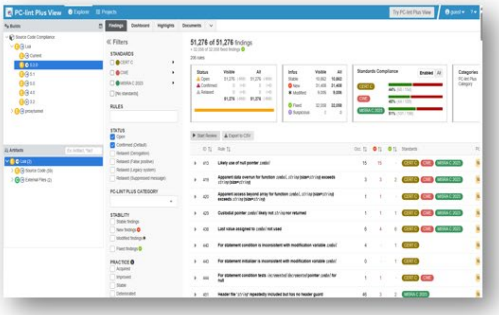
レポート出力ができます。

Measures：

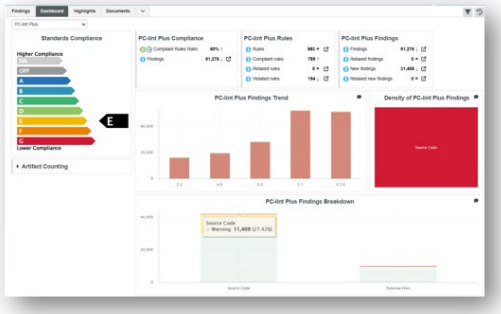
コーディングスタンダード別や指摘レベル別など、カテゴリごとのデータが確認できます。

Comments：

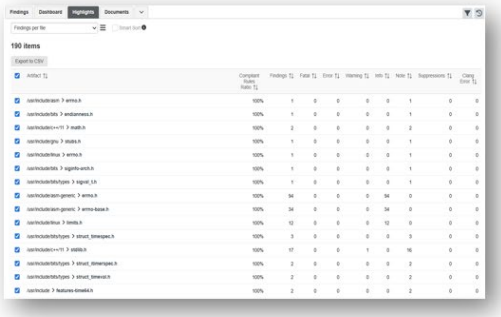
メッセージごとにコメントを挿入することが可能です。Comments タブにはコメントを挿入したメッセージがまとめて表示されます。



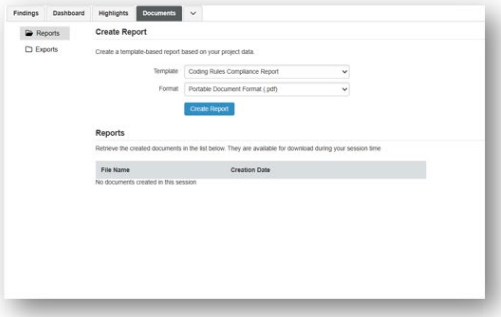
Findings



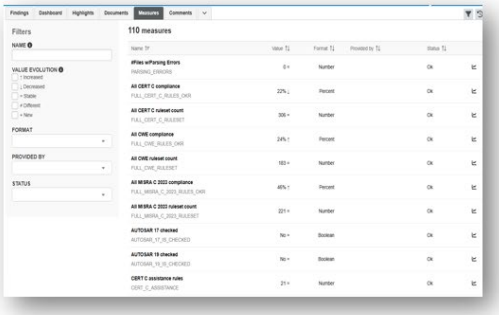
Dashboard



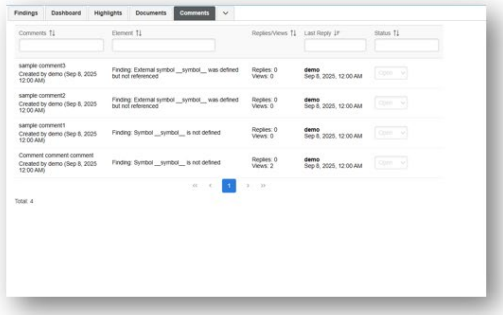
Highlight



Documents



Measures



Comments

図 21 PC-lint Plus View 画面

5.5.2 結果のレビュー

Findings ページで「Start Review」をクリックすると、解析結果を1メッセージずつ順に見ていくことができます。ここにはメッセージ内容や該当するコーディングスタンダードなど、必要な情報がすべて集約されており、コメントを残すことも可能です。コーディングスタンダードについては、直接外部リンクに移動し、規約の詳細を確認することもできます。

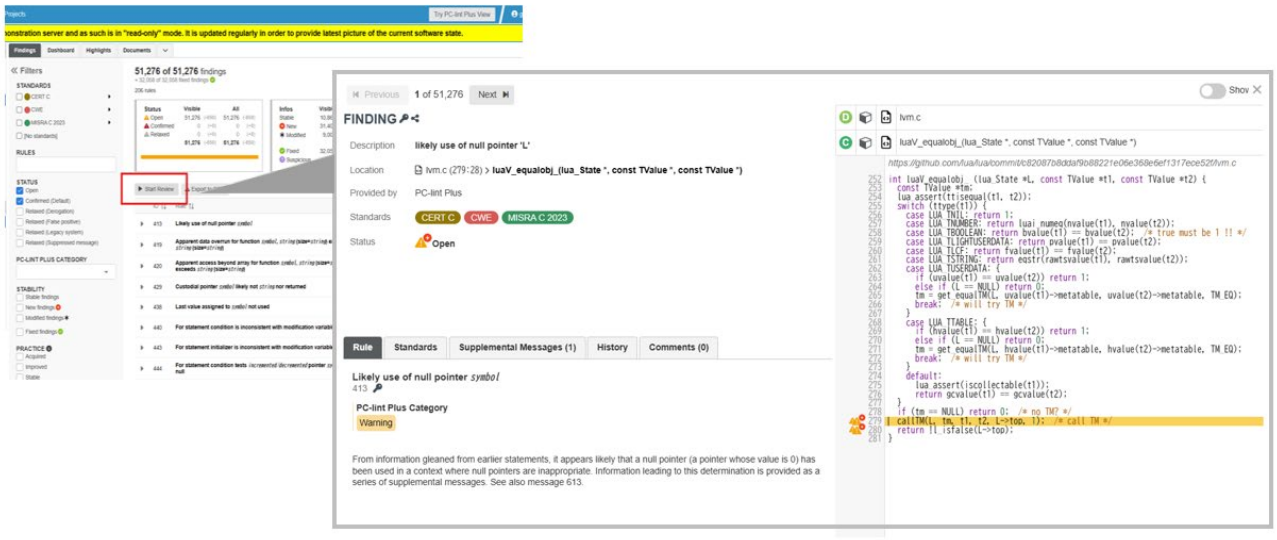


図 22　解析結果のレビュー

また、指摘箇所をソースコード上でも参照することができます。

さらに、ソースコード右上「Compare with」から、過去のバージョンを選択すると、選択したバージョンと現在のバージョンの差分を比較し、新しく追加された指摘や修正対応された箇所などの確認が可能です。（「5.4.3 サーバーへ結果をプッシュする」のオプション設定（p.22）が必要です。）

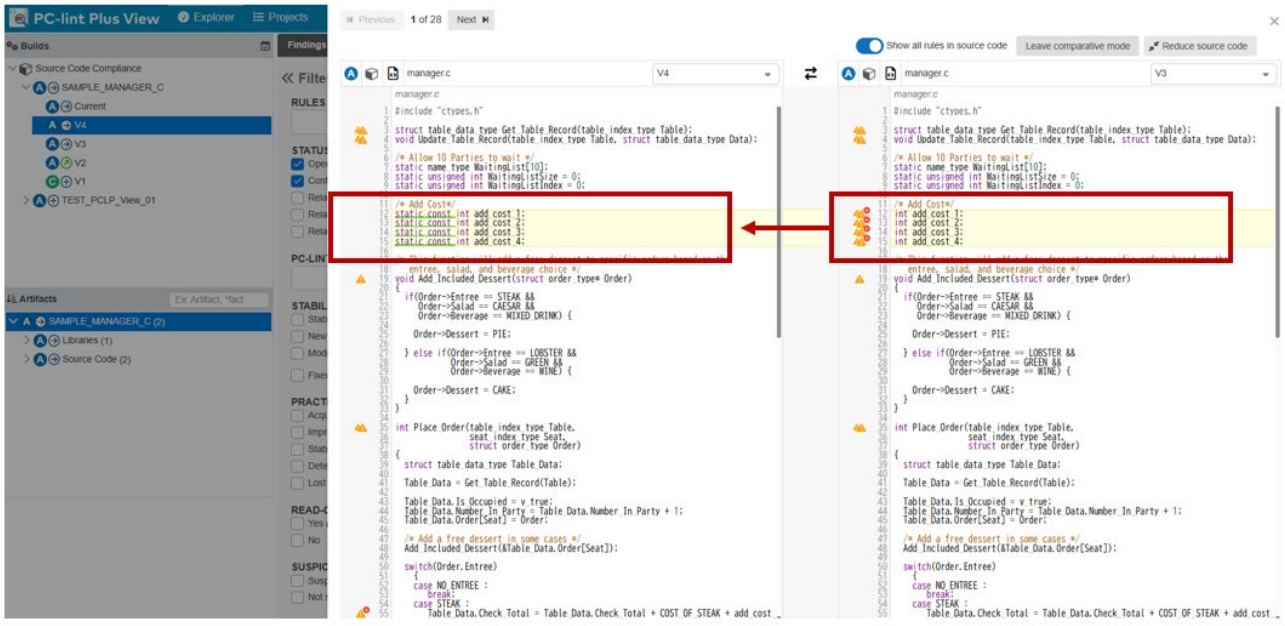


図 23　ソースコード変更比較

5.5.3 レポートの出力

レポートの出力は Documents タブから行います。PC-lint Plus View からは、以下の形式でレポート出力が可能です。

・PDF

「Reports」から「Format」が.pdf になっていることを確認し「Create Report」で可能です。

・EXCEL

「Exports」から「Name」欄にて「PC-lint Plus Findings report to Excel」を選択します。

・SARIF

「Exports」から「Name」欄にて「Findings to a JSON file in SARIF format」を選択します。

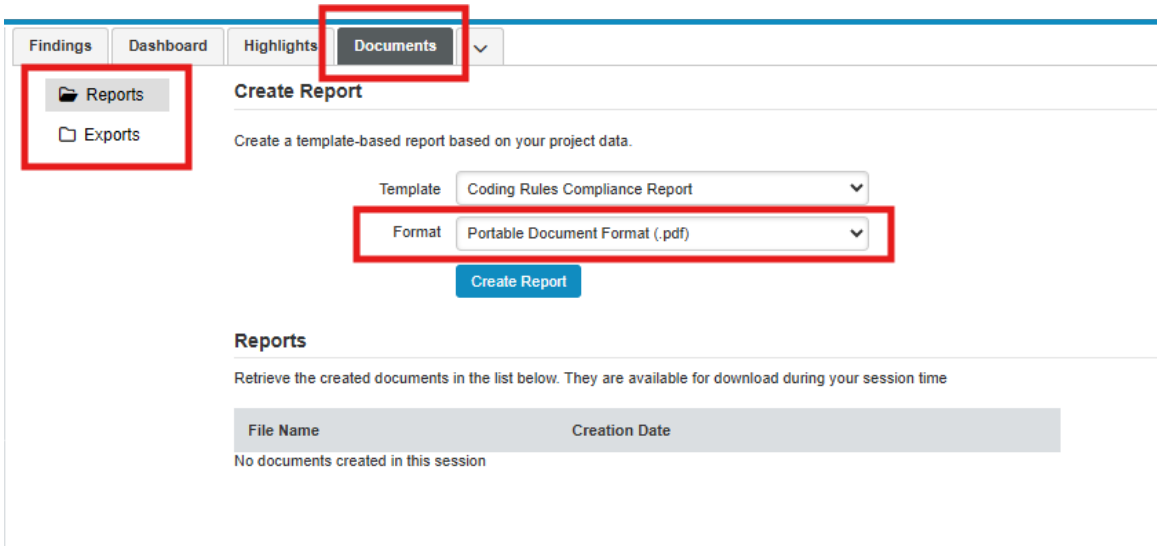


図 24 レポートの出力方法（PDF の場合）

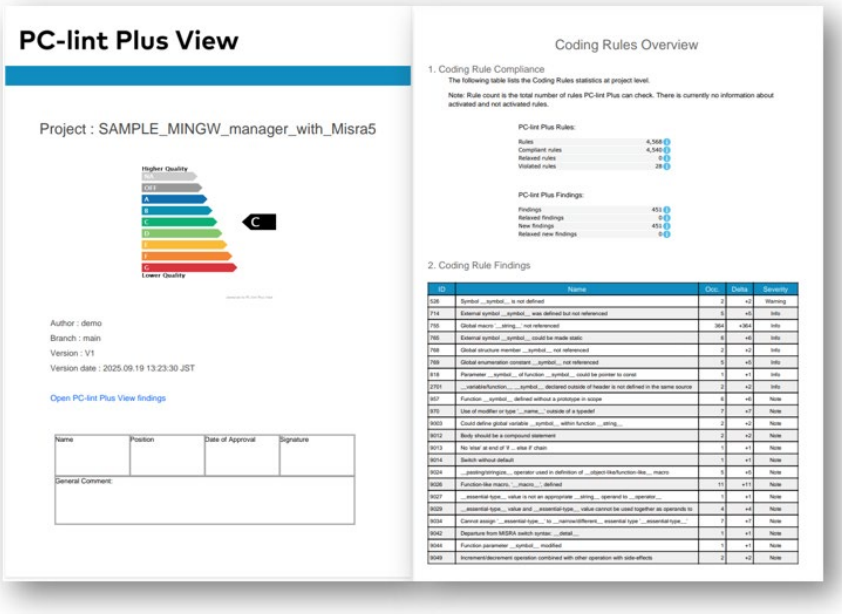


図 25 レポート例（PDF の場合）

6. あとがき

本稿『はじめての PC-lint Plus』では、静的解析の基礎や PC-lint Plus の基本的な機能についてご説明しました。静的解析は一見単純な作業に思えますが、ツールを適切に使いこなすことで、ソフトウェアの品質向上と開発コストの削減に非常に大きく貢献します。PC-lint Plus は、セキュアで高速な静的解析を実現します。

ツールに関するご相談、お問い合わせは、お気軽にベクター・ジャパンまでお寄せください。

以上、本稿が皆様の開発業務の一助となれば幸いです。

- ✓ **PC-lint Plus　製品ホームページ**

  https://www.vector.com/jp/ja/products/products-a-z/software/pc-lint-plus/

- ✓ **ホワイトペーパー「複雑化と規模拡大に悩む組み込みソフト業界での静的コード解析ツールの新たな選択肢（PC-lint Plus）」**

  https://www.vector.com/jp/ja/download/wp-tf-pc-lint-plus-1/

- ✓ **はじめてシリーズ「はじめての VectorCAST」**

  ベクターの単体テストツール「VectorCAST」について紹介しています

  https://www.vector.com/jp/ja/know-how/vj-beginners/vectorcast/

7. 参考資料

本稿でチュートリアルとして使用したサンプルコードを掲載します。

プログラム内容:

レストランでの入店待ち、注文状況に応じた価格決定などを行うプログラム。

関数:

① Add_Party_To_Waiting_List

関数定義:void Add_Party_To_Waiting_List(char* Name)

機能概要:入店待ちのお客様が入力した氏名を、待機リストの配列に追加する

② Add_Included_Dessert

関数定義:void Add_Included_Dessert(struct order_type* Order)

機能概要:お客様が注文したメニューに応じて、デザートを自動追加する

③ Place_Order

関数定義:int Place_Order(table_index_type Table, seat_index_type Seat,struct order_type Order)

機能概要:お客様が注文したメイン料理の金額を、テーブルの合計金額に加算する

④ Clear_Table

関数定義:int Clear_Table(table_index_type Table)

機能概要:テーブルの注文履歴、合計金額などを初期化する

```
#include "ctypes.h"

struct table_data_type Get_Table_Record(table_index_type Table);
void Update_Table_Record(table_index_type Table, struct table_data_type Data);

/* Allow 10 Parties to wait */
static name_type WaitingList[10];
static unsigned int WaitingListSize = 0;
static unsigned int WaitingListIndex = 0;

/* This function will add a free dessert to specific orders based on the
   entree, salad, and beverage choice */
void Add_Included_Dessert(struct order_type* Order)
{
  if(Order->Entree == STEAK &&
     Order->Salad == CAESAR &&
     Order->Beverage == MIXED_DRINK) {

    Order->Dessert = PIE;

  } else if(Order->Entree == LOBSTER &&
            Order->Salad == GREEN &&
            Order->Beverage == WINE) {

    Order->Dessert = CAKE;
  }
}

int Place_Order(table_index_type Table,
                seat_index_type Seat,
                struct order_type Order)
{
  struct table_data_type Table_Data;

  Table_Data = Get_Table_Record(Table);

  Table_Data.Is_Occupied = v_true;
  Table_Data.Number_In_Party = Table_Data.Number_In_Party + 1;
  Table_Data.Order[Seat] = Order;
```

```
  /* Add a free dessert in some cases */
  Add_Included_Dessert(&Table_Data.Order[Seat]);

  switch(Order.Entree)
   {
   case NO_ENTREE :
     break;
   case STEAK :
     Table_Data.Check_Total = Table_Data.Check_Total + COST_OF_STEAK;
     break;
   case CHICKEN :
     Table_Data.Check_Total = Table_Data.Check_Total + COST_OF_CHICKEN;
     break;
   case LOBSTER :
     Table_Data.Check_Total = Table_Data.Check_Total + COST_OF_LOBSTER;
     break;
   case PASTA :
     Table_Data.Check_Total = Table_Data.Check_Total + COST_OF_PASTA;
     break;
   }

  Update_Table_Record(Table, Table_Data);
  return 0;
}

int Clear_Table(table_index_type Table)
{
  struct table_data_type Table_Data;
  seat_index_type Seat;
  Table_Data = Get_Table_Record(Table);
  Table_Data.Is_Occupied = v_false;
  Table_Data.Number_In_Party = 1;

  for (Seat=0; Seat < SEATS_AT_ONE_TABLE; Seat++){
     Table_Data.Order[Seat].Soup     = NO_SOUP;
     Table_Data.Order[Seat].Salad    = NO_SALAD;
     Table_Data.Order[Seat].Entree   = NO_ENTREE;
     Table_Data.Order[Seat].Dessert  = NO_DESSERT;
```

```
      Table_Data.Order[Seat].Beverage = NO_BEVERAGE;
                        }

  Table_Data.Check_Total = 0;

  Update_Table_Record(Table, Table_Data);
  return 0;
}

FLOAT Get_Check_Total(table_index_type Table)
{
  struct table_data_type Table_Data;
  Table_Data = Get_Table_Record(Table);
  return (Table_Data.Check_Total);
}

void Add_Party_To_Waiting_List(char* Name)
{
  int i = 0;
  if(WaitingListSize > 9)
    WaitingListSize = 0;

  while(Name && *Name) {
    WaitingList[WaitingListSize][i++] = *Name;
    Name++;
  }
  WaitingList[WaitingListSize++][i] = 0;
}

char* Get_Next_Party_To_Be_Seated(void)
{
  if(WaitingListIndex > 9)
    WaitingListIndex = 0;
  return WaitingList[WaitingListIndex++];
}
```

8. 評価用ライセンスのご案内とお問い合わせ窓口

8.1 評価用ライセンス

下記リンクより PC-lint Plus 評価用ライセンスのダウンロードが可能です。リクエストを送信するとすぐにダウンロードが始まり、14 日間の無償トライアル期間が開始しますので、ご注意ください。

- ✓ 貸出依頼フォーム
  https://vector-softwarequality.com/static-code-analysis/pc-lint-plus/trial
- ✓ PC-lint Plus Online Manual （日本語はブラウザの翻訳機能をお使いください）
  https://help.vector.com/pclintplus/index.html
- ✓ 日本語版アドバンストマニュアル「PC-lint Plus の使用方法」
  https://portal.vector.com/shared/dc8ca4d2-acc5-481d-883d-cd6180ca24c8

評価用ライセンス利用時はサポートさせていただきますので、下記までご連絡ください。

- ✓ ベクター・ジャパン 営業部
  TEL：（東京）03-4586-1808　E-mail：sales@jp.vector.com

8.2 お問い合わせ窓口

技術関連のお問い合わせは、ベクターサポートまでお寄せください。

- ✓ サポートリクエスト（Web サイト経由のサポート問い合わせフォーム）
  www.vector.com/jp/ja/support-downloads/support/

【お問い合わせ時のお願い】

お問い合わせの際、サポート開始前に製品のバージョン（例：PC-lint Plus バージョン xx.x など）、シリアルナンバーなどをお伺いいたします。お手数ではございますが、事前にご確認のうえ、お問い合わせくださいますようお願い申し上げます。

【サポート対応について】

夜間／休日に頂いたお問い合わせは翌営業日の対応となります。また、内容によりご回答までにお時間を頂く場合がございます。なお、不具合が発生した際にご使用されていた関連プログラム一式のコピーを、検証のためご提供いただくようお願いする場合がございます。何卒ご理解ご協力のほど、よろしくお願い申し上げます。

- ✓ **お見積もり／保守契約関連のお問い合わせ**
  営業部　TEL：（東京）03-4586-1808　E-mail：sales@jp.vector.com

VECTOR >

MORE INFORMATION
Visit our website for:

> News
> Products
> Demo software
> Support
> Training and workshops
> Contact addresses

vector.com