软件源代码相似性检测鉴定系统

CC Compare Forensic

使用手册
版本 V2.0
20220708

目录

第一章：系统概况

1.1 系统简介

软件源代码相似性检测鉴定系统，英文缩写为 CC Compare 或 CC Compare Forensic(Code Content Compare Forensic)，由北京正乙科技有限公司(www.zyinsight.com)设计开发。该软件是一款高度智能化、操作极简化、比对批量化、跨平台、全场景应用的软件源代码相似检测鉴定系统，主要功能包括代码集合的相似性检测、侵权证据发现、专家校验、智能分析、鉴定报告、鉴定附件等功能。系统由具有 10 余年软件知识产权司法鉴定经历专家基于 300 余件真实的软件诉讼案件特征，采用独有的源代码相似性检测算法，按照普通模式，专家一级、专家二级、专家三级、专家四级、专家五级别对软件源代码进行各种深入程度的相似性检测。与一般的代码比对工具不同，该系统能够有效的发现代码重构的侵权证据，对于多个代码文件拆分、重命名，单个代码文件重命名、代码顺序混淆、代码截断、逻辑混淆等能够进行有效的检测，并能够实现批量的代码对应关系发现、相似性检测、检测结果智能分析、可视化、自动撰写鉴定报告和生成鉴定附件。同时，该系统支持多台计算机并行处理，缓存机制，用以加快大规模比对的速度。

为了提高比对速度，简化用户操作，软件采用 DOS 界面，键盘指令操作的模式，使用户在 5 分钟内掌握软件基本使用方法。

1.2 系统用户

与一般面向软件开发人员的代码比对工具不同，该系统主要面向软件知识产权纠纷企业、鉴定机构、公安、检察院等，适用于软件著作权和商业秘密纠纷的各种场景。当你不确定代码是否侵权时，试试 CC Compare Forensic！当你有大量代码需要快速检测时，试试 CC Compare Forensic！当你只有目标代码，没有源代码时，试试 CC Compare Forensic！当你需要快速出具权威检测鉴定报告时，试试 CC Compare Forensic！当你不确定鉴定机构能否给出期望的鉴定结论时，试试 CC Compare Forensic！

1.3 典型案例

CC Compare Forensic 经典案例应用：王者荣耀自由战士游戏软件纠纷案、APUS 手机清理软件被侵权纠纷案、王者之剑斗龙传游戏软件纠纷案、小鱼易连宝利通视频软件纠纷案、理正建筑信息管理软件被侵权案、傲创机场管理系统软件被侵权案、江苏金智教育系列软件被侵权案、迈瑞理邦医疗设备软件纠纷案、康凯斯车载导航软件被侵权案、GE 色谱仪嵌入式软件被侵权案。

第二章：安装注册

2.1 系统安装

软件采用了绿色版，64 位程序，解压缩后即可运行，需要安装 64 位 Net6.0 以上环境，下载地址为：https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-6.0.4-windows-x64-installer，操作系统建议 Windows10 或 Windows11，但同时支持 Mac 和 Linux 系统，仅需安装对应的 Net 环境即可。

2.2 系统注册

软件使用必须带有注册文件，注册文件与计算机硬件、IP 地址绑定，需要从我司单独购买，否则软件无法启动，也无法使用。软件启动必须联网用以验证软件是否正版，验证通过后可断网工作。

	IP 锁定	Mac 锁定	单次数量锁定
基础版本（初级权限）	☑	☑	☑
高级版本（中级权限）	☒	☑	☑
集团版本（高级权限）	☒	☒	☒

基础版本，高级版本限定在一台计算机使用，集团版本可在多台计算机使用。

2.3 各版本功能

V2.0：1）基于内存进行文件缓存，速度较 1.0 版本提升 2 个数量级

2）增加硬盘缓存，提升报告生成速度

3）优化 HTML 格式报告显示样式

4）增加 IP 地址、Mac 地址、单次比对数量锁定

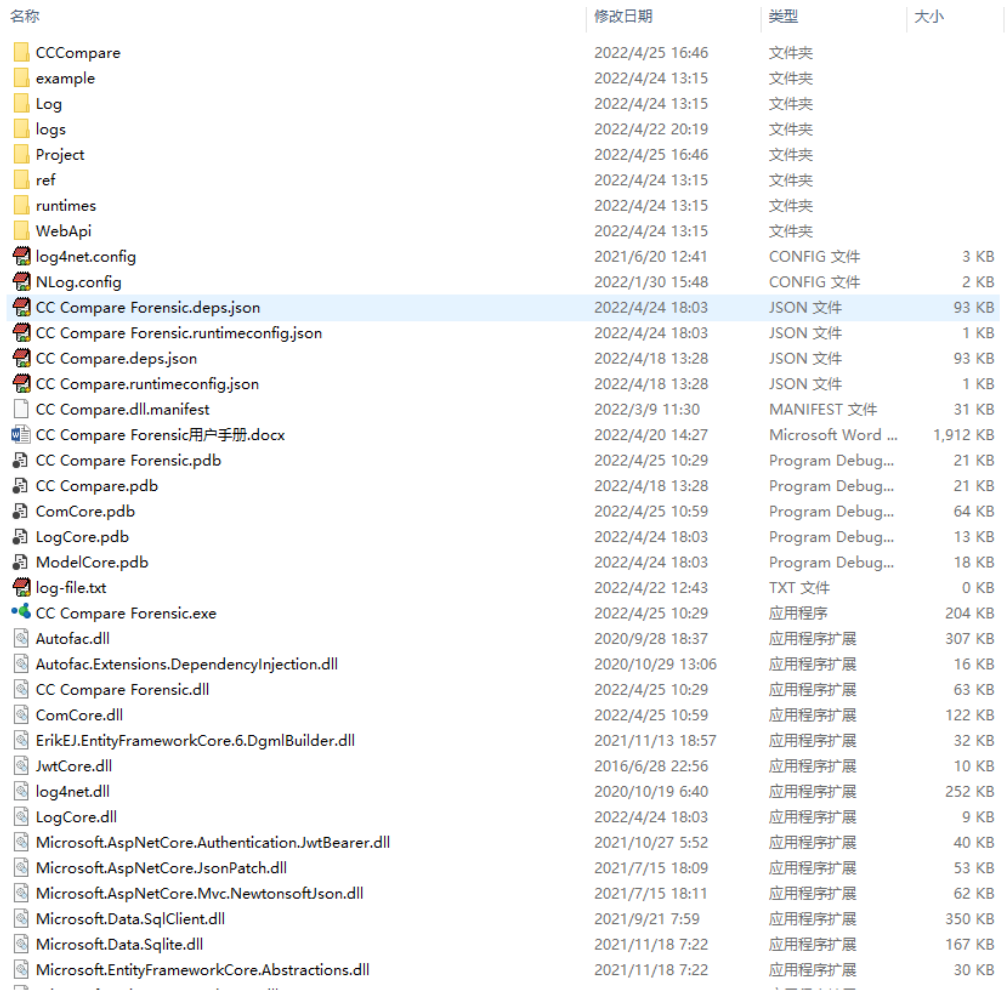
V1.0：代码集合的相似性检测、侵权证据发现、专家校验、智能分析、鉴定报告、鉴定附件加大功能。

2.4 推荐硬件

一般普通计算机即可使用软件，推荐使用 8G 以上内存；在大数据量情况下，建议提高内存至 32G，可明显提高比对速度。

第三章：系统目录

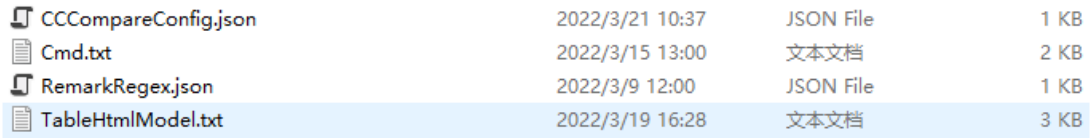
软件目录如下图所示：



名称	修改日期	类型	大小
CCCompare	2022/4/25 16:46	文件夹	
example	2022/4/24 13:15	文件夹	
Log	2022/4/24 13:15	文件夹	
logs	2022/4/22 20:19	文件夹	
Project	2022/4/25 16:46	文件夹	
ref	2022/4/24 13:15	文件夹	
runtimes	2022/4/24 13:15	文件夹	
WebApi	2022/4/24 13:15	文件夹	
log4net.config	2021/6/20 12:41	CONFIG 文件	3 KB
NLog.config	2022/1/30 15:48	CONFIG 文件	2 KB
CC Compare Forensic.deps.json	2022/4/24 18:03	JSON 文件	93 KB
CC Compare Forensic.runtimeconfig.json	2022/4/24 18:03	JSON 文件	1 KB
CC Compare.deps.json	2022/4/18 13:28	JSON 文件	93 KB
CC Compare.runtimeconfig.json	2022/4/18 13:28	JSON 文件	1 KB
CC Compare.dll.manifest	2022/3/9 11:30	MANIFEST 文件	31 KB
CC Compare Forensic用户手册.docx	2022/4/20 14:27	Microsoft Word ...	1,912 KB
CC Compare Forensic.pdb	2022/4/25 10:29	Program Debug...	21 KB
CC Compare.pdb	2022/4/18 13:28	Program Debug...	21 KB
ComCore.pdb	2022/4/25 10:59	Program Debug...	64 KB
LogCore.pdb	2022/4/24 18:03	Program Debug...	13 KB
ModelCore.pdb	2022/4/24 18:03	Program Debug...	18 KB
log-file.txt	2022/4/22 12:43	TXT 文件	0 KB
CC Compare Forensic.exe	2022/4/25 10:29	应用程序	204 KB
Autofac.dll	2020/9/28 18:37	应用程序扩展	307 KB
Autofac.Extensions.DependencyInjection.dll	2020/10/29 13:06	应用程序扩展	16 KB
CC Compare Forensic.dll	2022/4/25 10:29	应用程序扩展	63 KB
ComCore.dll	2022/4/25 10:59	应用程序扩展	122 KB
ErikEJ.EntityFrameworkCore.6.DgmlBuilder.dll	2021/11/13 18:57	应用程序扩展	32 KB
JwtCore.dll	2016/6/28 22:56	应用程序扩展	10 KB
log4net.dll	2020/10/19 6:40	应用程序扩展	252 KB
LogCore.dll	2022/4/24 18:03	应用程序扩展	9 KB
Microsoft.AspNetCore.Authentication.JwtBearer.dll	2021/10/27 5:52	应用程序扩展	40 KB
Microsoft.AspNetCore.JsonPatch.dll	2021/7/15 18:09	应用程序扩展	53 KB
Microsoft.AspNetCore.Mvc.NewtonsoftJson.dll	2021/7/15 18:11	应用程序扩展	62 KB
Microsoft.Data.SqlClient.dll	2021/9/21 7:59	应用程序扩展	350 KB
Microsoft.Data.Sqlite.dll	2021/11/18 7:22	应用程序扩展	167 KB
Microsoft.EntityFrameworkCore.Abstractions.dll	2021/11/18 7:22	应用程序扩展	30 KB

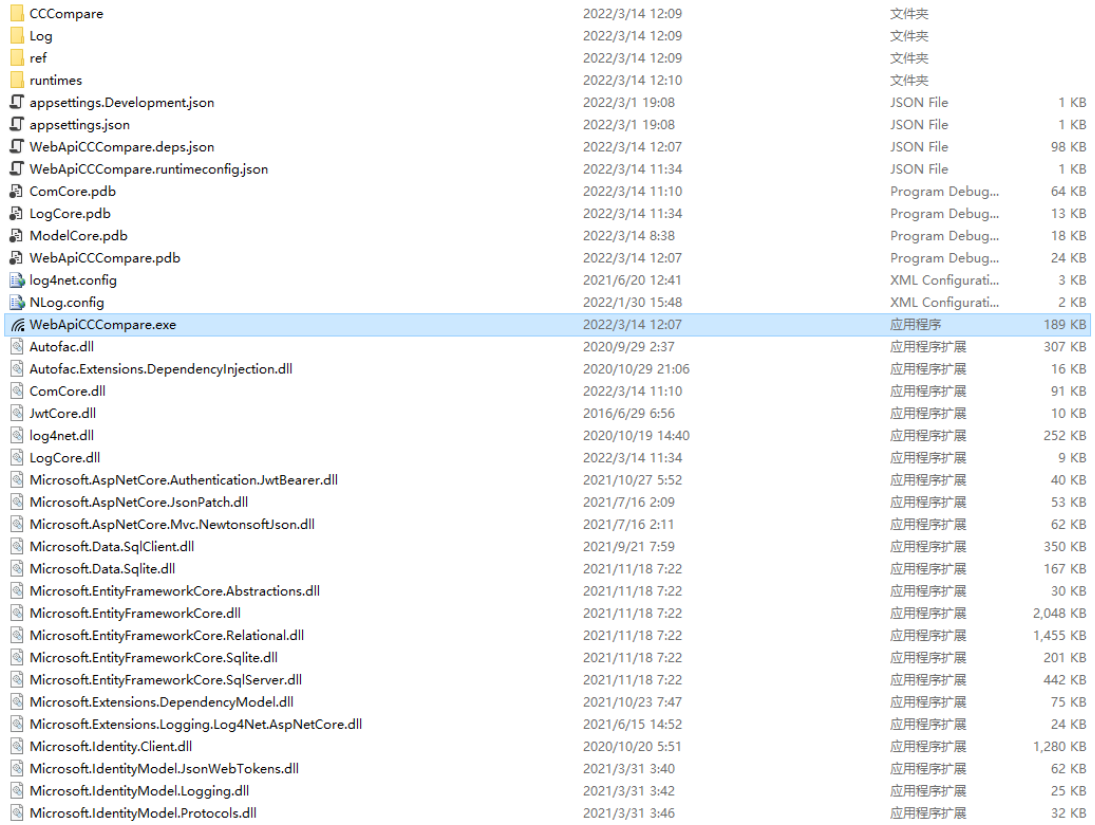
其中，CCCompare 存储系统设置的相关文件，Project 存储项目相关文件，WebApi 存储专家校验的 Web 服务程序，Log 存储日志信息，其他为软件必备的组件。

CCCompare 目录下文件构成如下图所示，分别为系统配置文件，数据库初始化脚本，各种编程语言的注释规则所对应的正则表达式，生成鉴定报告的 HTML 格式模板。其中编程语言的注释规则默认情况下仅提供部分编程语言，用户可自行扩充，也可以委托我司进行扩充；HTML 格式模板也可以根据用户的审美偏好进行修改。



CCCompareConfig.json	2022/3/21 10:37	JSON File	1 KB
Cmd.txt	2022/3/15 13:00	文本文档	2 KB
RemarkRegex.json	2022/3/9 12:00	JSON File	1 KB
TableHtmlModel.txt	2022/3/19 16:28	文本文档	3 KB

WebApi 目录下文件构成如下图所示，在使用专家校验功能时，需双击.exe 文件启动该服务。（专家校验功能支持专家在互联网进行验证，也就是专家不到现场，通过网络查看比对结果，进行比对结果确认，但相关配置需要我司根据用户网络环境进行设置）



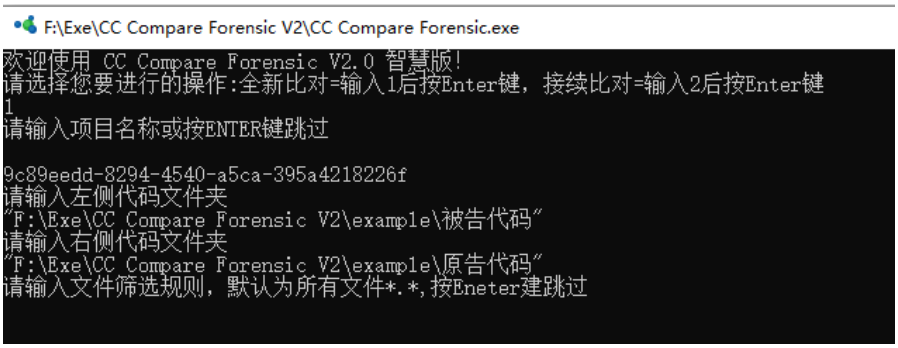
CCCompare	2022/3/14 12:09	文件夹	
Log	2022/3/14 12:09	文件夹	
ref	2022/3/14 12:09	文件夹	
runtimes	2022/3/14 12:10	文件夹	
appsettings.Development.json	2022/3/1 19:08	JSON File	1 KB
appsettings.json	2022/3/1 19:08	JSON File	1 KB
WebApiCCCompare.deps.json	2022/3/14 12:07	JSON File	98 KB
WebApiCCCompare.runtimeconfig.json	2022/3/14 11:34	JSON File	1 KB
ComCore.pdb	2022/3/14 11:10	Program Debug...	64 KB
LogCore.pdb	2022/3/14 11:34	Program Debug...	13 KB
ModelCore.pdb	2022/3/14 8:38	Program Debug...	18 KB
WebApiCCCompare.pdb	2022/3/14 12:07	Program Debug...	24 KB
log4net.config	2021/6/20 12:41	XML Configurati...	3 KB
NLog.config	2022/1/30 15:48	XML Configurati...	2 KB
WebApiCCCompare.exe	2022/3/14 12:07	应用程序	189 KB
Autofac.dll	2020/9/29 2:37	应用程序扩展	307 KB
Autofac.Extensions.DependencyInjection.dll	2020/10/29 21:06	应用程序扩展	16 KB
ComCore.dll	2022/3/14 11:10	应用程序扩展	91 KB
JwtCore.dll	2016/6/29 6:56	应用程序扩展	10 KB
log4net.dll	2020/10/19 14:40	应用程序扩展	252 KB
LogCore.dll	2022/3/14 11:34	应用程序扩展	9 KB
Microsoft.AspNetCore.Authentication.JwtBearer.dll	2021/10/27 5:52	应用程序扩展	40 KB
Microsoft.AspNetCore.JsonPatch.dll	2021/7/16 2:09	应用程序扩展	53 KB
Microsoft.AspNetCore.Mvc.NewtonsoftJson.dll	2021/7/16 2:11	应用程序扩展	62 KB
Microsoft.Data.SqlClient.dll	2021/9/21 7:59	应用程序扩展	350 KB
Microsoft.Data.Sqlite.dll	2021/11/18 7:22	应用程序扩展	167 KB
Microsoft.EntityFrameworkCore.Abstractions.dll	2021/11/18 7:22	应用程序扩展	30 KB
Microsoft.EntityFrameworkCore.dll	2021/11/18 7:22	应用程序扩展	2,048 KB
Microsoft.EntityFrameworkCore.Relational.dll	2021/11/18 7:22	应用程序扩展	1,455 KB
Microsoft.EntityFrameworkCore.Sqlite.dll	2021/11/18 7:22	应用程序扩展	201 KB
Microsoft.EntityFrameworkCore.SqlServer.dll	2021/11/18 7:22	应用程序扩展	442 KB
Microsoft.Extensions.DependencyModel.dll	2021/10/23 7:47	应用程序扩展	75 KB
Microsoft.Extensions.Logging.Log4Net.AspNetCore.dll	2021/6/15 14:52	应用程序扩展	24 KB
Microsoft.Identity.Client.dll	2020/10/20 5:51	应用程序扩展	1,280 KB
Microsoft.IdentityModel.JsonWebTokens.dll	2021/3/31 3:40	应用程序扩展	62 KB
Microsoft.IdentityModel.Logging.dll	2021/3/31 3:42	应用程序扩展	25 KB
Microsoft.IdentityModel.Protocols.dll	2021/3/31 3:46	应用程序扩展	32 KB

第四章：系统使用

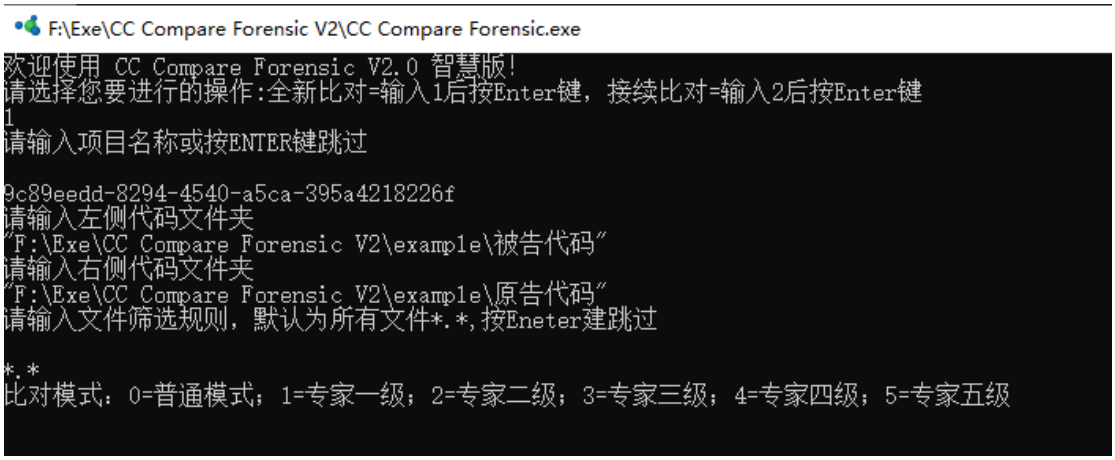
3.1 全新比对

系统为绿色版，无需安装，解压缩后点击 CC Compare Forensic.exe 进入运行界面，授权用户会显示如下截图内容，非授权用户提示用户进行授权。

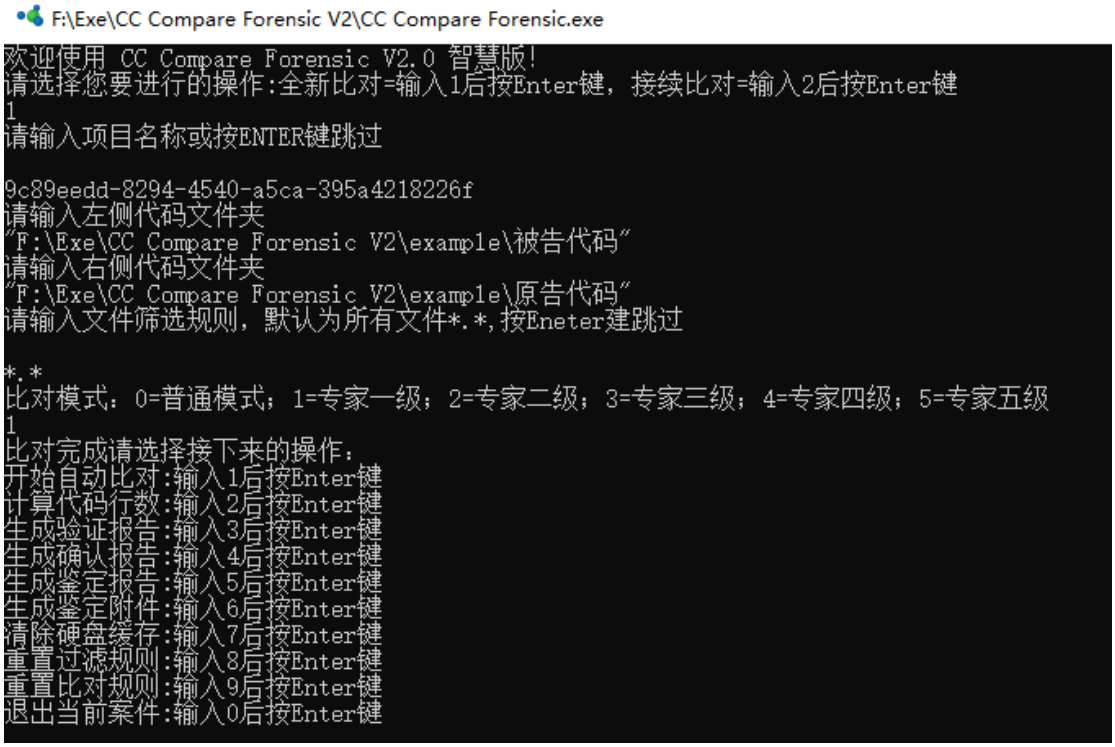
全新比对是指一个全新的检测鉴定案件，接续比对是指上一次没有比对完，本次接续比对的案件。如输入 1 后按回车键为全新比对，然后，输入项目名称，如下图：



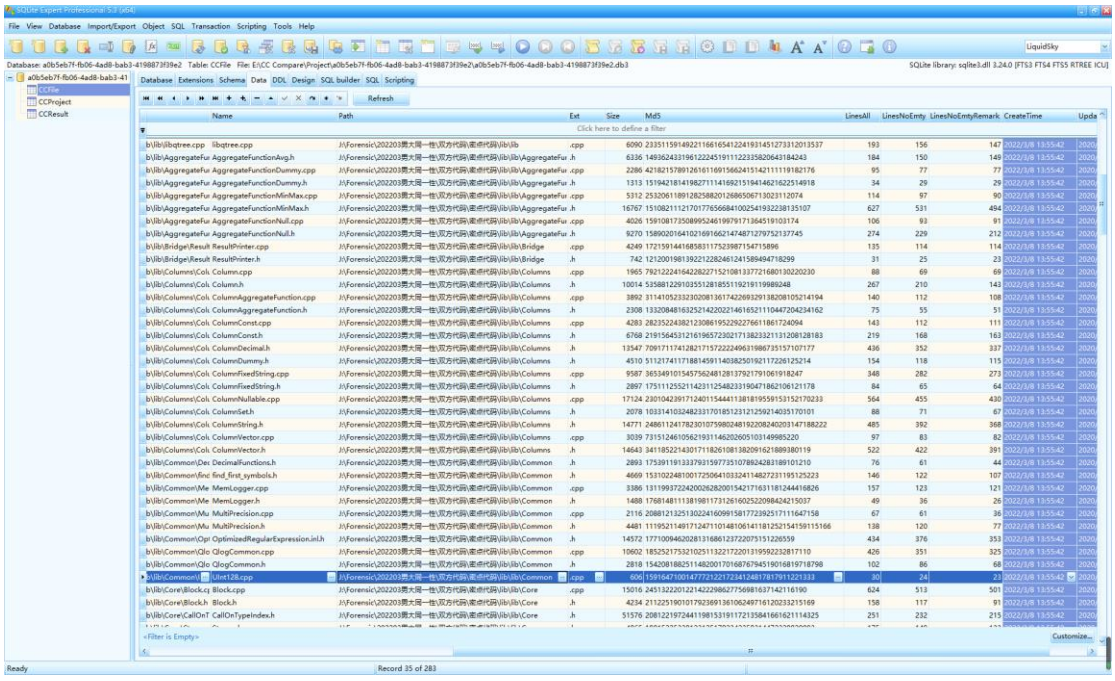
输入待比对的代码文件位置，用鼠标拖拽代码文件夹到屏幕，并输入比对文件的类型，如下图：

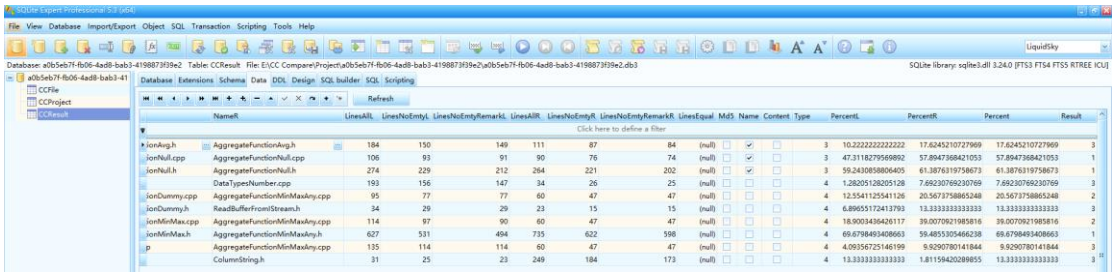


进入比对模式选项，一般情况下普通模式即可满足绝大部分案件的比对要求，专家模式的五个等级，级数越高，比对更加细致，但是时间相对于普通模式更长。选择比对模式后，开始比对。比对完成后，提示用户进行下一步操作。如下图：



选择 1 开始比对，查看比对结果，打开 project 下的项目数据库文件，如下图：





数据库文件为 Sqlite 格式，三个表分别存储的是所有待比对文件列表 CCFile，比对结果列表 CCResult，项目基础信息 CCProject。

选择 2 计算代码文件行数，该步骤可跳过。

选择 3 生成验证报告，报告存储在项目文件夹，下图分别为校验报告的相似度分布表和每对文件的具体比对结果。其中比对结论为计算机自动计算，如需要修订，点击后面最后一列的比对结论即可，相似度分布表和每对文件之间通过超链接进行导航，用浏览器打开。

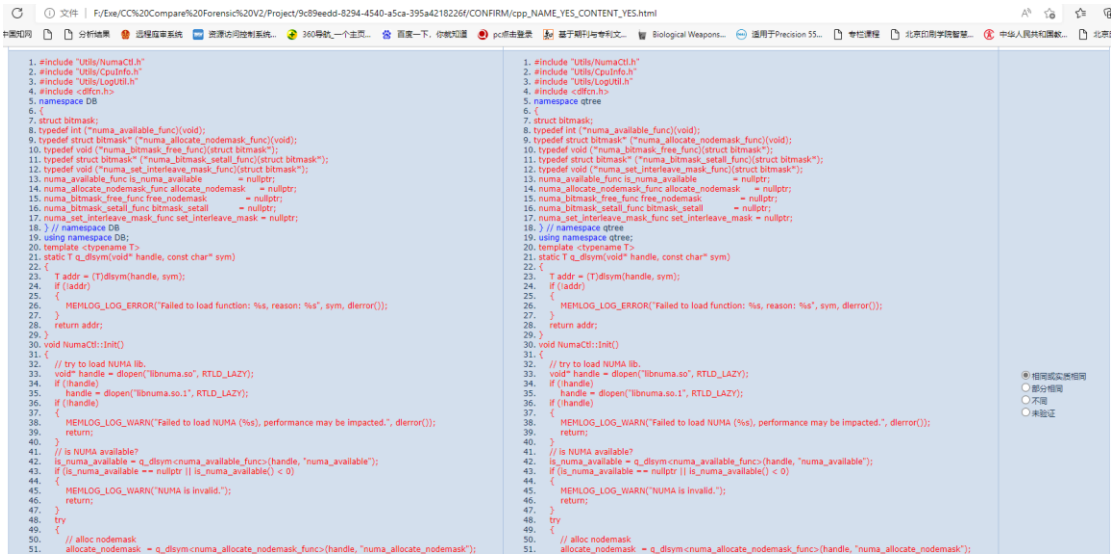
MD5值	文件名	相似度 (%)	文件数量
相同	相同	100	2
相同	相同	100	1
不同	相同	[90,100)	68
不同	相同	[80,90)	8
不同	相同	[70,80)	3
不同	相同	[60,70)	3
不同	相同	[50,60)	3
不同	相同	[40,50)	2
不同	相同	[30,40)	2
不同	相同	[20,30)	4
不同	相同	[10,20)	7
不同	相同	[0,10)	9
不同	不同	[60,70)	1
不同	不同	[30,40)	1
不同	不同	[10,20)	1
不同	不同	[0,10)	4

上图中红色代码为完全相同的代码行，深蓝色为相同代码，但是代码分散在不同行，浅蓝色为不同代码。

选择 4 生成确认报告，报告存储在项目文件夹，如下图所示。确认报告与验证报告不同之处在于按照比对结论进行导航，而不是按照相似度进行导航。

文件类型	左侧文件数量	右侧文件数量	同名文件数量	相同实质相同文件数量	部分相同文件数量	同名但内容不同文件数量
.h	69	94	64	53(同名52,不同名1)	3(同名3,不同名0)	9
.dll	3	1	0	0(同名0,不同名0)	0(同名0,不同名0)	0
.bmp	1	0	0	0(同名0,不同名0)	0(同名0,不同名0)	0
.png	1	0	0	0(同名0,不同名0)	0(同名0,不同名0)	0
.cpp	50	71	48	38(同名38,不同名0)	4(同名3,不同名1)	7
.license	0	1	0	0(同名0,不同名0)	0(同名0,不同名0)	0
.exe	0	1	0	0(同名0,不同名0)	0(同名0,不同名0)	0

比对结论的修改方式与验证报告相同。

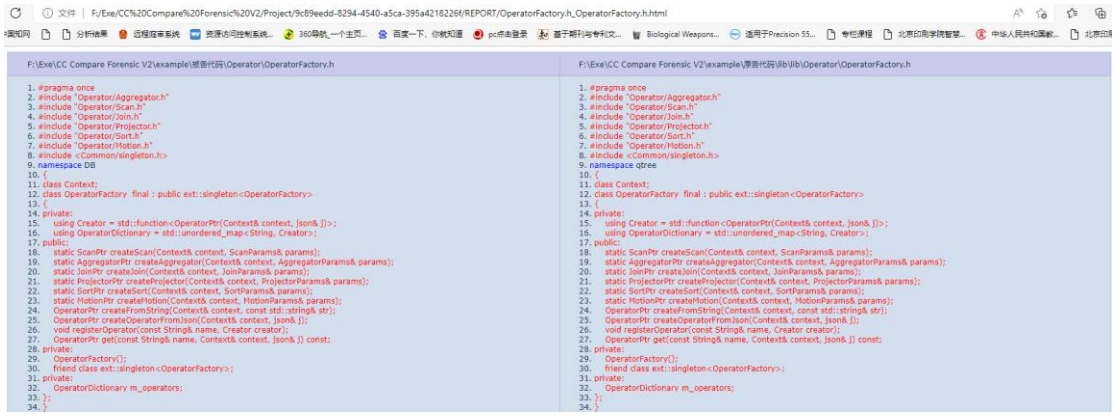


选择 5 生成鉴定报告，鉴定报告是在确认报告的基础上，去掉了专家确认的操作，可以

直接写进鉴定报告，如下图所示。分别为比对结论表，比对文件对应关系表，每对文件具体比对结果。

文件类型	左侧文件数量	右侧方文件数量	同名文件数量	相同实质相同文件数量	部分相同文件数量	同名但内容不同文件数量
.h	69	94	64	53(同名52,不同名1)	3(同名3,不同名0)	9
.dll	3	1	0	0(同名0,不同名0)	0(同名0,不同名0)	0
.bmp	1	0	0	0(同名0,不同名0)	0(同名0,不同名0)	0
.png	1	0	0	0(同名0,不同名0)	0(同名0,不同名0)	0
.cpp	50	71	48	38(同名38,不同名0)	4(同名3,不同名1)	7
.license	0	1	0	0(同名0,不同名0)	0(同名0,不同名0)	0
.exe	0	1	0	0(同名0,不同名0)	0(同名0,不同名0)	0
ALL	124	168	112	91(同名90,不同名1)	7(同名6,不同名1)	16

序号	左侧文件	右侧文件	左侧文件行数（去除空行）	左侧文件行数（去除空行和注释行）	右侧文件行数（去除空行）	右侧文件行数（去除空行和注释行）	相同行数
1	F:\Exe\CC Compare Forensic V2\example\原告代码\Utils\MemoryWrap.h	F:\Exe\CC Compare Forensic V2\example\原告代码\lib\lib\Utils\MemoryWrap.h	53	46	53	46	53
2	F:\Exe\CC Compare Forensic V2\example\原告代码\Server\ConnectionHandler.h	F:\Exe\CC Compare Forensic V2\example\原告代码\lib\lib\Server\ConnectionHandler.h	38	38	38	38	37
3	F:\Exe\CC Compare Forensic V2\example\原告代码\Utils\MPSCQueue.h	F:\Exe\CC Compare Forensic V2\example\原告代码\lib\lib\Utils\MPSCQueue.h	111	93	111	93	109
4	F:\Exe\CC Compare Forensic V2\example\原告代码\Utils\MPSCQueueRaw.h	F:\Exe\CC Compare Forensic V2\example\原告代码\lib\lib\Utils\MPSCQueueRaw.h	109	91	109	91	107
5	F:\Exe\CC Compare Forensic V2\example\原告代码\Operator\Sort\MergeSortingBlocksBlockInputStream.h	F:\Exe\CC Compare Forensic V2\example\原告代码\lib\lib\Operator\Sort\MergeSortingBlocksBlockInputStream.h	48	32	48	32	47
6	F:\Exe\CC Compare Forensic V2\example\原告代码\Network\Protocol.h	F:\Exe\CC Compare Forensic V2\example\原告代码\lib\lib\Network\Protocol.h	81	80	81	80	80



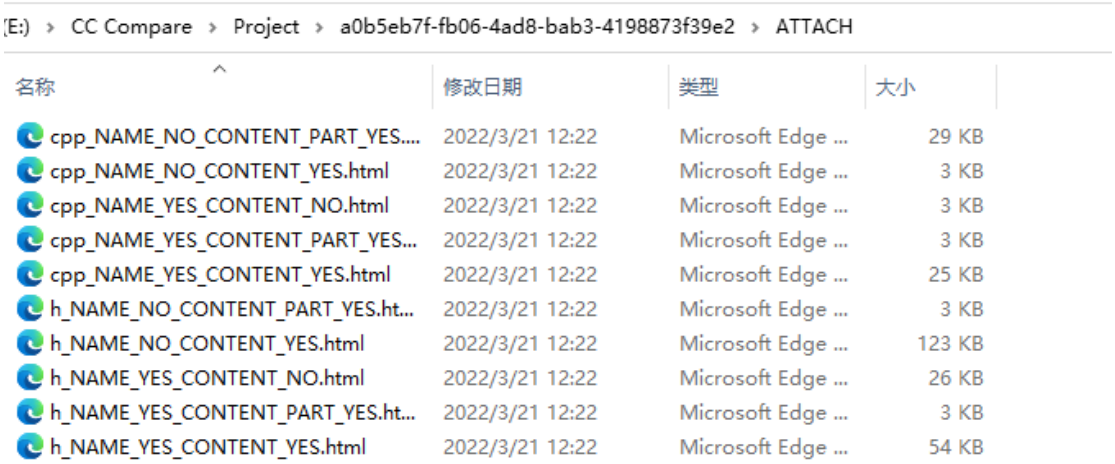
F:\Exe\CC Compare Forensic V2\example\原告代码\Operator\OperatorFactory.h

```
1. #pragma once
2. #include "Operator/Aggregator.h"
3. #include "Operator/Scan.h"
4. #include "Operator/Join.h"
5. #include "Operator/Projector.h"
6. #include "Operator/Sort.h"
7. #include "Operator/Motion.h"
8. #include <Common/singleton.h>
9. namespace DB
10. {
11. class Context;
12. class OperatorFactory  final : public ext::singleton<OperatorFactory>
13. {
14. private:
15.     using Creator = std::function<OperatorPtr(Context& context, json& )>;
16.     using OperatorDictionary = std::unordered_map<String, Creator>;
17. public:
18.     static ScanPtr createScan(Context& context, ScanParams& params);
19.     static AggregatorPtr createAggregator(Context& context, AggregatorParams& params);
20.     static JoinPtr createJoin(Context& context, JoinParams& params);
21.     static ProjectorPtr createProjector(Context& context, ProjectorParams& params);
22.     static SortPtr createSort(Context& context, SortParams& params);
23.     static MotionPtr createMotion(Context& context, MotionParams& params);
24.     OperatorPtr createFromString(Context& context, const std::string& str);
25.     OperatorPtr createOperatorFromJson(Context& context, json& );
26.     void registerOperator(const String& name, Creator creator);
27.     OperatorPtr get(const String& name, Context& context, json& ) const;
28. private:
29.     OperatorFactory();
30.     friend class ext::singleton<OperatorFactory>;
31. private:
32.     OperatorDictionary m_operators;
33. };
34. }
```

F:\Exe\CC Compare Forensic V2\example\原告代码\lib\lib\Operator\OperatorFactory.h

```
1. #pragma once
2. #include "Operator/Aggregator.h"
3. #include "Operator/Scan.h"
4. #include "Operator/Join.h"
5. #include "Operator/Projector.h"
6. #include "Operator/Sort.h"
7. #include "Operator/Motion.h"
8. #include <Common/singleton.h>
9. namespace qtree
10. {
11. class Context;
12. class OperatorFactory  final : public ext::singleton<OperatorFactory>
13. {
14. private:
15.     using Creator = std::function<OperatorPtr(Context& context, json& )>;
16.     using OperatorDictionary = std::unordered_map<String, Creator>;
17. public:
18.     static ScanPtr createScan(Context& context, ScanParams& params);
19.     static AggregatorPtr createAggregator(Context& context, AggregatorParams& params);
20.     static JoinPtr createJoin(Context& context, JoinParams& params);
21.     static ProjectorPtr createProjector(Context& context, ProjectorParams& params);
22.     static SortPtr createSort(Context& context, SortParams& params);
23.     static MotionPtr createMotion(Context& context, MotionParams& params);
24.     OperatorPtr createFromString(Context& context, const std::string& str);
25.     OperatorPtr createOperatorFromJson(Context& context, json& );
26.     void registerOperator(const String& name, Creator creator);
27.     OperatorPtr get(const String& name, Context& context, json& ) const;
28. private:
29.     OperatorFactory();
30.     friend class ext::singleton<OperatorFactory>;
31. private:
32.     OperatorDictionary m_operators;
33. };
34. }
```

选择 6 生成鉴定附件，鉴定附件是每一对文件比对的详情，存储路径如下图：



(E:) › CC Compare › Project › a0b5eb7f-fb06-4ad8-bab3-4198873f39e2 › ATTACH

名称	修改日期	类型	大小
cpp_NAME_NO_CONTENT_PART_YES....	2022/3/21 12:22	Microsoft Edge ...	29 KB
cpp_NAME_NO_CONTENT_YES.html	2022/3/21 12:22	Microsoft Edge ...	3 KB
cpp_NAME_YES_CONTENT_NO.html	2022/3/21 12:22	Microsoft Edge ...	3 KB
cpp_NAME_YES_CONTENT_PART_YES...	2022/3/21 12:22	Microsoft Edge ...	3 KB
cpp_NAME_YES_CONTENT_YES.html	2022/3/21 12:22	Microsoft Edge ...	25 KB
h_NAME_NO_CONTENT_PART_YES.ht...	2022/3/21 12:22	Microsoft Edge ...	3 KB
h_NAME_NO_CONTENT_YES.html	2022/3/21 12:22	Microsoft Edge ...	123 KB
h_NAME_YES_CONTENT_NO.html	2022/3/21 12:22	Microsoft Edge ...	26 KB
h_NAME_YES_CONTENT_PART_YES.ht...	2022/3/21 12:22	Microsoft Edge ...	3 KB
h_NAME_YES_CONTENT_YES.html	2022/3/21 12:22	Microsoft Edge ...	54 KB

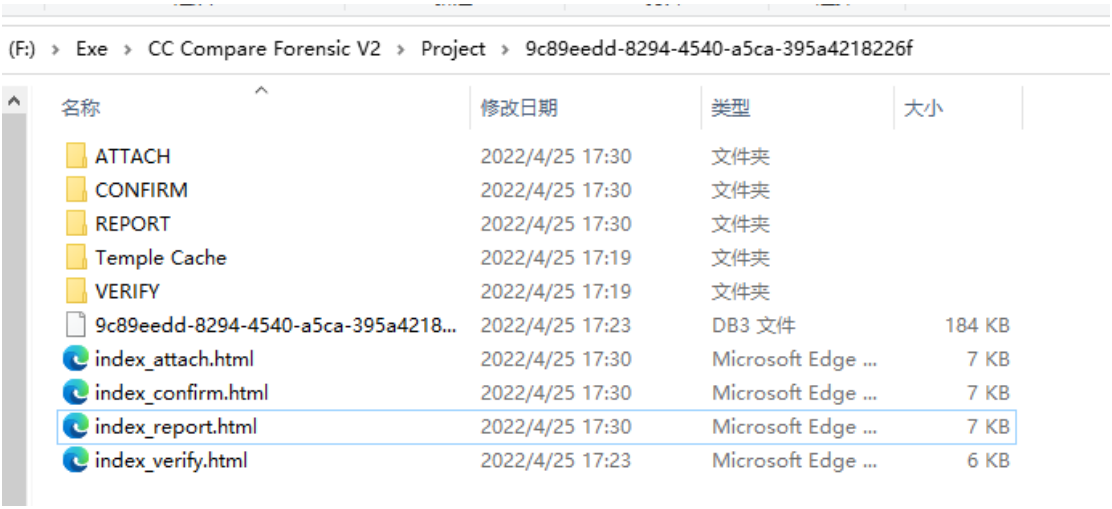
最后，本次比对的所有文件存储如下图所示，各个文件含义分别为：index_verify.html 为专家校验报告， index_confirm.html 为专家确认报告， index_report.html 为鉴定报告，

index_attach.html 为鉴定附件；文件夹 VERIFY、CONFIRM、REPORT、ATTACH 分别为各类报告的具体比对结果；XXXX.db3 为项目数据库，也就是项目的主体核心文件；Temple Cache 为比对过程中为了加快比对的临时硬盘缓存文件。如此操作完成一个全新案件的比对、检测、分析和报告过程。

选择 7 清除缓存，缓存可加快比对速度，除硬盘空间不足外，不建议清除缓存。

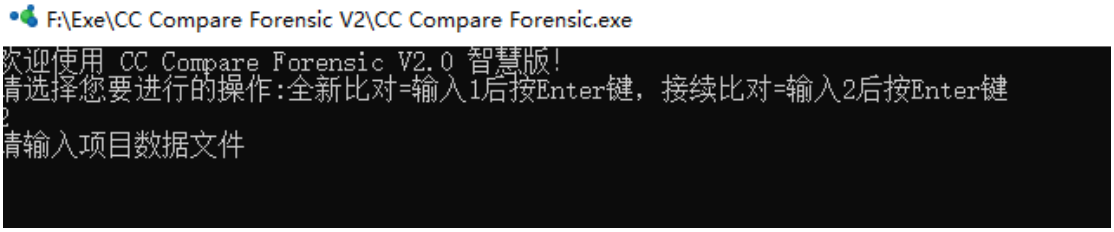
选择 8 重置过滤规则，比如第一次比对*.c 格式代码，第二次比对*.h 代码，需要重置过滤规则，当然也可以一次比对两种文件，只需要设置过滤规则为*.c|*.h 即可。

选择 9 重置比对规则，比如，采用普通模式比对后，认为比对效果不好，可更改为专家模式重新进行比对。



3.2 接续比对

系统启动后，选择 2 进入接续比对，如下图所示：项目数据是全新比对中的 XXX.db3 数据库文件，将其拖拽到屏幕即可，后续操作与全新比对相同。



第五章：系统设置

CCCompare 下的 CCCompareConfig.json 文件为系统设置文件，具体内容如下图所示。其中：

5.1 并行化比对设置

IdStart、IdEnd 为比对文件的 id 范围，默认 0-100000000，表示全部比对，当一个项目

在多台计算机进行比对时，通过该参数实现多台计算机并行比对。

5.2 代码混淆比对设置

ABSSwitch 为语法分析器开关，用于混淆代码比对，开启时比对更准确，但是更耗时，建议不开启该开关。

5.3 代码编码设置

CodeEncodingString 为涉案代码的编码格式，默认为 gb2312。

5.4 其他比对设置

SimilarityAlpha 为比对的行单元，默认为 2 行。

SimilarityDegree 为相似度的阈值，用于区分不同分析结论与相似度关系。

CompareLevel 为比对等级，普通、专家一到五级。

ABSStringSpliterExpertl-ABSStringSpliterExpert5 为专家 1-5 级的比对粒度参数设置，也就是划分比对单元的分隔符，专家级别越高，分隔符种类越多。

AbsTxtEx 为默认的非源代码文件，这些文件不参与比对，即使用户选择了*.*过滤规则，这些文件也不参与内容比对，仅进行 MD5 值的比对。

ABSMaxLine 为代码文件的最大行数。

SimilarityDegree 为划分代码相同/实质相同、部分相同、不同的相似度阈值。

```
{
  "IdStart": "0",

  "IdEnd": "1000000000",

  "ABSSwitch": false,

  "CodeEncodingString": "gb2312",

  "AbsTxtEx": [
    ".exe",
    ".dll",
    ".so",
    ".baml",
    ".bin",
    ".png",
    ".jpg",
    ".bpm",
    ".ico",
    ".cache",
    ".zip",
    ".rar",
    ".iso",
    ".pdf",
    ".doc",
    ".docx",
    ".xls",
    ".xlsx",
    ".ppt",
    ".pptx"
  ],

  "ABSMaxLine": "10000",

  "SimilarityAlpha": "2",

  "SimilarityDegree": [ "0.4", "0.2", "0.0" ],

  "CompareLevel": "Normal",

  "ABSStringSpliterExpert1": [],

  "ABSStringSpliterExpert2": [ " " ],

  "ABSStringSpliterExpert3": [ " ", "(", ")", "[", "]" ],

  "ABSStringSpliterExpert4": [ " ", "(", ")", "[", "]", "/", "/*", "*/", "::", "&", "!" ],

  "ABSStringSpliterExpert5": [
    " ",
    "::",
    "+",
    "-",
    "*",
    "/",
    "=",
    "!",
    "&",
    "|"
```


第六章：比对原理

6.1 确定代码对应关系

1、根据双方代码文件 MD5 值，筛选出完全相同的代码文件对儿。

2、对于 MD5 值不同的代码文件，根据文件名筛选出名称相同的代码文件对儿。

3、对于没有相同名称的代码文件，根据代码内容筛选出最相似的代码文件对儿。

6.2 代码内容比对

1、MD5 值相同的文件对儿，完全相同。

2、MD5 值不同的文件对儿，进行内容比对：普通模式，以代码行为单位进行顺序比对；专家 1 级，以代码行为单位进行交叉比对；专家 2-5 级，以单词、字符为单位进行交叉比对，级数越高，单词和字符划分的越细致；代码和字符比对是多对多的比对，即代码 A 中的单词或字符，会找出代码 B 中所有的该单词或字符进行匹配，采用蓝色进行显示。

6.3 比对算法

比对算法为我司根据历史鉴定案件设计的独有的比对算法，未向社会公开。

6.4 免责声明

免责声明：本软件比对结果不作为任何公检法等国家司法行政机关采信的必然结果，请用户结合相关法律法规、专家知识综合判断。