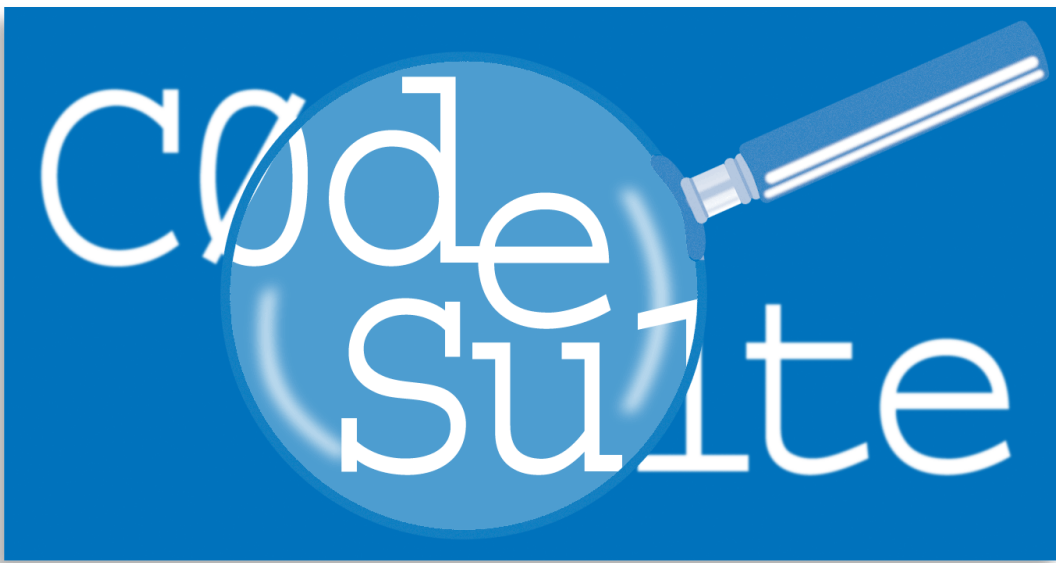


## User's Guide

---



**CodeSuite-AC Version 1.6**

a product of

**S.A.F.E.**

**Software Analysis & Forensic Engineering  
Corporation**



## Table of Contents

CodeSuite-AC .....	1
Copyrights, Trademarks, Patents .....	3
Using CodeSuite-AC .....	4
System Requirements .....	4
Licenses .....	5
The Toolbar .....	7
Comparing Source Code Files .....	9
Compare Code .....	9
CodeMatch Algorithms .....	12
FileCount.....	25
Running FileCount.....	25
FileIdentify.....	26
Running FileIdentify.....	26
Languages .....	28
Languages Supported .....	28
Contacting SAFE Corporation.....	29
Contacting SAFE Corporation .....	29
Index .....	30



# CodeSuite-AC

CodeSuite-AC® is the academic version of the CodeSuite® collection of computer code analysis tools. The individual tools that comprise the suite of tools include CodeMatch®, CodeSplit®, SourceDetective®, FileCount™, and FileIdentify™, all of which are described below.



CodeMatch compares thousands of source code files in multiple directories and subdirectories to determine which files are the most highly correlated. This can be used to significantly speed up the work of finding source code plagiarism, because it can direct the examiner to look closely at a small amount of code in a handful of files rather than thousands of combinations. CodeMatch is also useful for finding open source code within proprietary code, determining common authorship of two different programs, and discovering common, standard algorithms within different programs.

CodeMatch compares every file in one directory with every file in another directory, including all subdirectories if requested. CodeMatch produces an HTML basic report that lists the most highly correlated pairs of files. You can click on any particular pair listed in the HTML basic report see an HTML detailed report that shows the specific items in the files (statements, comments, strings, identifiers, or instruction sequences) that caused the high correlation.

CodeMatch uses unique algorithms to find various different ways that source code files are correlated. These algorithms can find directly copied source code and even source code that has been modified to avoid detection.



CodeSplit takes all source code files in multiple directories and subdirectories and splits them into basic elements (statements, comments, strings, and identifiers). The resulting database can be used in conjunction with SourceDetective to find Internet evidence that the source code was derived from third-party code found on the Internet.



# SourceDetective

SourceDetective is a utility that searches the Internet for all references to matching statements, comments, and identifiers found in a CodeSuite database. SourceDetective is used to determine whether statements, comments, and identifiers found in two sets of files are commonly used or not, depending on how many references can be found on the Internet.



# FileCount

FileCount is a utility that counts the number of files, non-blank lines, and bytes in a large set of files in a directory tree. FileCount is useful when using CodeDiff to generate statistics about a set of source code files.



# FileIdentify

FileIdentify is a utility that examines all of the file types in a given directory, or an entire directory tree, and reports the associated programming languages if known.

# **Copyrights, Trademarks, Patents**

## **Copyrights**

The materials in this user's guide are copyright 2005-2025 by Software Analysis and Forensic Engineering Corporation.

All written materials from SAFE Corporation regarding CodeSuite, BitMatch, CodeCLOC, CodeCross, CodeDiff, CodeMatch, CodeSplit, FileCount, FileIdentify, FileIsolate, and SourceDetective, including the material in this User's Guide and the source code for all versions of CodeSuite, BitMatch, CodeCLOC, CodeCross, CodeDiff, CodeMatch, CodeSplit, FileCount, FileIdentify, FileIsolate, and SourceDetective are the copyright of SAFE Corporation.

## **Trademarks**

SAFE Corporation, the SAFE Corporation logo, the SAFE Corporation brand, CodeSuite, the CodeSuite logo, BitMatch, CodeCLOC, CodeCross, CodeDiff, CodeMatch, CodeSplit, FileCount, FileIdentify, FileIsolate, SourceDetective, and all other SAFE Corporation product names referenced herein are registered trademarks or trademarks of SAFE Corporation. All other brand and product names mentioned herein are trademarks of their respective owners.

## **Patents**

CodeSuite-AC is covered by U.S. patents 7,503,035, 7,823,127, 8,255,885, 8,261,237, 8,495,586, 9,003,366, 9,043,375, and 9,053,296.

# Using CodeSuite-AC

## System Requirements

CodeSuite-AC will run on any computer using any of the following versions of the Microsoft Windows operating system:

- Windows VistaWindows 7
- Windows 8
- Windows 10
- Windows 11

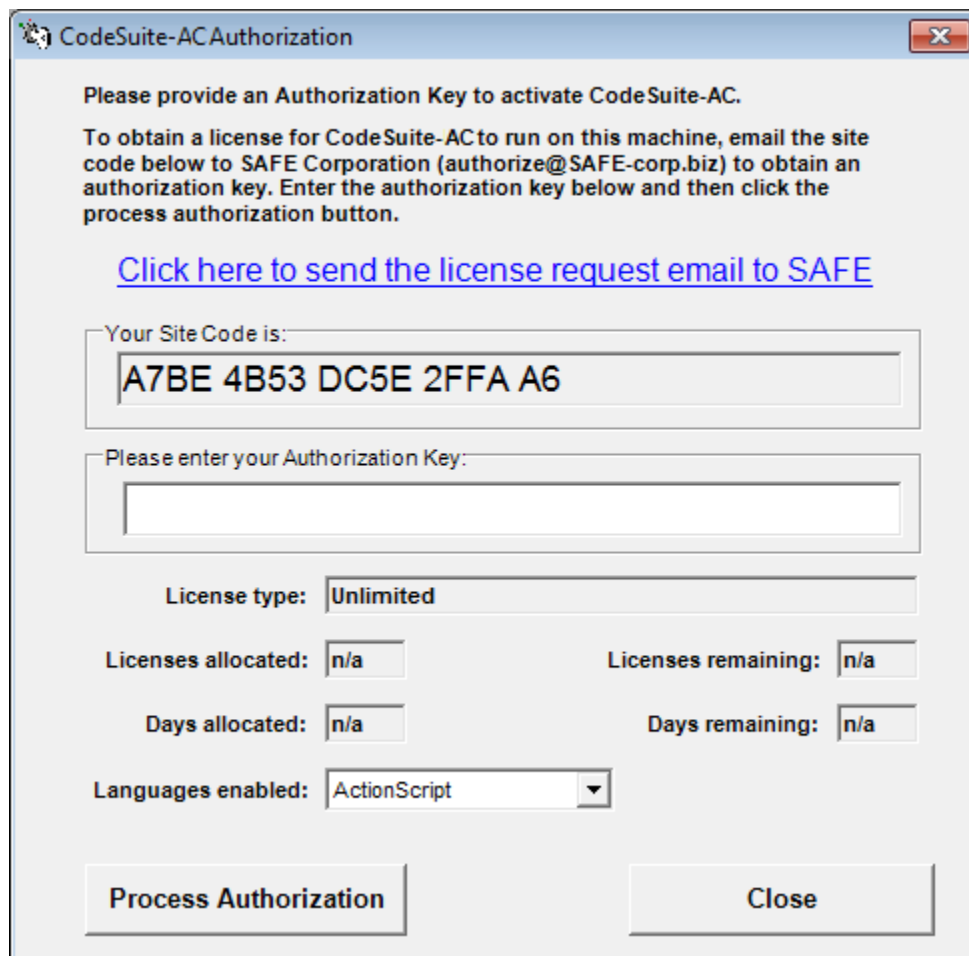
Note that CodeSuite-AC will not run on a virtual system and may not run on some systems using a remote desktop.



## Licenses

Licenses must be purchased from SAFE Corporation. The FileCount and FileIdentify functions of CodeSuite-AC do not require a license.

To request licenses, open the authorization form shown below from the Help menu. Send the site code to SAFE Corporation and the number of licenses requested, along with appropriate payment. SAFE Corporation will send back an Authorization Key that must be entered into the field in the form. Press the process authorization button and the form will show the following information. Licenses are enabled for only one PC and cannot be transferred to another PC.

The image shows a Windows-style dialog box titled "CodeSuite-AC Authorization". It contains instructions for obtaining a license, a link to request one, a text field for the site code (A7BE 4B53 DC5E 2FFA A6), a text field for the authorization key, and several fields for license details: License type (Unlimited), Licenses allocated (n/a), Licenses remaining (n/a), Days allocated (n/a), Days remaining (n/a), and Languages enabled (ActionScript). At the bottom are "Process Authorization" and "Close" buttons.

**CodeSuite-AC Authorization**

Please provide an Authorization Key to activate CodeSuite-AC.

To obtain a license for CodeSuite-AC to run on this machine, email the site code below to SAFE Corporation (authorize@SAFE-corp.biz) to obtain an authorization key. Enter the authorization key below and then click the process authorization button.

[Click here to send the license request email to SAFE](#)

Your Site Code is:

A7BE 4B53 DC5E 2FFA A6

Please enter your Authorization Key:

License type: Unlimited

Licenses allocated: n/a      Licenses remaining: n/a

Days allocated: n/a      Days remaining: n/a

Languages enabled: ActionScript

Process Authorization      Close

## License Type

The license can be one of three types.

- **File size based.** Used to examine a fixed amount of bytes of source code. Licenses are used up as source code is examined. SourceDetective searches of the Internet also use up licenses.
- **Time based.** Used to examine any amount of code for a fixed number of days. Note that there is still a limit to the number of SourceDetective searches of the Internet that can be performed. If that limit is reached, no more searching can be done for the remainder of the license term unless a new license is purchased.
- **Unlimited.** There is no limit on the number of megabytes that can be examined and there is no expiration date.

## **Licenses Allocated and Licenses Remaining**

These fields indicate the number of licenses that were originally allocated and how many unused licenses remain. These fields are valid only for a megabyte-based license. For other licenses, the fields are not applicable ("n/a").

## **Days Allocated and Days Remaining**

These fields indicate the number of days that were originally allocated for the license and how many days remain on the license. These fields are valid only for a time-based license. For other licenses, the fields are not applicable ("n/a").

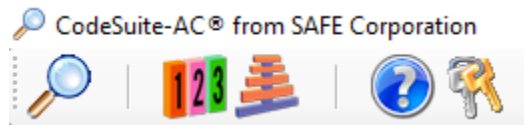
## **Languages Enabled**

This pulldown list shows all of the programming languages that are enabled for analysis by the license.

See the SAFE Corporation website for license costs, as they may change.

## The Toolbar

The CodeSuite-AC toolbar is shown below.



**Compare Code**

This menu selection brings up the Compare Code form, the main function of CodeSuite-AC. See the section entitled Compare Code for more information.



**FileCount**

This menu selection brings up the FileCount form. See the section entitled Running FileCount for more information.



**FileIdentify**

This menu selection brings up the FileIdentify form. See the section entitled Running FileIdentify for more information.



**Help**

This menu selection brings up this user's guide.



### **Authorize**

This menu selection brings up the authorization form for entering licenses to enable the various tools. See the section entitled Licenses for more information.

# Comparing Source Code Files

## Compare Code

Below is a screen shot of the form to compare source code files. Following that are step-by-step instructions.

**Compare Code**

**Code folder:** Z:\SAFE Corporation\code development\test\CodeSuite-AC\files 1 Browse...

☒ Include subdirectories Language: C File type: \*.c;\*.h

☒ Use templates

**Templates folder:** Z:\SAFE Corporation\code development\test\CodeSuite-AC\templates Browse...

☒ Include subdirectories

**Options:**

☒ CodeMatch: Compare files

- ☒ Filter out common elements
- ☐ Compare sequences
- ☐ Compare partial identifiers

☐ CodeSplit: Find code on websites

**GO**

### Step 1

Select the folder containing source code files for comparison by clicking on the browse button or entering the path in the text field. Check the box to include files in all subdirectories, if desired.

### Step 2

Select a source code language from the pulldown menu.

### Step 3

Select the files types to compare from the pulldown menu. You can type over the suggested file types with your own file types. Separate multiple file types with a semicolon. Use the \* and ? wildcard characters if needed.

### Step 4

Check the box to specify templates, which are source code files that contain program elements that may be common to all of the programs and should thus be ignored for the comparison.

### Step 5

Select the folder containing source code template files by clicking on the browse button or entering the path in the text field. Check the box to include files in all subdirectories, if desired.

### Step 6

Check the CodeMatch box to use CodeMatch® to compare all the source code files in the folder to each other. When this box is checked, two other boxes will appear:

- Check the first box to use SourceDetective® to search the Internet and filter out all matching code elements in the files that are commonly used by other programmers.
- Check the second box to include a comparison of instruction sequences. This option shows cases where code may have been copied but significantly modified, but the sequences of instructions (the basic functionality) are correlated.
- Check the third box to include a comparison of partially matching identifiers when comparing source code files. This option will allow you to see where identifier names have been copied and modified, but will require more manual effort to review the results.

### Step 7

Check the CodeSplit box to use CodeSplit® to split the source code files into basic code elements and then run SourceDetective to search the Internet to find websites with these code elements. The report will show all source code files and URLs where the code elements can be found. This is useful for finding code that was copied from the Internet.

### Step 8

Click on the GO button. The number of licenses, if any, that are required for this run of CodeSuite will be shown. You will have the ability to cancel the CodeSuite run at this point without using up licenses.

You will be then asked for the names of the HTML report files to be created.

### Resulting HTML reports

After the comparison, HTML reports will be generated. The CodeMatch basic report shows file pairs and their correlation scores. By clicking on a score, a detailed HTML report will come up for that file pair. These detailed reports are kept in subfolders. The detailed reports give more information about how the score was determined, showing specific similarities or differences between the files. The file names are given at the top of the report and include hyperlinks that, when clicked, allow the file to be brought up in a viewer or editor. The back and next buttons on the detailed reports allow you to navigate the detailed reports without going back to the basic report.

The CodeSplit basic report includes the top websites and URLs where code elements were found online for each source code file. This can help determine whether code was copied from an online source.

For examples of the reports, see the sections entitled CodeMatch Basic Report, CodeMatch Detailed Report, and CodeSplit Basic Report.

# CodeMatch Algorithms

## The Algorithms

CodeMatch uses several algorithms to determine similarity between two source code files. These algorithms are described below. When multiple files are compared, each match is given a weight and all weights are combined into a single matching score called the correlation score. The file pairs are then ranked by correlation score so that you can examine the most similar files.

### Statement matching

CodeMatch looks for identical program statements (i.e., functional source code), ignoring whitespace and eliminating comments and strings. Statements that contain only programming language keywords are not considered matching. For statements to be considered matches, they must contain at least one identifier (non-keyword) such as a variable name or function name.

### Comment/string matching

CodeMatch looks for identical comments and strings, ignoring whitespace. Comment lines and strings that contain only programming language keywords are still considered matches.

### Instruction sequence matching

CodeMatch looks for sequences of instructions that match. CodeMatch notes the longest such sequence in each pair of files. A sequence matches if the initial programming language statement on each line is identical, regardless of what follows it. Even if variable names are altered in one file, CodeMatch will report similarities in the files. The following shows an example of two identical instruction sequences in C:

```
// File 1
if (x == 5)
{
    // Loop on j here
    for (j = 0; j < Index; j++)
        printf("x = %i", j);
}
else
    break; // Here's the break

// File 2
if (xyz < 2)
    for (jjj = 0; jjj < i; jjj++)
    {
```



```
        printf("Hello world\n");  
    }  
    else  
        break;
```

### Identifier matching

CodeMatch finds every instance in each file where identifiers match exactly. It eliminates programming language keywords and only reports matches for non-keyword identifiers such as variable names and function names.

CodeMatch also finds every instance where an identifier in one file is part of a larger identifier in the other file. For example, the variable name "Index" in one file would partially match the variable names "NewIndex" and "Index1" in the other file.

CodeMatch eliminates programming language keywords and only reports matches for non-keyword identifiers such as variable names and function names.

### Correlation Score

CodeMatch produces a total correlation score based on the combination of above algorithms. The minimum score is 0 while the maximum score is 100.

**CodeMatch Basic Report**

Version: 5.7.2 | Date: 07/24/18 | Time: 23:02:59

SETTINGS | RESULTS | UNCOMPARED FILES | URLS | TOTALS

**SETTINGS**

Compare files in folder	Z:\SAFE Corporation\test\sorting\files 1 <i>Not including subdirectories</i>
File types	*.c;*.h
Programming language	C
To files in folder	Z:\SAFE Corporation\test\sorting\files 1 <i>Not including subdirectories</i>
File types	*.c;*.h
Programming language	C
Algorithms selected	<ul style="list-style-type: none"> <li>• Statement Matching</li> <li>• Comment/String Matching</li> <li>• Identifier Matching</li> <li>• Instruction Sequence Matching</li> <li>• List sequences</li> </ul>
Reporting file threshold	4 files

**RESULTS**

Z:\SAFE Corporation\test\sorting\files 1\aaa.c

Z:\SAFE Corporation\test\sorting\files 1\aaa.c

Score	Compared to file
82	Z:\SAFE Corporation\test\sorting\files 1\aaa_case.c
82	Z:\SAFE Corporation\test\sorting\files 1\aaa_whitespace.c

Z:\SAFE Corporation\test\sorting\files 1\aaa\_case.c

Score	Compared to file
82	Z:\SAFE Corporation\test\sorting\files 1\aaa_whitespace.c

Z:\SAFE Corporation\test\sorting\files 1\aaa\_with\_comments.c

Score	Compared to file
77	Z:\SAFE Corporation\test\sorting\files 1\abc_with_comments.c
71	Z:\SAFE Corporation\test\sorting\files 1\aaa_with_comments.c
71	Z:\SAFE Corporation\test\sorting\files 1\aaa_with_comments.c

Z:\SAFE Corporation\test\sorting\files 1\aaa\_whitespace.c

Score	Compared to file
71	Z:\SAFE Corporation\test\sorting\files 1\aaa_with_comments.c
65	Z:\SAFE Corporation\test\sorting\files 1\abc_with_comments.c
65	Z:\SAFE Corporation\test\sorting\files 1\abc_with_comments.c
65	Z:\SAFE Corporation\test\sorting\files 1\abc_with_comments.c
56	Z:\SAFE Corporation\test\sorting\files 1\bpf_image.c
56	Z:\SAFE Corporation\test\sorting\files 1\bpf_image2.c

## URLs

Hits	Website
92	libseccomp/scmp_bpf_disasm.c at master · seccomp ...
88	libpcap/bpf_image.c at master · the-tcpdump-group/libpcap ...

86	libpcap: bpf/net/bpf_filter.c Source File - doxygen ...
60	bpf_filter.c - Apple Inc.
54	bpf(4) - Berkeley Packet Filter - GSP Services
38	Description - man pages section 7: Device and Network ...
17	STRING: functional protein association networks
17	String   Definition of String by Merriam-Webster
17	String Class (System) - msdn.microsoft.com
17	String (Java Platform SE 7 ) - Oracle Help Center

## TOTALS

**Total number of bytes in files in folder 1 = 23685**

**Total run time = 2 Seconds**



CodeSuite copyright 2003-2018 by Software Analysis and Forensic Engineering Corporation

**CodeMatch Detailed Report**

Version: 5.3.1 | Date: 08/28/08 | Time: 11:33:11

**SETTINGS**

Compare file 1:	C:\test\C\files 1\bpf_image.c
To file 2:	C:\test\C\files 2\.svn\bpf_image.c
Links to results:	<a href="#">Matching Statements</a> <a href="#">Matching Comments and Strings</a> <a href="#">Matching Instruction Sequences</a> <a href="#">Matching Identifiers</a> <a href="#">Partially Matching Identifiers</a> <a href="#">Score</a>

**RESULTS**

Matching Statements		
File1 Line#	File2 Line#	Statement
22	22	#include <windows.h>
23	23	#include <sys/types.h>
35	35	char *fmt, *op
36	36	static char image[256]
37	37	char operand[64]
39	39	v = p->k
40	40	switch (p->code) {

199 204 209 214	199	case BPF_ALU BPF_OR BPF_X:
254	254 259 264 269 270	case BPF_ALU BPF_NEG:



### Matching Comments and Strings

File1 Line#	File2 Line#	Comment/String
2	2	* Copyright (c) 1990, 1991, 1992, 1994, 1995, 1996
3	3	* The Regents of the University of California. All rights reserved.
5	5	* Redistribution and use in source and binary forms, with or without
6	6	* modification, are permitted provided that: (1) source code distributions
7	7	* retain the above copyright notice and this paragraph in its entirety, (2)
8	8	* distributions including binary code include the above copyright notice and
9	9	* this paragraph in its entirety in the documentation or other materials
10	10	* provided with the distribution, and (3) all advertising materials mentioning
11	11	* features or use of this software display the following acknowledgement:



### Matching Instruction Sequences

File1 Line#	File2 Line#	Number of matching instructions
22	22	202
43	129	71
46	51	64
46	56	60
46	61	56

46	66	52
46	71	48
46	76	44
46	81	40
46	86	36
46	91	32



### Matching Identifiers

256	64	BPF_A	BPF_ABS	BPF_ADD	BPF_ALU	BPF_AND	BPF_B
BPF_CLASS	BPF_DIV	BPF_H	bpf_image	BPF_IMM	BPF_IND	bpf_insn	BPF_JA
BPF_JEQ	BPF_JGE	BPF_JGT	BPF_JMP	BPF_JSET	BPF_K	BPF_LD	BPF_LDX
BPF_LEN	BPF_LSH	BPF_MEM	BPF_MISC	BPF_MSH	BPF_MUL	BPF_NEG	BPF_OP
BPF_OR	BPF_REX	BPF_RSH	BPF_ST	BPF_STX	BPF_SUB	BPF_TAX	BPF_TXA
BPF_W	BPF_X	code	fmt	image	INT	jf	jt
op	operand	stdio	string	sys	types	windows	



### Partially Matching Identifiers

#### File1 Identifiers

0x00FF	BPF_ALU	bpf_filter	BPF_IMM	BPF_IN	BPF_LEN	BPF_MEMWORDS	BPF_RET
BPF_STMT	BPF_SUB	EXTRACT_LONG	INT	netlong	types	UCHAR	W32N_htonl
winsoc							

#### File2 Identifiers

0x0004	0x0005	__stdcall	_TEXT	_W32N_AD	_WAdapt	_WAdapt	_WAdapt
				A	er0	er1	er2

## Comparing Source Code Files

dwDataL en	DWOR D	dwType	ERR_IMPL IED	ERR_SUCC ESS	H_LOCA L	hAdapter	hClassNet
KEY_RE AD	LONG	pAdapterI nfo	PCHAR	PW_ADAP TER	QueryVal ue	TChar	VER_WI N32
W0Adapt er	W0Wind ow	W0Windo ws	W32N_Ada pt	W32N_NET	WINCAR DS	wsprintf	



SCORE 100

CodeSuite copyright 2003-2010 by Software Analysis and Forensic Engineering Corporation



**CodeSplit Basic Report**Version: 1.0.0 | Date: 09/06/18 | Time:  
09:40:34**SETTINGS**

Split files in folder	Z:\SAFE Corporation\test\CodeSuite-AC\files 1 <i>Including subdirectories</i>
File types	*.c;*.h
Programming language	C
Algorithms selected	<ul style="list-style-type: none"> <li>• Extract Statements</li> <li>• Extract Comments and Strings</li> <li>• Extract Identifiers</li> </ul>

**RESULTS**

Z:\SAFE Corporation\test\CodeSuite-AC\files 1\\_line\_test.c

Hits	Website
------	---------

0	NONE
---	------

Z:\SAFE Corporation\test\CodeSuite-AC\files 1\aaa\_case.c

Hits	Website
------	---------

1	X9 REGISTRY FOR CHECK IMAGE TESTS - x9.org
1	The Auxiliary On-Us Field: Why It Is Important To You THE ...
1	STRING: functional protein association networks

1	String Class (System)   Microsoft Docs
1	String (Java Platform SE 7 ) - Oracle Help Center
1	String (computer science) - Wikipedia
1	Private   Definition of Private by Merriam-Webster
1	Private   Define Private at Dictionary.com
1	Private Browsing - Use Firefox without saving history ...
1	Private (rank) - Wikipedia
1	Null   Definition of Null by Merriam-Webster
1	Null set - Wikipedia
1	null - JavaScript   MDN
1	Null - definition of null by The Free Dictionary
1	NULL - cppreference.com
1	Null (SQL) - Wikipedia


Z:\SAFE Corporation\test\CodeSuite-AC\files 1\aaa\_whitespace.c

Hits	Website
1	X9 REGISTRY FOR CHECK IMAGE TESTS - x9.org
1	The Auxiliary On-Us Field: Why It Is Important To You THE ...
1	STRING: functional protein association networks
1	String Class (System)   Microsoft Docs
1	String (Java Platform SE 7 ) - Oracle Help Center
1	String (computer science) - Wikipedia
1	Private   Definition of Private by Merriam-Webster
1	Private   Define Private at Dictionary.com
1	Private Browsing - Use Firefox without saving history ...

1	Private (rank) - Wikipedia
1	Null set - Wikipedia
1	NULL - cppreference.com
1	Doctors from Mangaluru's KMC identify rare bloodgroup 'P ...
1	c++ - Why do I say int *p = NULL in the declaration, but p ...
1	Axios - Official Site
1	Auxin - Wikipedia

Z:\SAFE Corporation\test\CodeSuite-AC\files 1\aaa\_with\_comments.c

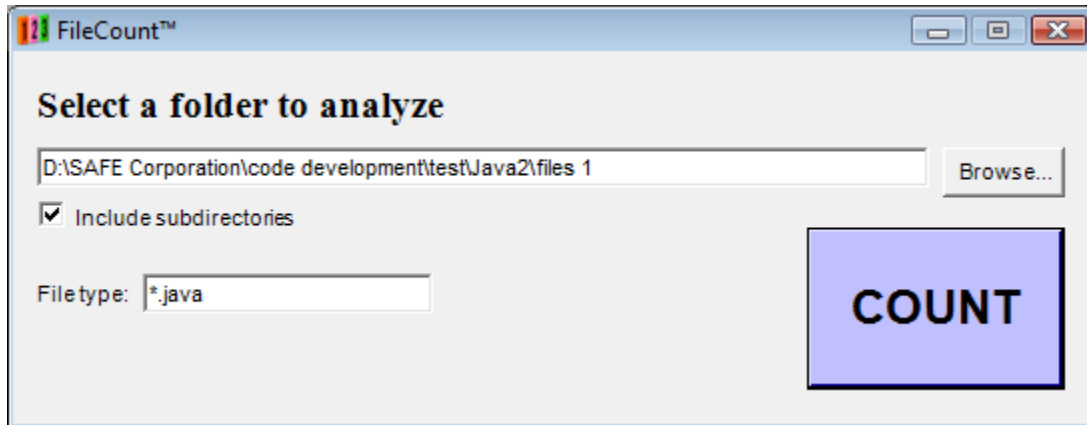
Hits	Website
1	YES Network - Official Site
1	YES
1	Yes - Official Site
1	Yes (band) - Wikipedia
1	X9 REGISTRY FOR CHECK IMAGE TESTS - x9.org
1	Tré Melvin - YouTube
1	The Auxiliary On-Us Field: Why It Is Important To You THE ...
1	STRING: functional protein association networks
1	String Class (System)   Microsoft Docs
1	String (Java Platform SE 7 ) - Oracle Help Center
1	String (computer science) - Wikipedia
1	Private   Definition of Private by Merriam-Webster
1	Private   Define Private at Dictionary.com
1	Private Browsing - Use Firefox without saving history ...
1	Private (rank) - Wikipedia

1	Null set - Wikipedia
<b>TOTALS</b>	
<b>Total number of bytes in files in folder 1 = 776</b>	
<b>Total run time = 1 Second</b>	
 TOP	
CodeSuite copyright 2003-2018 by Software Analysis and Forensic Engineering Corporation	

# FileCount

## Running FileCount

FileCount is a utility that counts the number of files, non-blank lines, and bytes in a large set of files in a directory tree.



### Step 1

Select the folder where the files are that need to be counted by clicking on the browse button or entering the path in the text field. Check the box to include all subdirectories.

### Step 2

Type in the file types. Separate different file types with a semicolon. Use the \* and ? wildcard characters if needed.

### Step 3

Press the count button. FileCount will then search the directory and all subdirectories, if specified, counting all of the files that meet the file type, and counting the total number of non-blank lines and bytes. When complete, a dialog box will appear with these counts.

# FileIdentify

## Running FileIdentify

FileIdentify allows a directory or directory tree and lists all of the file types found, based on the file name extensions. It also reports all known programming language files based on the file types. Below is a screen shot of the FileIdentify form and step-by-step instructions for running FileIdentify.



### Step 1

Select the folder where the files are located that you want to analyze. Check the box to include all subdirectories if you want to analyze files in the subfolders also.

### Step 2

Press the go button. You will be asked for the file name and location for a spreadsheet showing all file types and their associated programming languages, if known. FileIdentify will then search the directory and all subdirectories, if specified.

Below is an example of a spreadsheet created by FileIdentify.

	A	B	C
1	Analysis of Extensions		
2	Analysis date	12/16/2012	
3	Folder	Z:\SAFE Corporation\test\FileIdentify\top	
4	Include subfolders	Yes	
5			
6	Files with no extension	0	
7	Files with an empty extension	0	
8	Folder paths too long	0	
9	File paths too long	0	
10			
11	File types	Number of files	Language (if known)
12	.as	37	ActionScript

13	.c	81	C
14	.cdb	6	
15	.csf	1	
16	.flr	1	
17	.gif	47	
18	.htm	181	
19	.jpg	8	
20	.js	413	JavaScript
21	.mako	1	
22	.php	8	PHP
23	.png	49	
24	.swf	517	
25	.txt	66	

The top line shows that the spreadsheet was an analysis of file extensions created by FileIdentify. The second line shows the date that the analysis was run. The third shows the folder name. The fourth line indicates whether or not subfolders were included in the analysis.

Line 6 gives the number of files that had no extension while line 7 gives the number of files that had an empty extension, meaning the file name ended in a dot. Line 8 gives the number of folders that exceeded the maximum number of characters and could thus not be examined while line 9 gives the number of file paths, meaning the folder name plus the file name, that exceeded the maximum number of characters and could thus not be examined.

Lines 12 through 25 show the files types that were found, in column A, the number of files for each file type, in column B, and the programming language, if known, in column C.

# Languages

## Languages Supported

The following programming languages are currently supported:

ABAP	ASM-6502	ASM-65C02	ASM-65816	ASM-C55x
ASM-C67x	ASM-M68k	BASIC	C	C++
C#	COBOL	D	Delphi	DRI ASM
Flash ActionScript	Fortran	FoxPro	Go	Java
JavaScript	Kotlin	LISP	LotusScript	Lua
MASM	MATLAB	MPE/iX	Objective-C	OpenEdge
Pascal	Perl	PHP	PL/M	PL/SQL
PowerBuilder	PowerHouse	PowerShell	Progress	Prolog
Python	RealBasic	Ruby	Scala	SQL
Structured Text	Swift	TAL	TCL	TypeScript
Verilog	VHDL	Visual Basic		

Check the SAFE Corporation website for new language modules, available at no charge, as they become available. If the language you need is not available, contact SAFE Corporation about creating it for a nominal fee.



# Contacting SAFE Corporation

## Contacting SAFE Corporation



Software Analysis and Forensic Engineering Corporation

Web: [www.SAFE-corp.com](http://www.SAFE-corp.com)

Email: [Support@SAFE-corp.com](mailto:Support@SAFE-corp.com)

# Index

## **A**

ABAP 28  
ASM-6502 28  
ASM-65816 28  
ASM-65C02 28  
ASM-C55x 28  
ASM-C67x 28  
ASM-M68k 28  
Authorization Key 5

## **B**

BASIC 28

## **C**

C programming language 28  
C# 28  
C++ 28  
COBOL 28  
CodeMatch 1, 10, 12  
CodeMatch algorithms 10  
CodeMatch Algorithms 12  
CodeMatch Basic Report 14  
CodeMatch Detailed Report 17  
CodeSplit 1, 10  
CodeSplit Basic Report 21  
Comment/String Matching 12  
Compare Code 7  
Copyrights 3  
Correlation Score 13

## **D**

D programming language 28  
Delphi 28  
DRI ASM 28

## **F**

FileCount 2  
FileCount 7  
FileCount 25  
FileIdentify 2  
FileIdentify 7  
FileIdentify 26  
Flash ActionScript 28  
Fortran 28

FoxPro 28

## **G**

Go 28

## **H**

HTML reports 10

## **I**

Identifier Matching 13  
Instruction Sequence Matching 12

## **J**

Java 28  
JavaScript 28

## **K**

Kotlin 28

## **L**

Languages Enabled 6  
Languages Supported 28  
License Type 5  
    File-size based 6  
    Time based 6  
    Unlimited 6  
Licenses 5, 8  
    Allocated 6  
    Remaining 6

LISP 28  
LotusScript 28  
Lua 28

## **M**

MASM 28  
MATLAB 28  
MPE/iX 28

## **O**

Objective-C 28  
OpenEdge 28

## **P**

Pascal 28  
Patents 3  
Perl 28  
PHP 28  
PL/M 28  
PL/SQL 28

PowerBuilder 28

PowerHouse 28

PowerShell 28

Progress 28

Prolog 28

Python 28

## **R**

RealBasic 28

Ruby 28

## **S**

SAFE Corporation 3, 29

Scala 28

Software Analysis and Forensic  
Engineering 29

SourceDetective 2

SourceDetective 10

SQL 28

Statement Matching 12

Structured Text 28

Swift 28

System Requirements 4

## **T**

TAL 28

TCL 28

Templates 10

Toolbar 7

Trademarks 3

TypeScript 28

## **U**

URLs 10

## **V**

Verilog 28

VHDL 28

Visual Basic 28

## **W**

Whitespace 12

Windows 10 4

Windows 11 4

Windows 7 4

Windows 8 4

Windows Vista 4