Fortify SCA
产品使用手册

道普云（山东）智能科技有限公司
技术支持：18266417701
客服微信：daopuyun

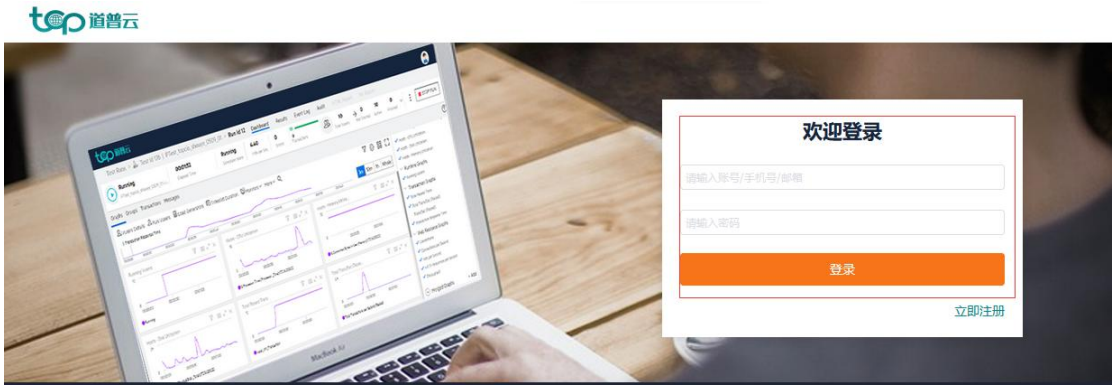
目录

1. 总体使用流程

用户登录-创建测试用户-创建测试项目域-创建测试项目-使用代码测试工具 Fortify SCA-同步测试数据

2. 用户登录

(1) 打开平台首页并输入登录用户名和密码



3. 创建测试用户

(1) 打开测试用户新增页面

(2) 输入测试用户信息

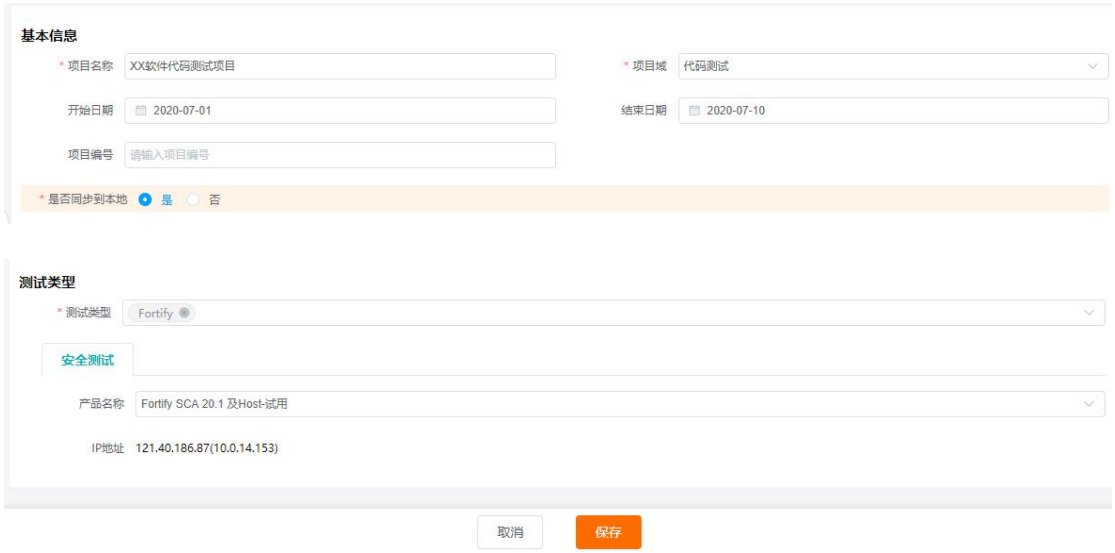
4. 创建代码测试项目

(1) 打开测试项目域新增页面

(2) 输入测试项目域名称

(3) 打开测试项目新增页面

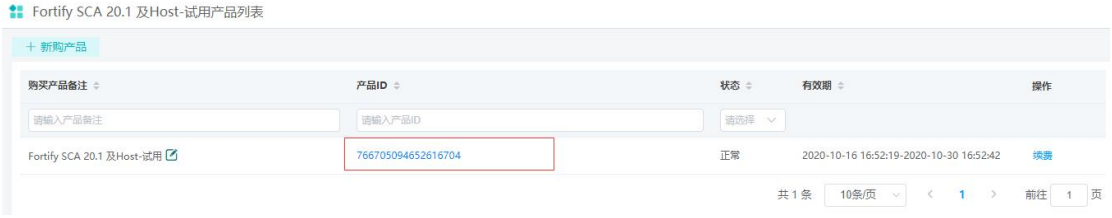
(4) 输入测试项目信息，包括基本信息、测试类型，点击保存，平台返回 Fortify SCA 工具所在云主机 IP 地址信息



5. 使用代码测试工具

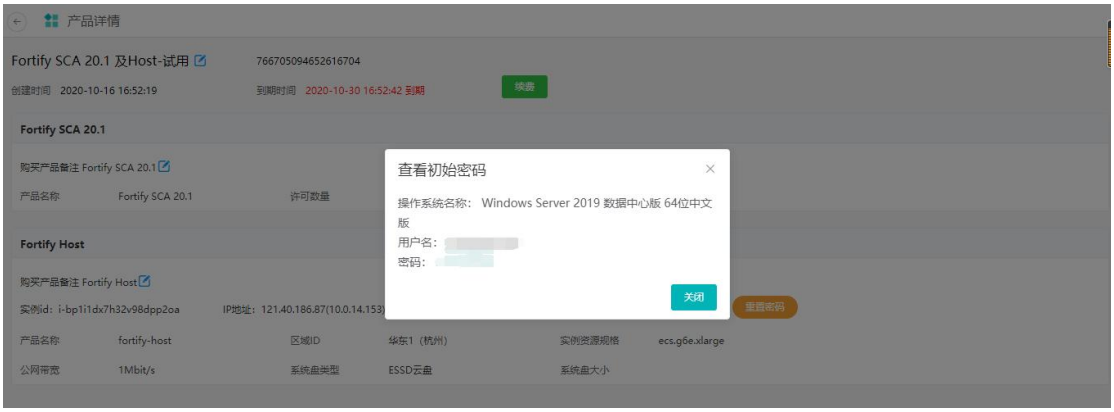
(1) 点击 Fortify SCA 产品信息，查看已购买或试用的产品信息

(2) 点击产品 ID，进入产品详情页

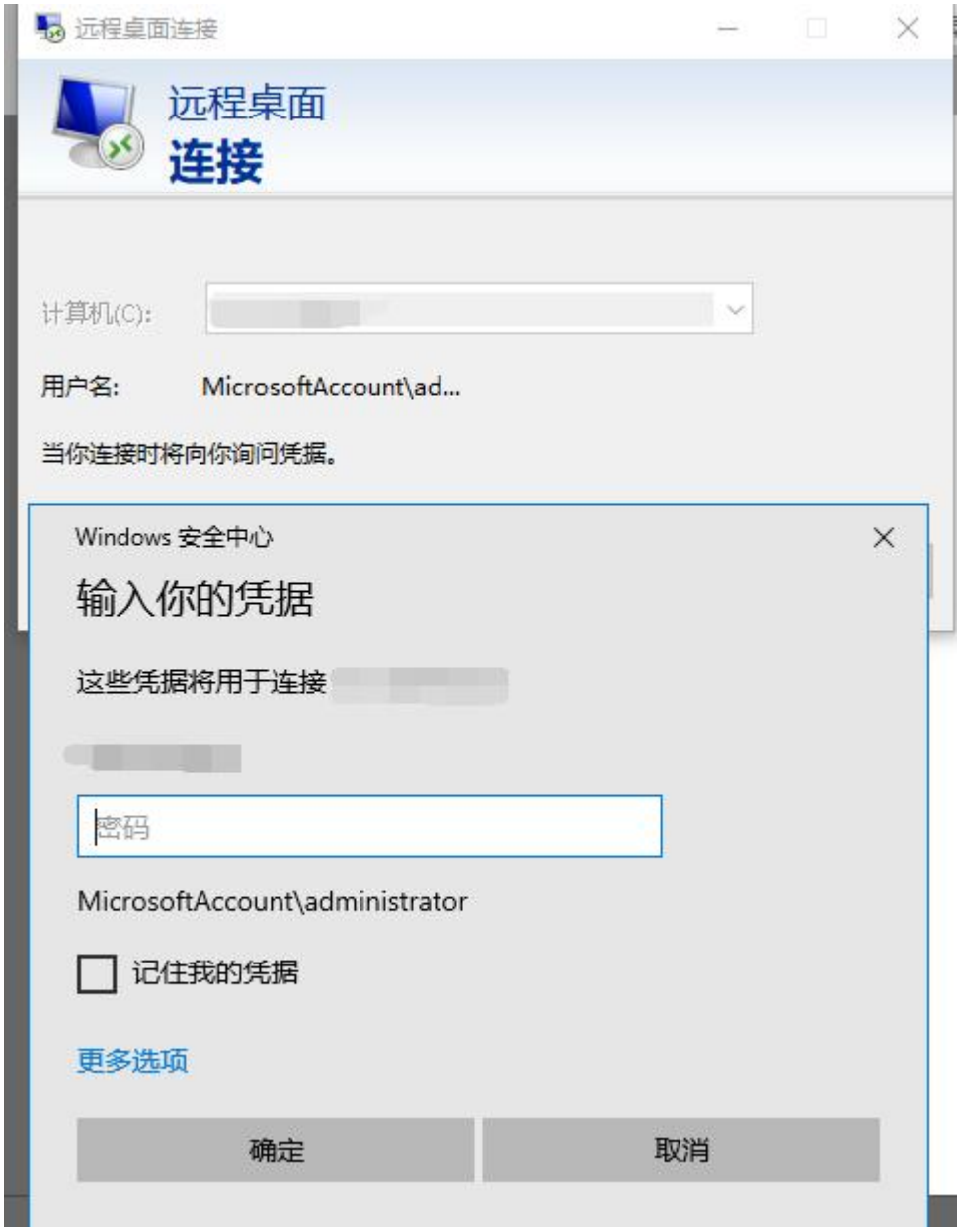


(3) 在产品详情页，点击“查看初始密码”

(4) 从弹出窗口查看“初始密码”



(5) 根据获取的 Fortify SCA 云主机 IP 地址、用户名和口令，远程登录到该云主机



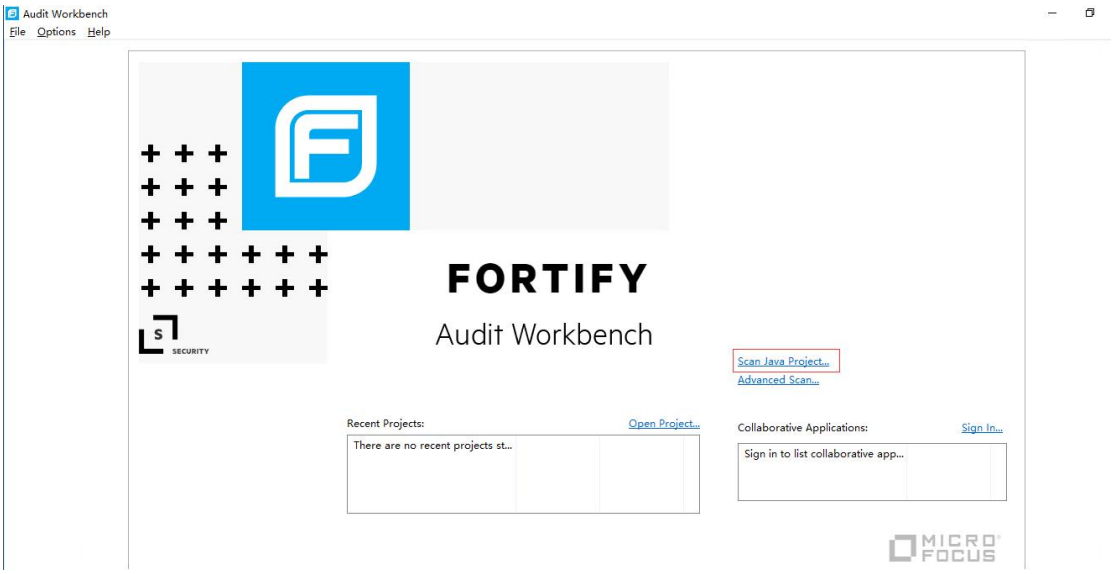
(6) 打开桌面 Fortify SCA 20.1 代码审计引擎

5.1. 通过“Audit Workbench”进行测试

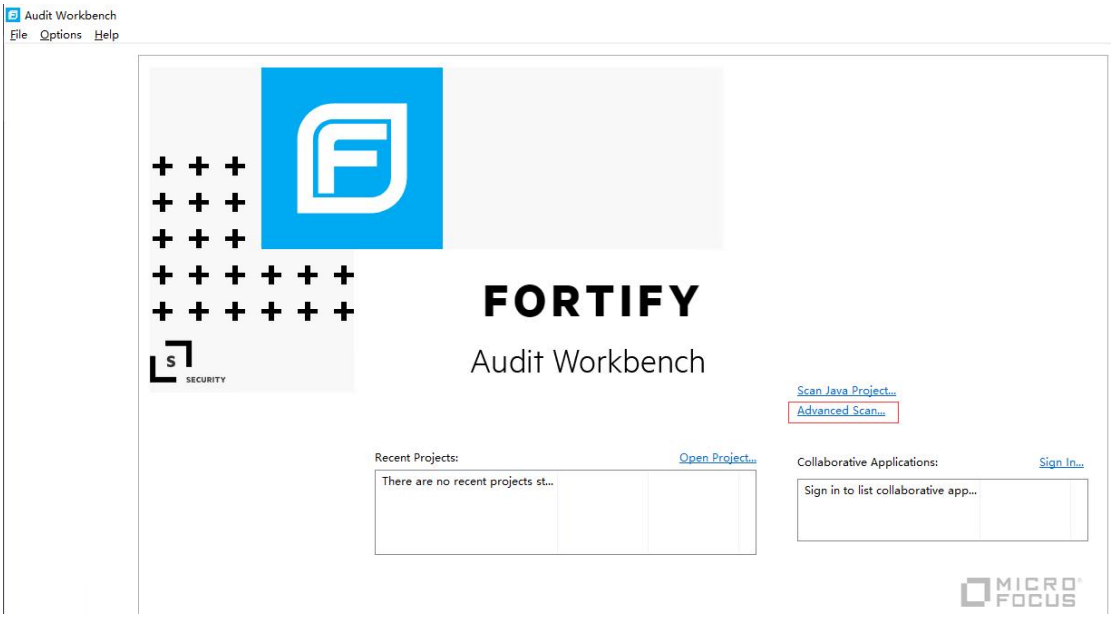
“Audit Workbench”支持 Java 语言源代码的测试。

(1) 在主页面选择代码测试语言类型

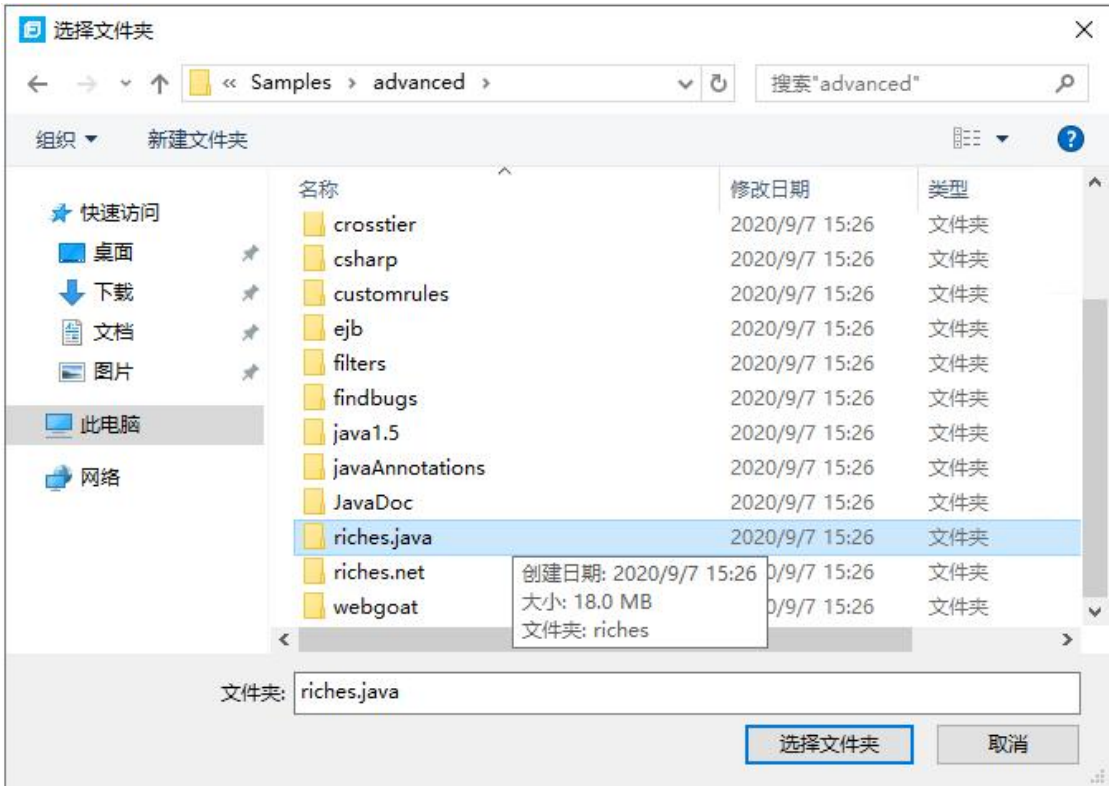
1）选择“Scan Java Project”



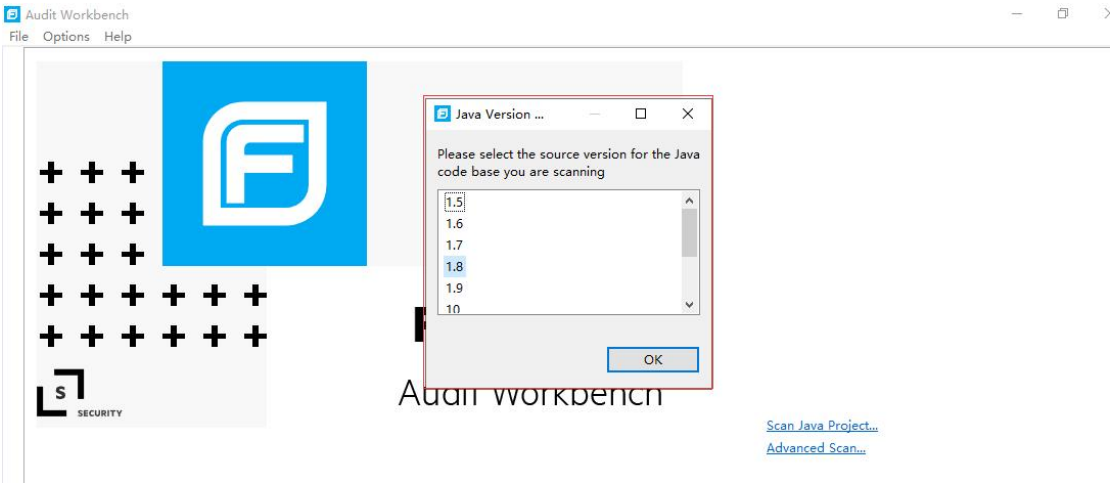
2）如果是非 Java 语言，选择“Advanced Scan”



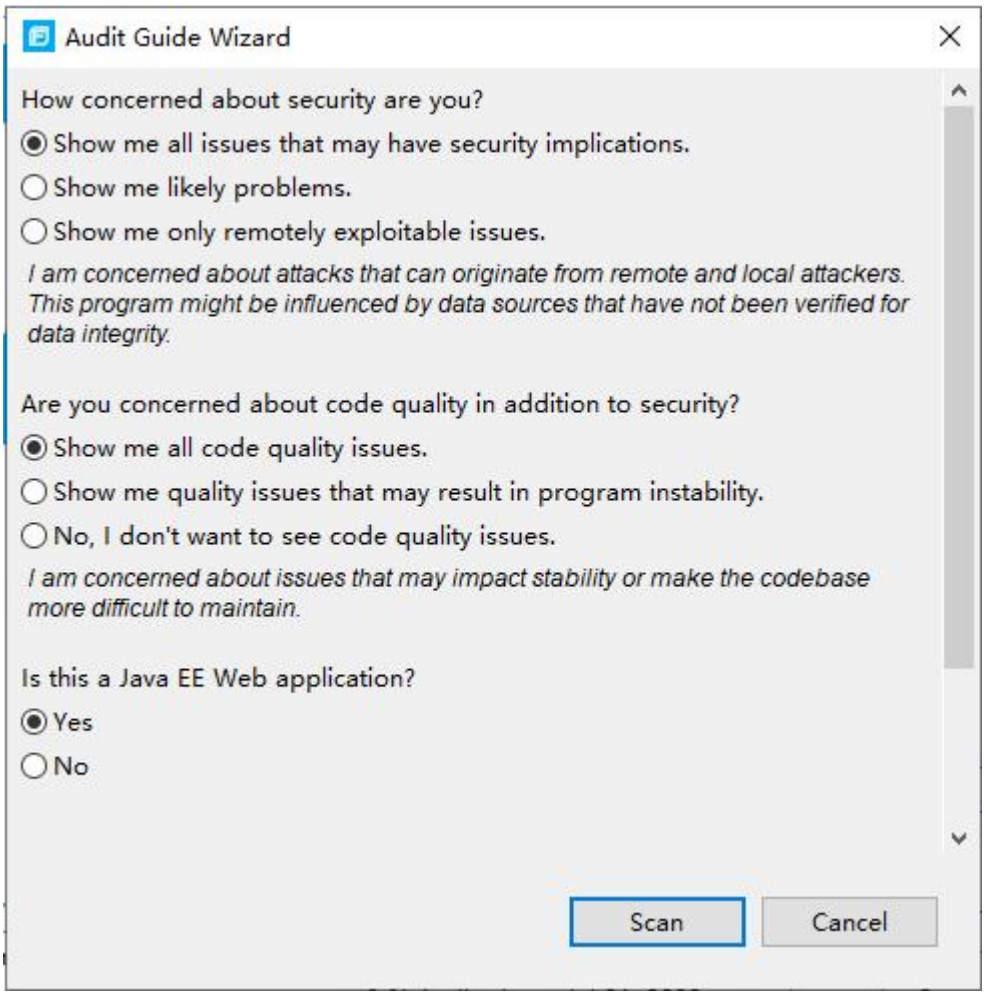
(2) 选择被测试代码所在目录



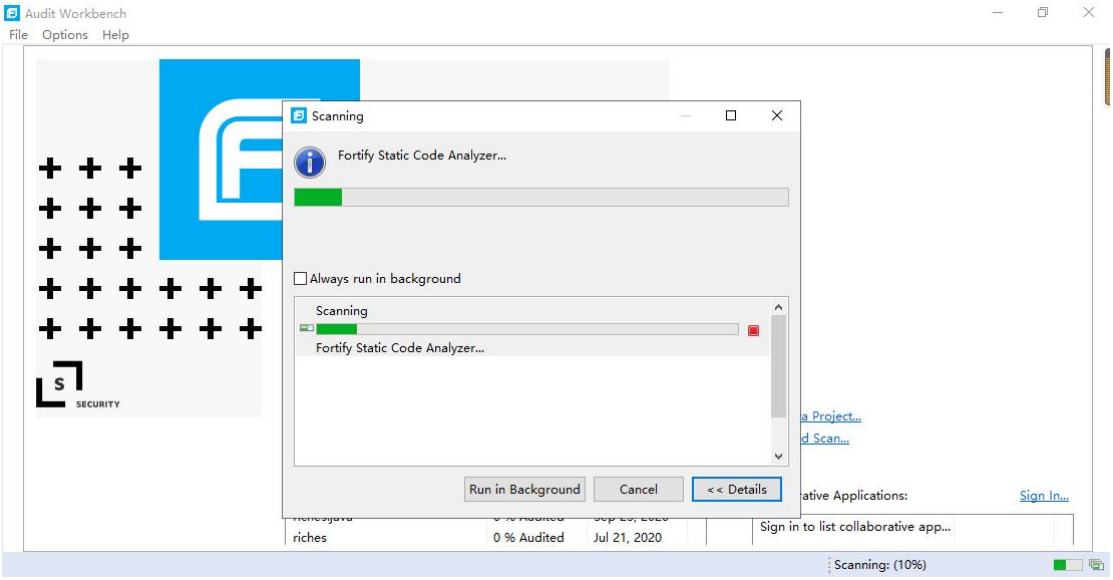
(3) 选择 Java 版本



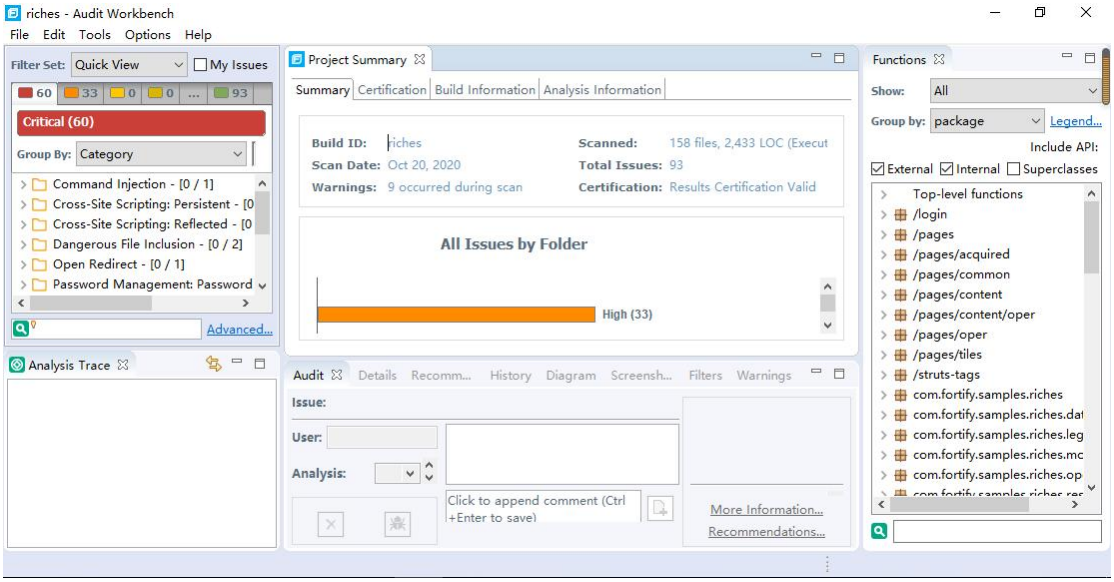
(4) 进行代码测试配置



(5) 运行代码测试



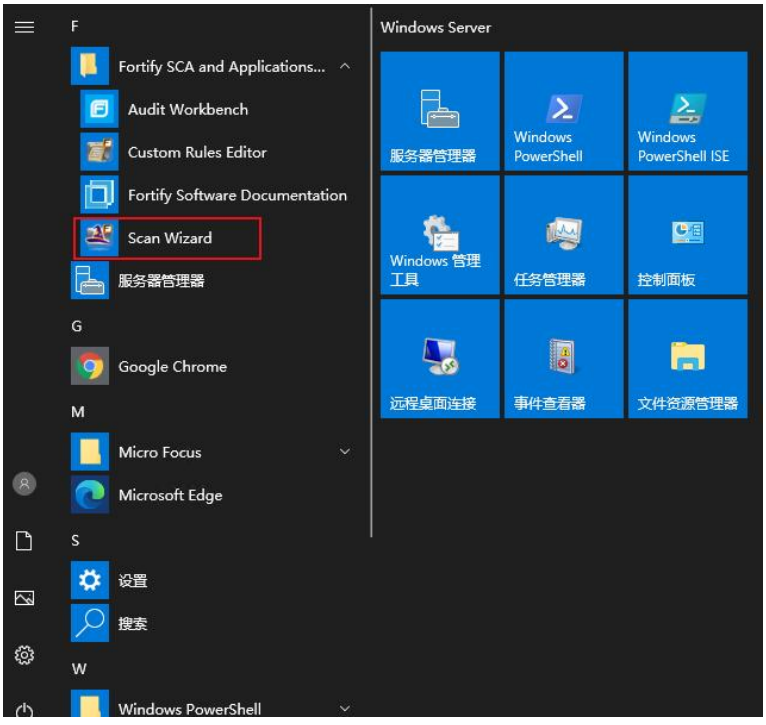
(6) 查看代码测试结果



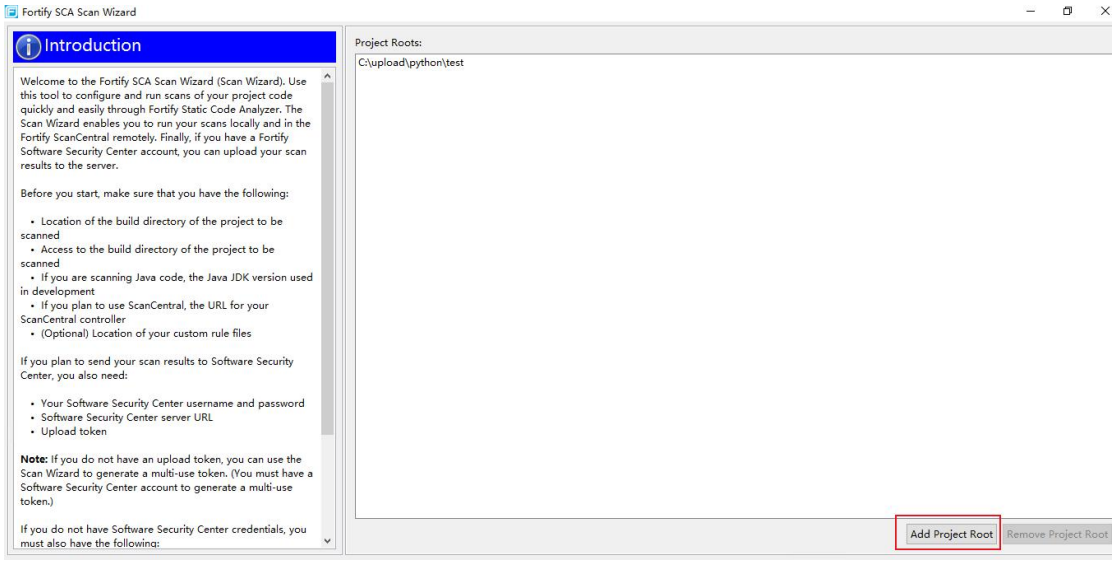
5.2. 通过“Scan Wizard”进行测试

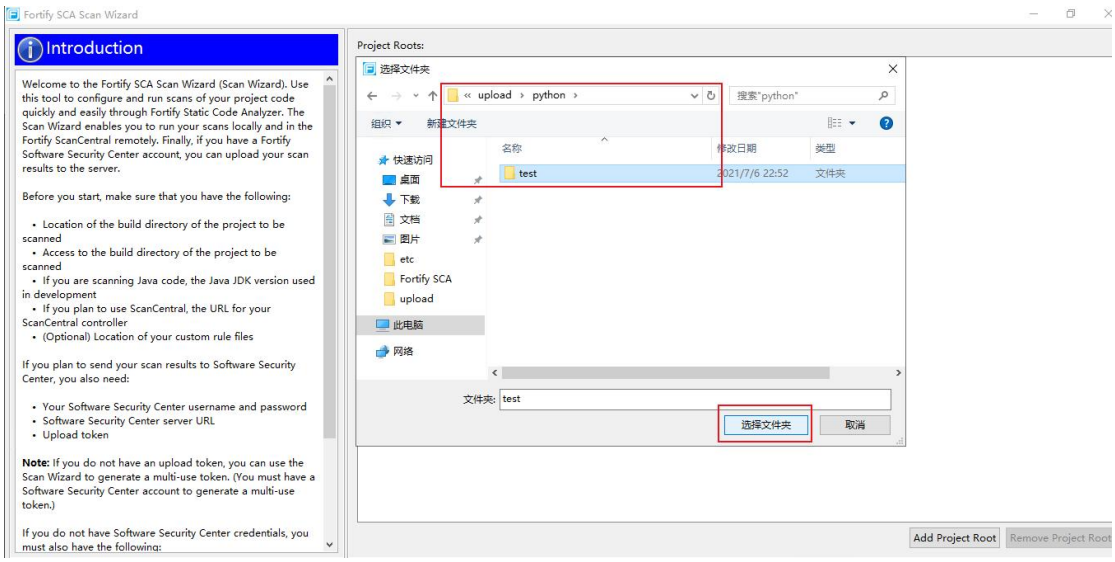
“Scan Wizard”支持 Java、Python、C/C++、.Net、Go、PHP、Flex、Action Script、HTML、XML、JavaScript、TypeScript、Kotlin、SQL、ABAP、ColdFusion 语言或框架源代码的测试。

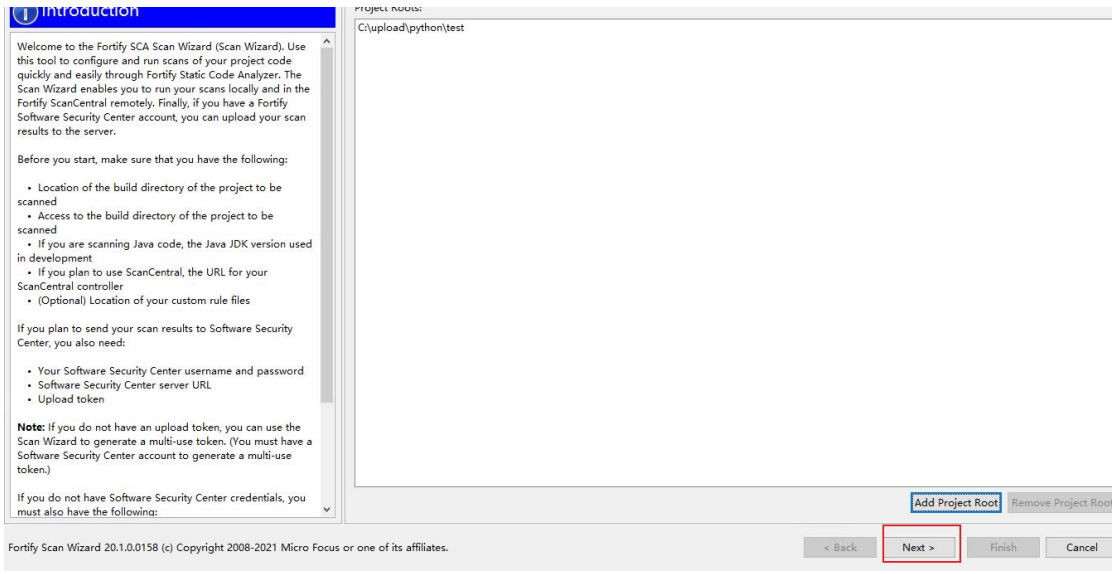
（1） 打开 Scan Wizard



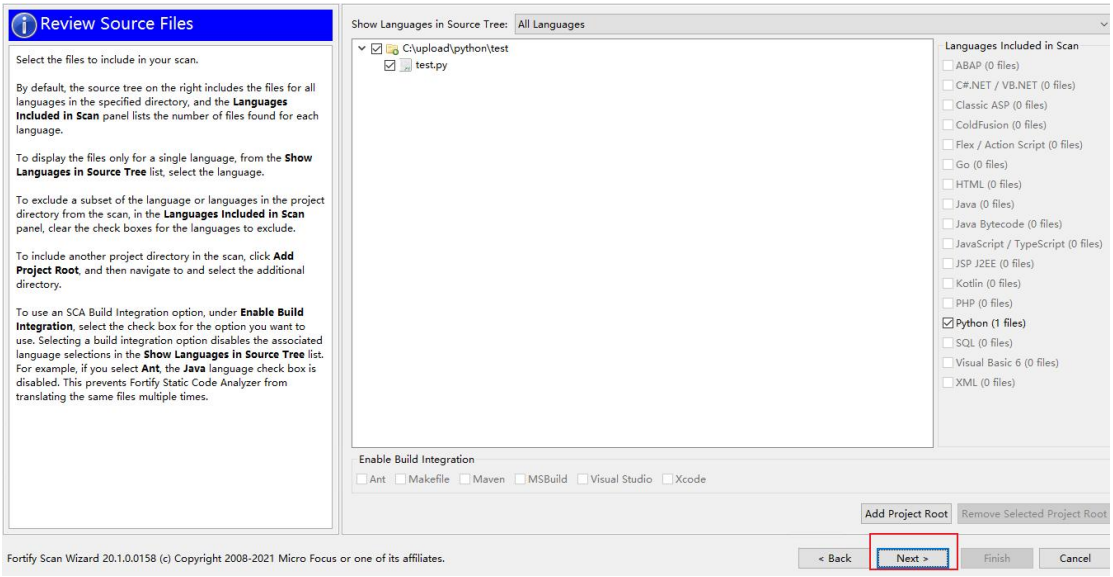
（2） 选择 Python 文件所在目录



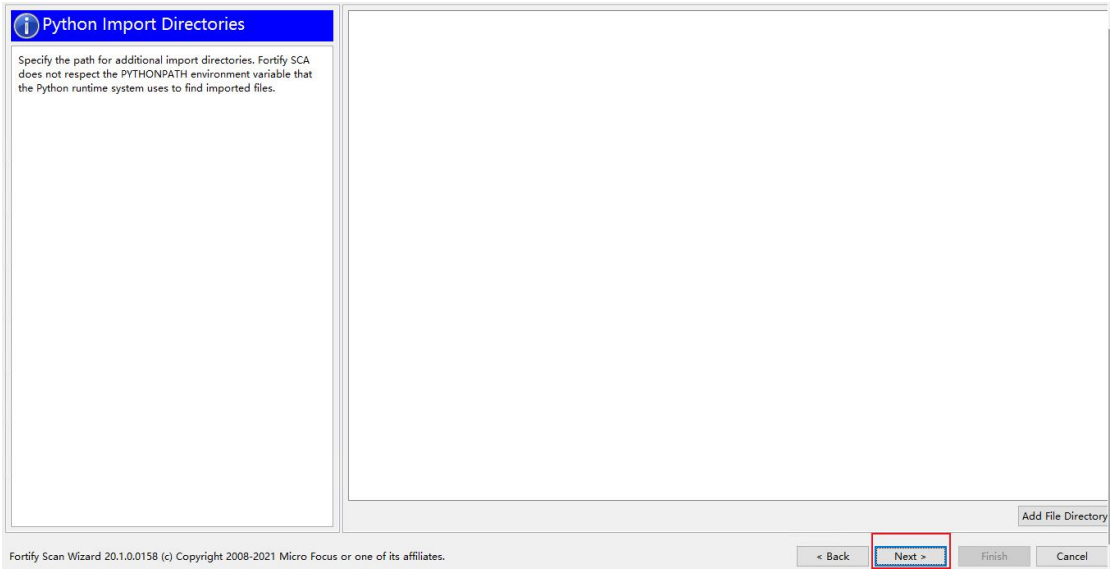




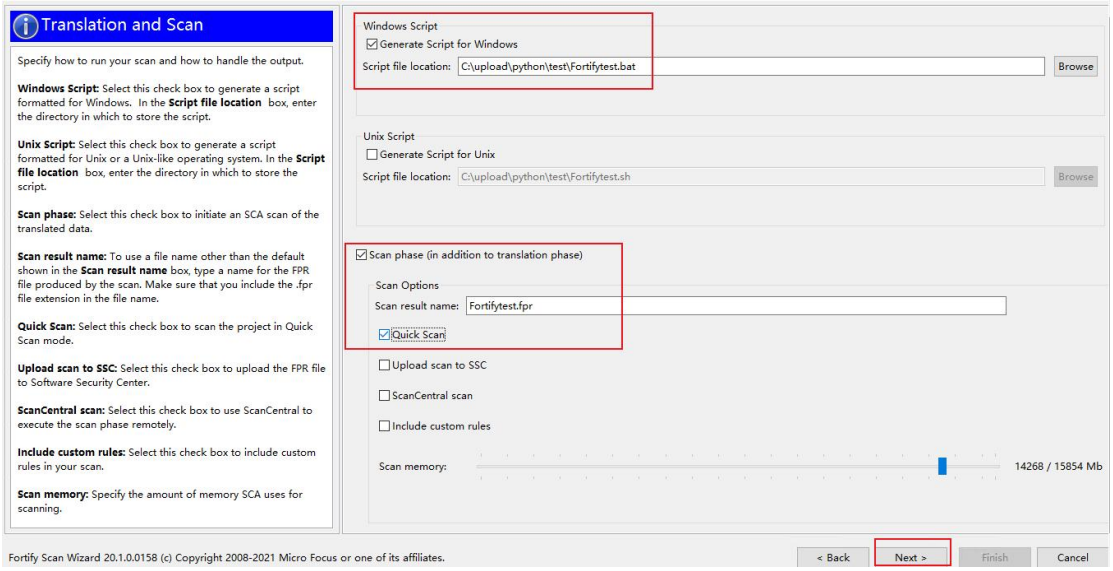
(3) 确认测试工具自动识别内容



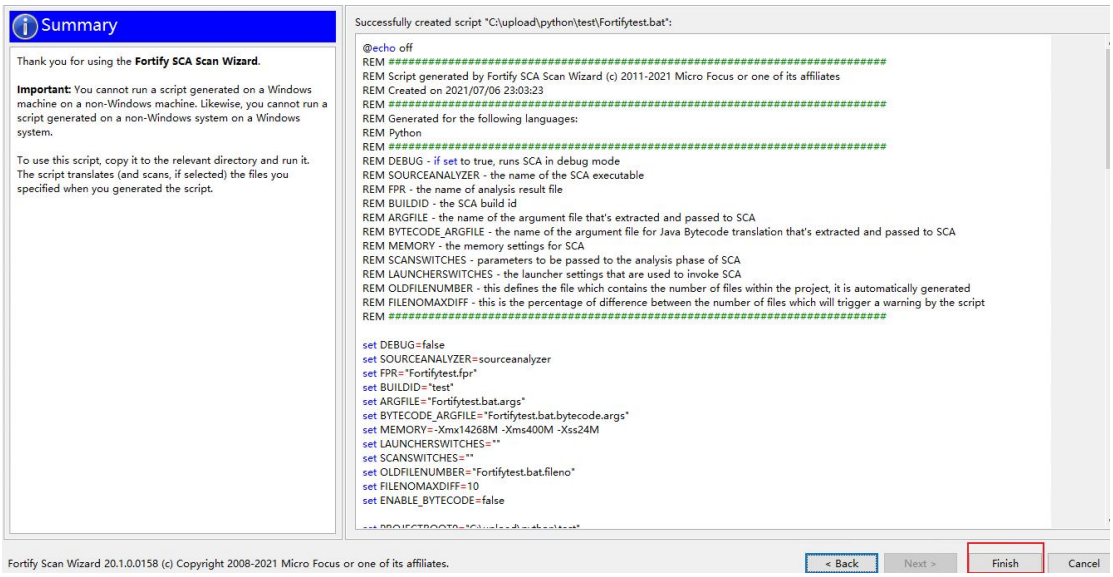
(4) 选择库文件



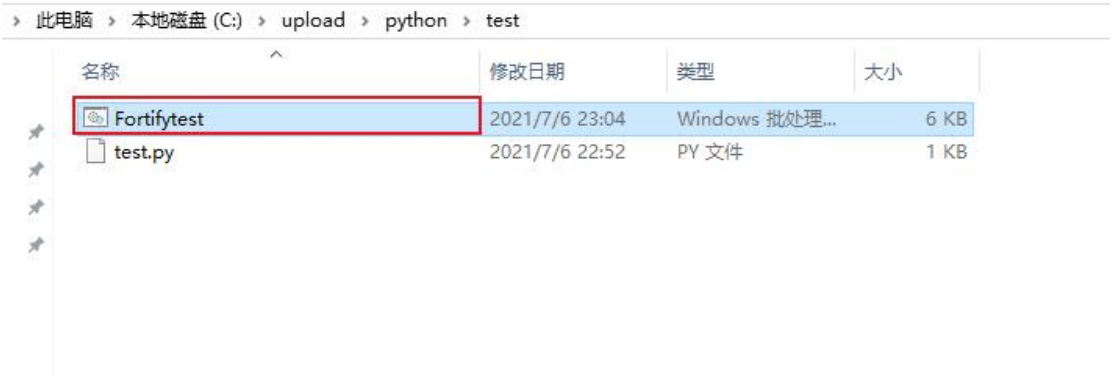
(5) 生成脚本文件



(6) 完成脚本文件生成



(7) 执行生成的脚本文件



5.3. 通过命令行进行测试

命令行方式支持各语言源代码的测试。

5.3.1. Linux 项目测试

以 Linux 下 C/C++程序代码测试为例：

1. 代码编译

   在代码测试执行前，首先需要进行 C/C++程序代码的编译，如下面的示例：

   gcc –I. –o hello.o –c helloworld.c

   通过 gcc 编译器将代码进行编译。

2. 代码测试

   在代码编译后，使用 sourceanalyzer 命令进行代码文件测试。

   sourceanalyzer –b *<build_id>* gcc –I. –o hello.o –c helloworld.c

3. 代码扫描结果文件生成

   在代码测试后，使用 sourceanalyzer 命令进行代码文件扫描及结果文件生成。

   sourceanalyzer –b *<build_id> –scan –f hello.fpr*

   其中，本命令中的<build_id>与第 2 步命令中的<build_id>相同。成功生成结果文件后，可以基于该结果文件生成测试报告。

4. 代码扫描结果文件生成

   在代码测试后，使用 sourceanalyzer 命令进行代码文件扫描及结果文件生成。

5.3.2. iOS 项目测试

1. iOS 项目测试条件

(1) iOS 项目需要使用 non-fragile Objective-C runtime 模式（ABI version 2 或 3）

(2) 使用 Apple “xcode-select command-line tool”设置 Xcode path，同时供 Fortify 使用。

(3) 确保项目相关依赖库文件已经包含在项目中。

(4) 针对 Swift 代码，确保所有第三方模块都已经被包含，包括 Cocoapods。

(5) 如果项目中包含二进制的属性列表文件，需要将它们转化为 XML 格式，通过 Xcode 的 putil 命令进行转换。

(6) 针对 Objective-C 项目，需要保证头文件能够被获取。

(7) 针对 WatchKit 应用，需要同时转化 iPhone 应用和 WatchKit 扩展目标。

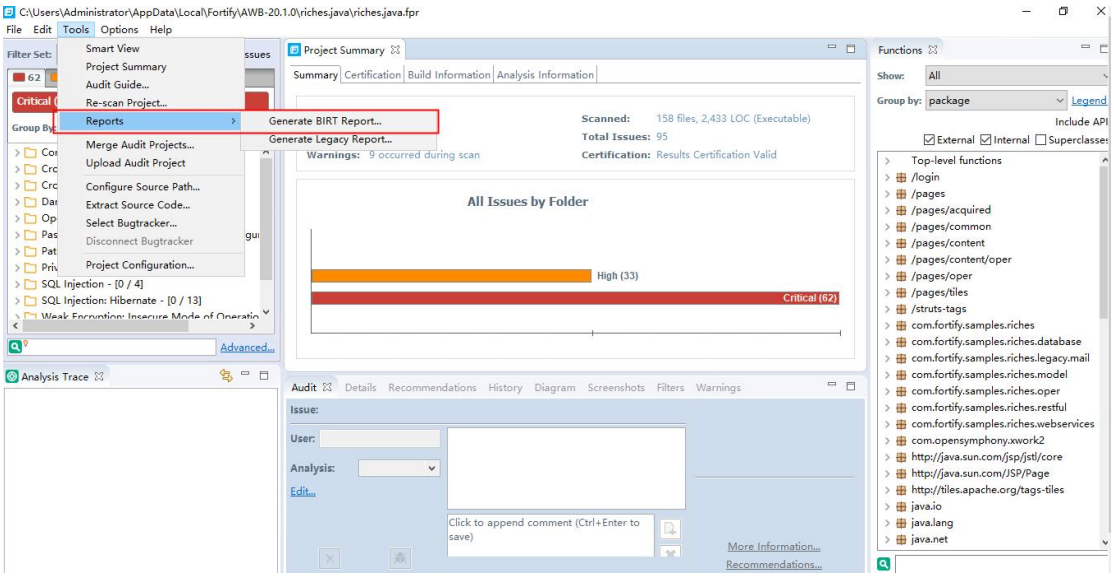
2. iOS 代码测试执行

sourceanalyzer -b *<build_id>* xcodebuild [*<compiler_options>*]

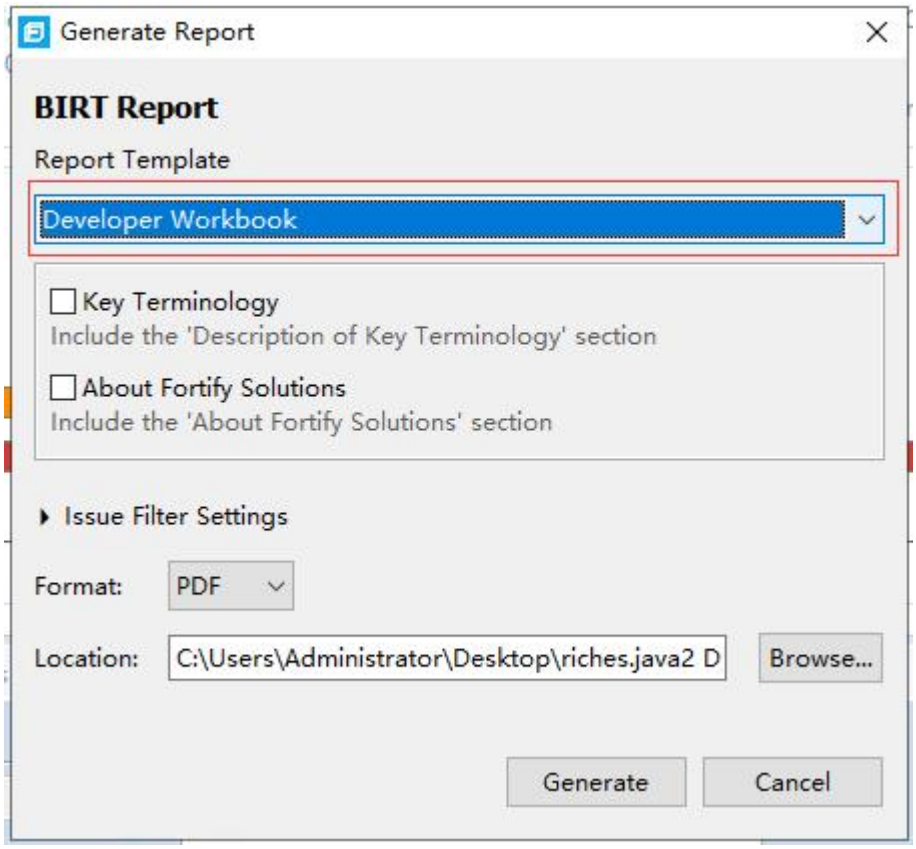
6. 测试报告生成

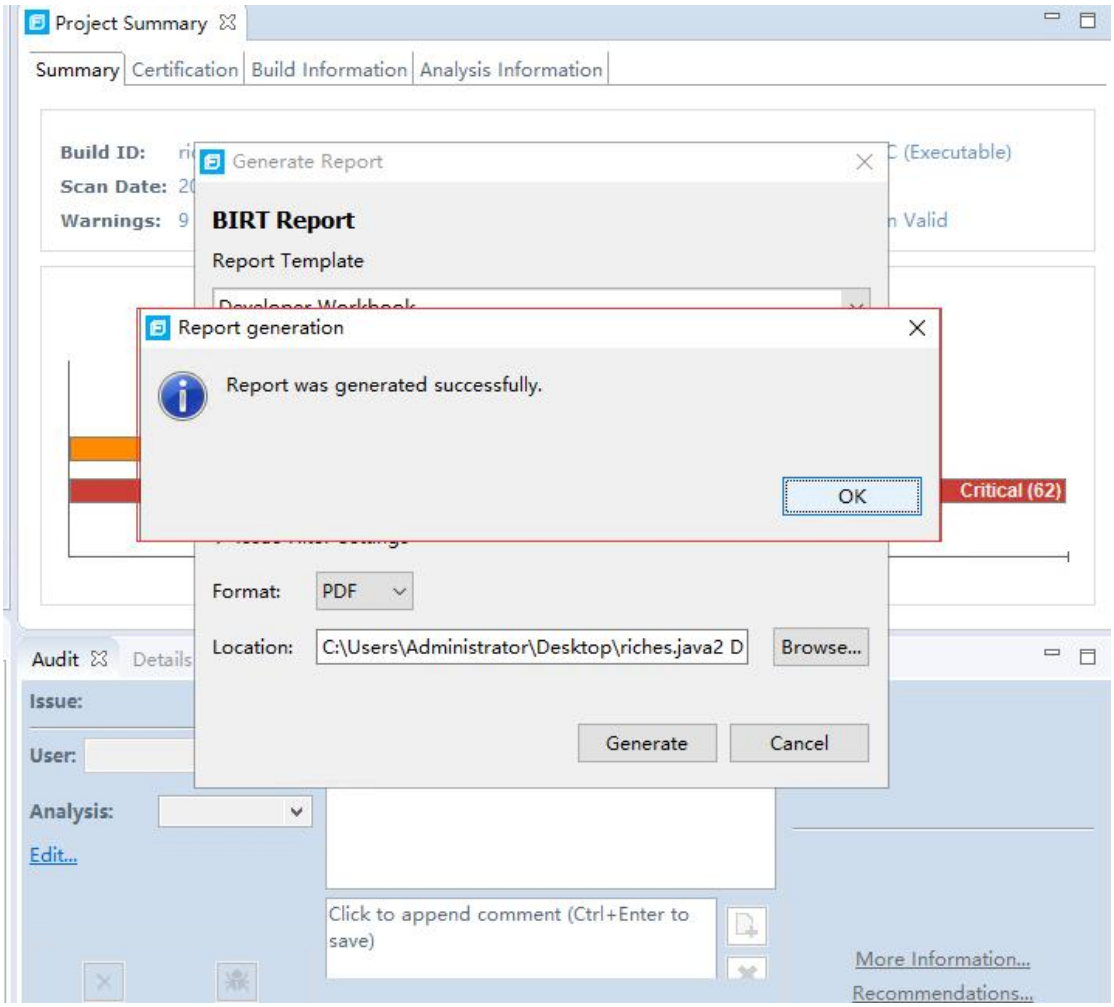
6.1. 通过“Audit Workbench”生成测试报告

1.打开“Audit Workbench”中的“Tools-Reports”，选择“Generate BIRT Report”或者“Generate Legacy Report”。



2.在报告模板“Report Template”中选择“Developer Workbook”，点击“Generate”按钮，工具会自动生成报告。





6.2. 通过“Scan Wizard”生成测试报告

通过“Scan Wizard”方式进行测试执行，会生成.fpr 测试结果文件，然后通过命令行方式基于测试结果文件生成测试报告文件。

6.3. 通过命令行生成测试报告

通过“Scan Wizard”方式或命令行方式生成测试结果文件后，可以基于“ReportGenerator”命令生成测试报告。

下面示例中，基于.fpr 结果文件生成 PDF 格式的测试报告。

ReportGenerator –format pdf –f *<my_report>*.pdf –source *<my_results>*.fpr

<my_report>.pdf 为命名的 PDF 格式测试报告名称，*<my_results>*.fpr 为测试结果

文件名称。