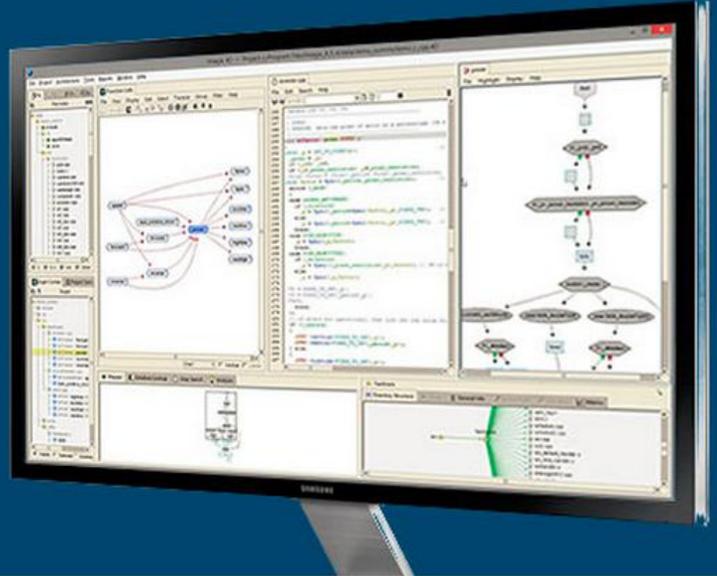


复杂的软件也能变得容易理解

逆向工程源代码分析

Imagix 4D 控制流和依赖关系分析。提高代码质量，降低软件风险。



**Imagix 4D**源代码质量分析工具

**Imagix** 



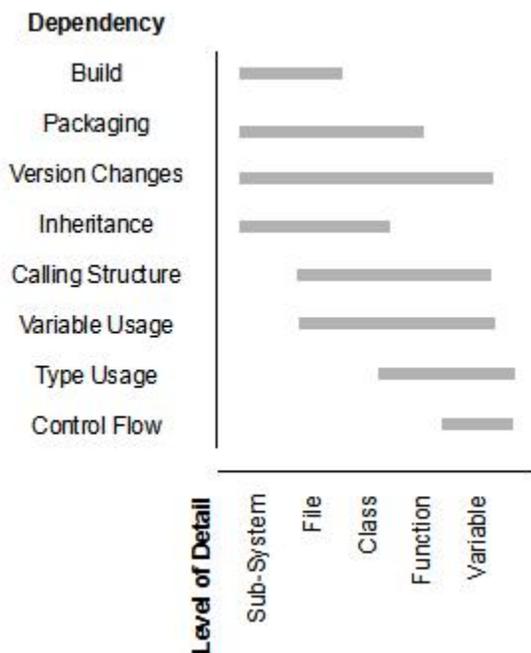
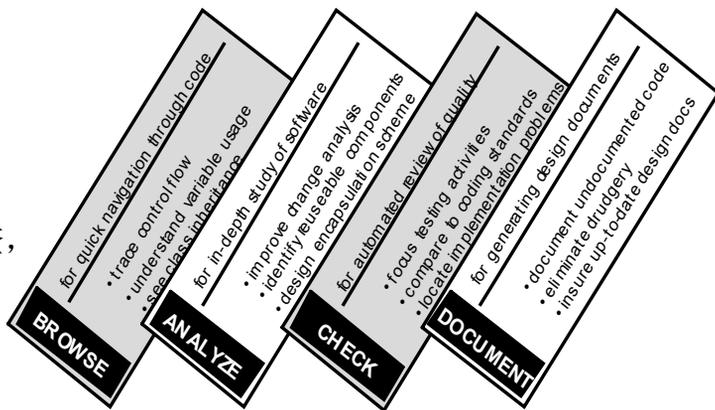
**Imagix 4D** 帮助开发人员了解复杂的，遗留的或开源的C，C++和Java源代码。通过使用Imagix 4D逆向工程和分析软件。加快开发，测试和维护的速度，学习不熟悉的代码，消除由于错误理解导致的bug。

**Imagix 4D**成立于1994年，创始人之前曾在Cadre Technologies， Engineering Animation， Intel， Siemens和Tektronix担任工程，营销和总经理职务。

商业客户包括戴尔，通用，惠普，日立，英特尔，三菱，日产，雷诺和三星  
 美国政府的客户包括FAA， JPL， 海军研究实验室， NASA和桑迪亚国家实验室。

### 工具特性：

- 支持**CERT-C** 和 **CERT-C++**标准
- 支持**MISRA C/C++**代码规范
- 使用可视化来实现快速直观的程序理解
- 提供广泛的视角，生成指标和自动化检查，以发现设计和编码中的问题
- 自动生成综合设计报告文件



- 从高级架构到详细构建，类和功能相关性的细节
- 不需要额外的工具来检查具体问题
- 在上下文中显示软件量度，以定位重点测试和维护工作
- 识别设计和编码标准的例外情况
- 自动分析嵌入式软件中存在问题的结构性问题源代码



**CWE™**通用缺陷列表（Common Weakness Enumeration）是已经在电脑软件中发现缺陷的通用在线词典。

**Imagix 4D**完全支持CWE标准，有助于促进工具的有效使用，识别、发现并解决电脑软件中的bug、缺陷和易受攻击点。

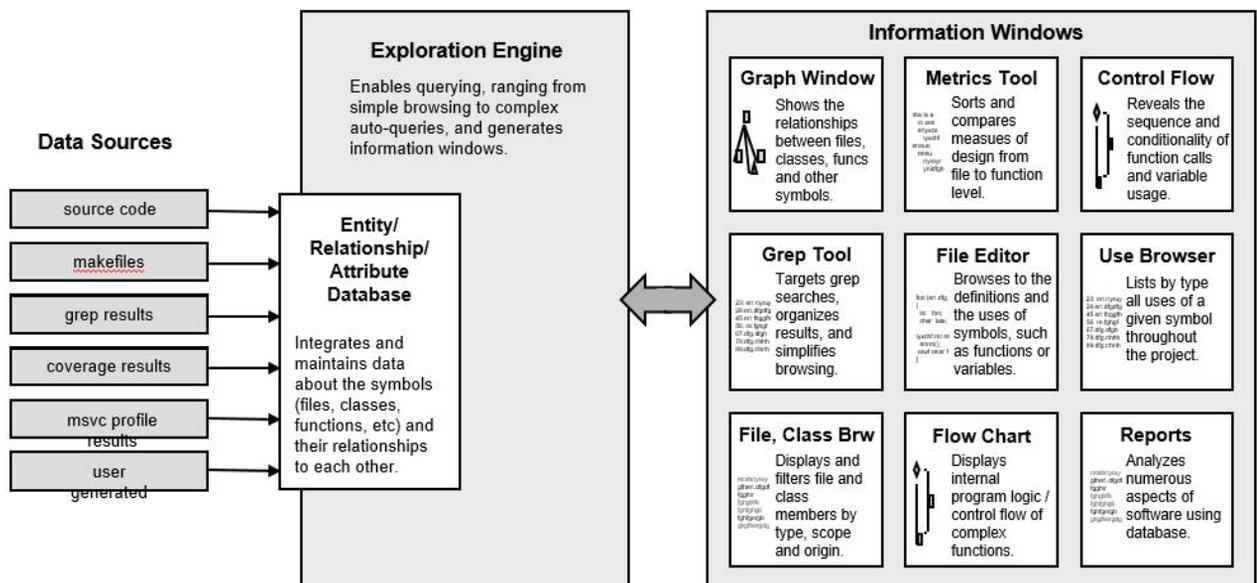


### 1000 - Research Concepts

- + Coding Standards Violation - (710)
- + Improper Access of Indexable Resource ('Range Error') - (118)
- + Improper Check or Handling of Exceptional Conditions - (703)
- + Improper Control of a Resource Through its Lifetime - (664)
- + Improper Enforcement of Message or Data Structure - (707)
- + Incorrect Calculation - (682)
- + Insufficient Comparison - (697)
- + Insufficient Control Flow Management - (691)
- + Interaction Error - (435)
- + Protection Mechanism Failure - (693)
- + Use of Insufficiently Random Values - (330)

## 可视化加快理解代码

- 提供一组显示窗口，每个窗口都进行了优化，以便快速、直观地了解特定类型的信息
- 使用可视化技术，如颜色和3D渲染来加快理解
- 超文本导航和上下文敏感的鼠标动作可以跨信息域进行跟踪





## 全面的语言支持包括:

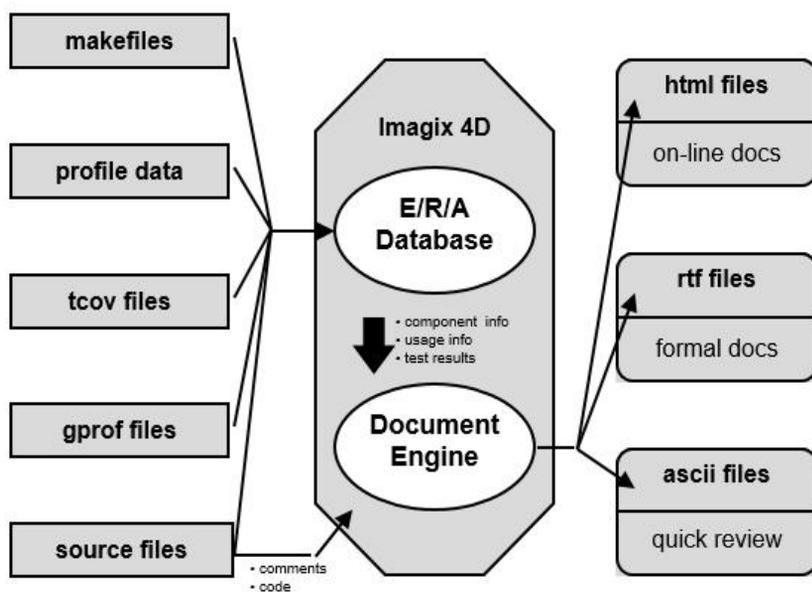
- 条件编译指令
- 类模板和命名空间
- Microsoft .NET扩展
- 通过Java反射API调用
- 函数指针
- 宏替换

## 配置文件支持:

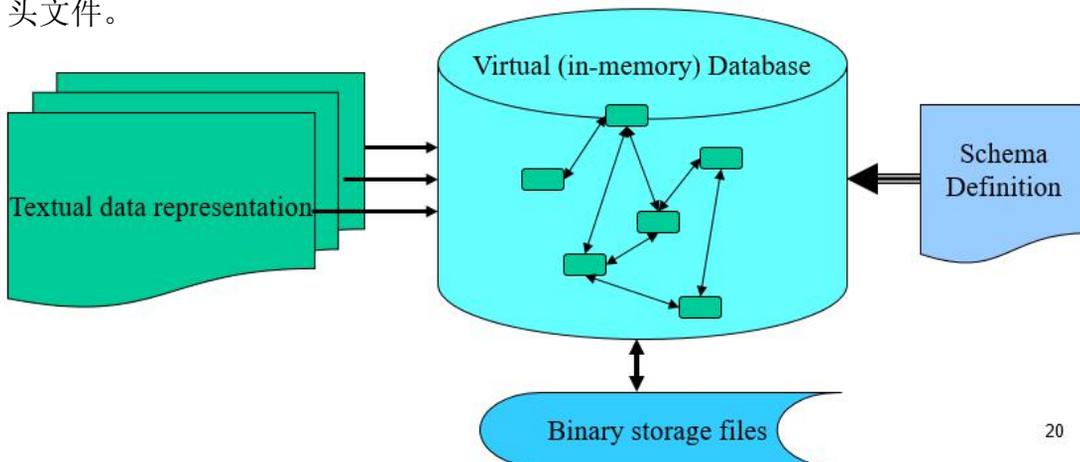
- GNU gcc编译器变种
- 其他原生编译器
- 嵌入式交叉编译器
- 多任务系统

## 文档引擎自动生成文档:

- 生成全面详细的设计文档
- 确保文件是最新和准确的
- 支持各种格式的不同用途
- 消除任务文档的时间和麻烦



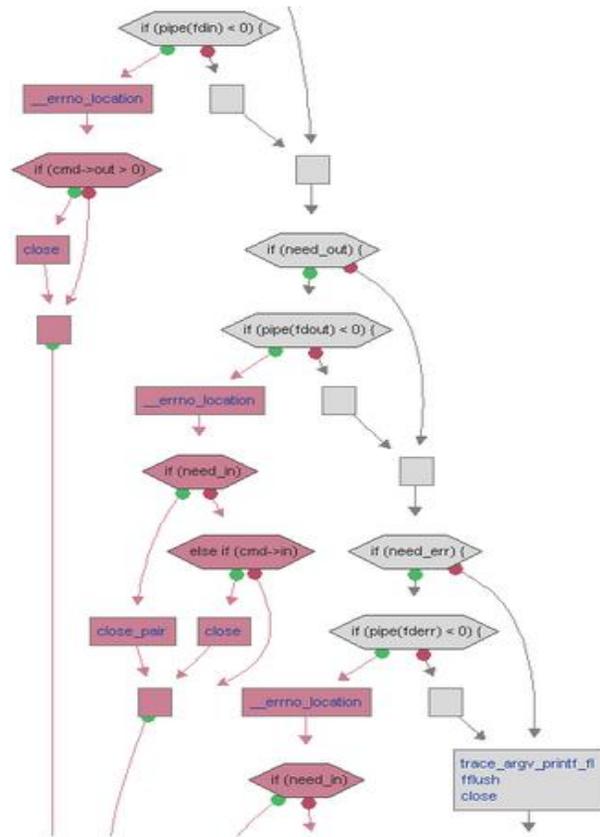
在Imagix 4D的强大的软件可视化和分析功能之下，是一个精确全面的数据库，通过对源代码的全面语义分析生成。通过使用编译器配置文件来实现，使Imagix 4D的源代码分析器能够以与特定编译器相同的方式解析您的代码，支持相同的语言扩展和关键字，并使用相同的系统头文件。



Imagix 4D的控制流分析显示，“计算树”显示了导致变量当前值的赋值信息。您可以检查特定变量值的所有初始化，集合和读取。通过数据流引擎从Imagix 4D的综合数据库自动生成，计算树通过功能边界和参数传递跟踪分配依赖关系。

### 计算树显示：

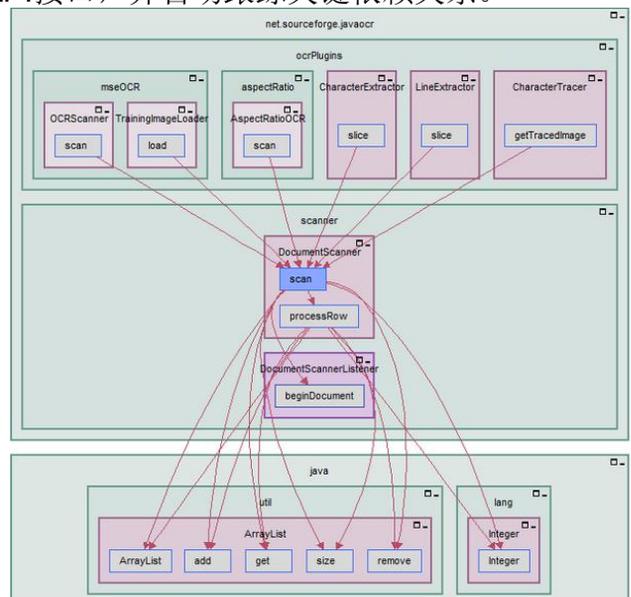
- 作业流程以图形方式显示所有层次结构
- 作业列表列出所有作业
- 变量依赖关系以图形方式显示所涉及的变量和参数的层次结构
- 变量表通过分配的变量组织单个作业。



Imagix 4D的软件可视化特别强大的是可以将显示器集中在感兴趣的信息上。通过数据库查找，图形查询和下拉功能，您可以识别和探索源代码的相关部分。通过分析器功能可以直接执行大量的分析。可以快速检查重要的API接口，并自动跟踪关键依赖关系。

### 分析器查询包括：

- 两个功能之间的路径
- 功能设置或读取变量的路径
- 共同的后裔或祖先的两个功能
- 功能类似函数
- 包含两个文件之间的路径
- 文件或类功能接口
- 文件或类变量接口
- 子系统，文件或类的内部调用层次结构
- 违反分层规则



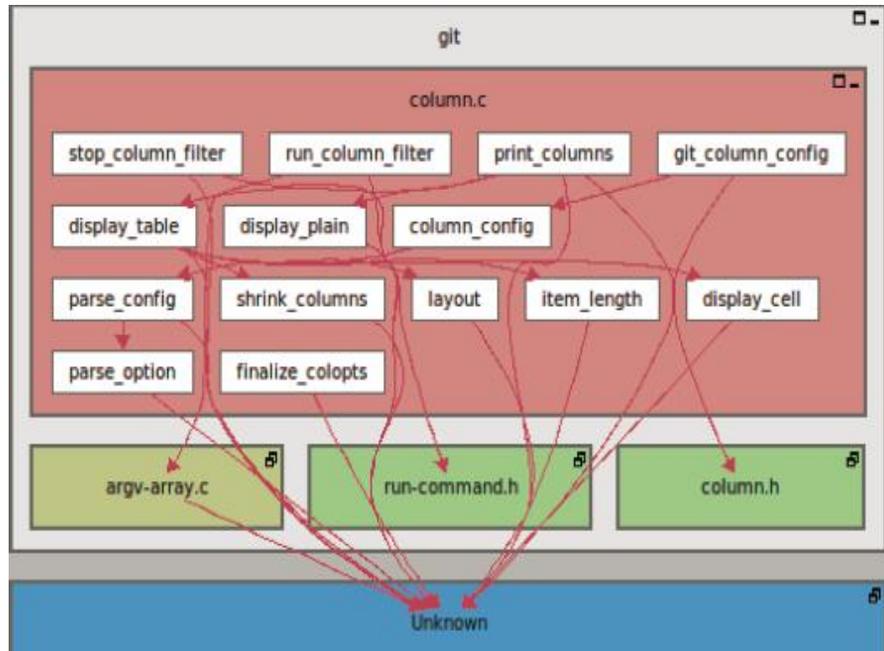
Imagix 4D提供多种视图类型，总共有15种，包括许多高级视图。这些为您的软件固有架构如：组件，依赖和包装提供了“图形化”视角和分析。这些高级视图使用抽象来表示源代码中实际发生的细节信息。例如，函数调用可以在类级别进行汇总，因此ClassA :: FunctionA调用ClassB :: FunctionB可以表示为ClassA调用ClassB。

#### 高级显示器包括：

- 子系统架构图
- 设计结构矩阵（DSM）
- UML类图
- 类继承视图
- 文件包含层次结构视图
- UML任务协作图

#### 抽象机制包括：

- 文件和类的抽象
- 子系统分层分析
- 用户通过分组定义抽象级别



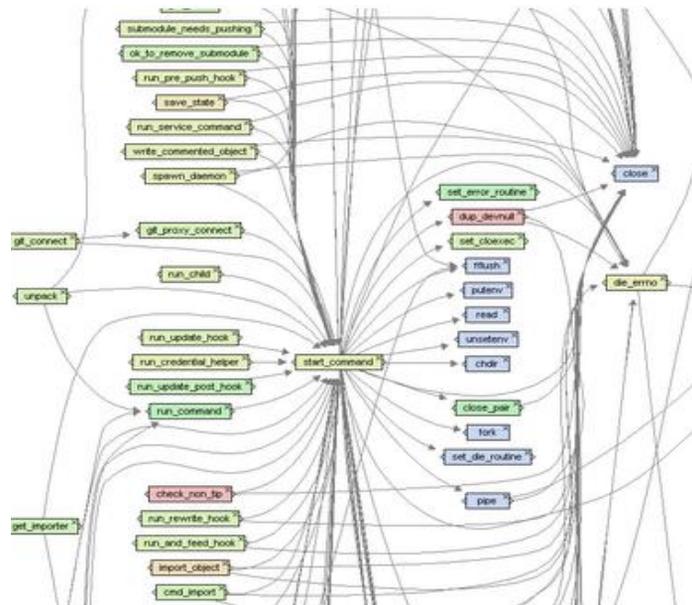
您也可以使用高级别的视图分析源代码，通过点击显示更多细节。通过逐渐细化的视图进行深入，更详细地了解软件的特定部分。找到并分析您的软件控制流，数据使用和类型依赖。通过Imagix 4D的交互式图表，可以遵循特定的实现方式，并追踪其对代码其他部分的影响。

#### 中级显示器包括：

- 函数调用
- 变量集和读取
- 功能宏使用
- 类使用，数据类型层次结构

#### 低级显示器包括：

- 显示功能内程序逻辑的流程图
- 控制流图显示调用顺序和函数调用的条件
- 具有可变使用的控制流程



# 携手开拓无限可能

Together to create all possible

