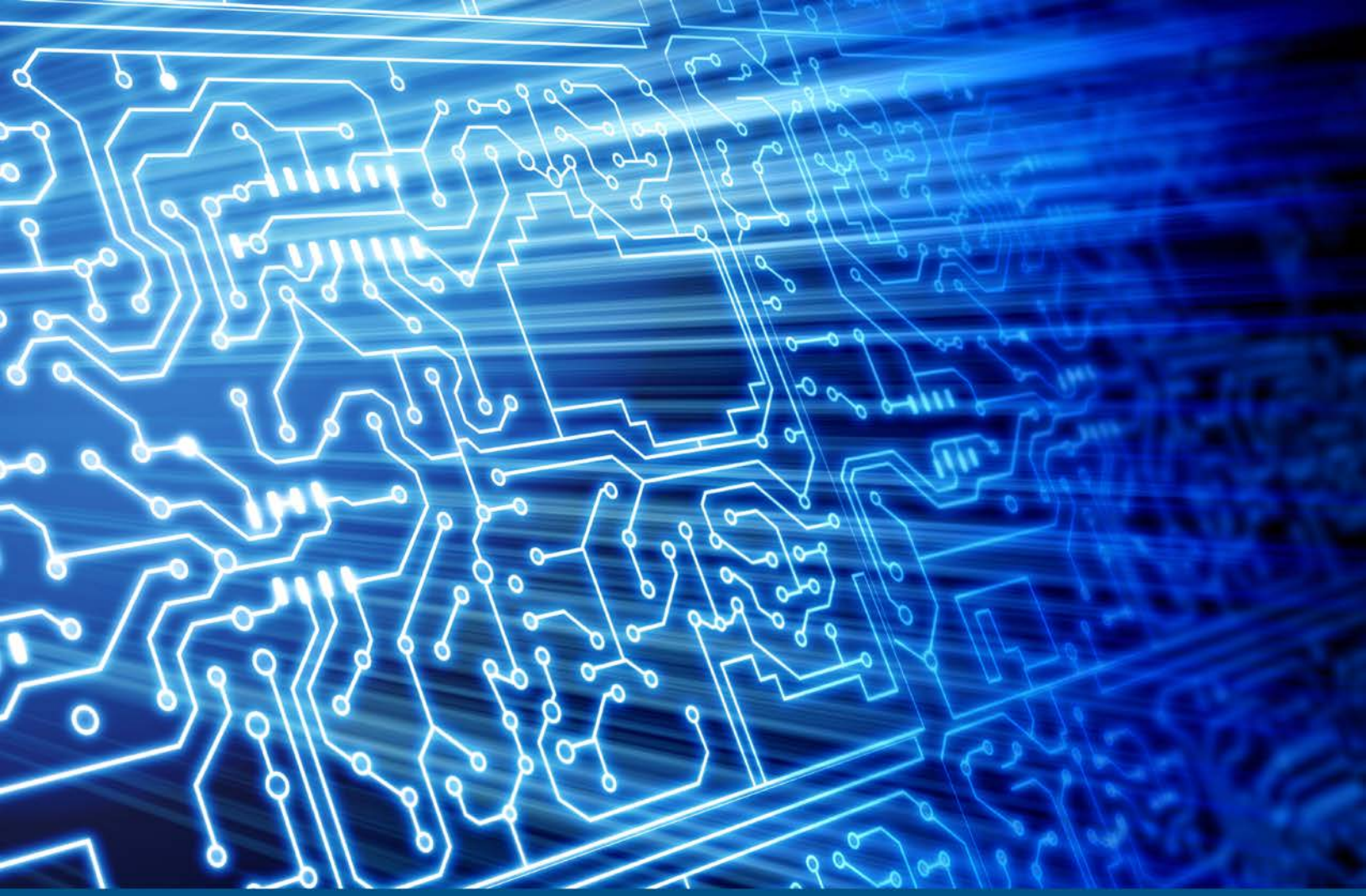

Autopsy Version 4.6.0

Test Results for String Search Tool

November 11, 2018

Homeland Security

Science and Technology

This report was prepared for the Department of Homeland Security Science and Technology Directorate Cyber Security Division by the Office of Law Enforcement Standards of the National Institute of Standards and Technology.

For additional information about the Cyber Security Division and ongoing projects, please visit the DHS website.

November 2018

Test Results for String Search Tool: Autopsy Version 4.6.0

Introduction

The Computer Forensics Tool Testing (CFTT) program is a joint project of the Department of Homeland Security (DHS), the National Institute of Justice (NIJ), and the National Institute of Standards and Technology (NIST) Special Programs Office and Information Technology Laboratory (ITL). CFTT is supported by other organizations, including the Federal Bureau of Investigation, the U.S. Department of Defense Cyber Crime Center, U.S. Internal Revenue Service Criminal Investigation Division Electronic Crimes Program, and the U.S. Department of Homeland Security's Bureau of Immigration and Customs Enforcement, U.S. Customs and Border Protection and U.S. Secret Service. The objective of the CFTT program is to provide measurable assurance to practitioners, researchers, and other applicable users that the tools used in computer forensics investigations provide accurate results. Accomplishing this requires the development of specifications and test methods for computer forensics tools and subsequent testing of specific tools against those specifications.

Test results provide the information necessary for developers to improve tools, users to make informed choices, and the legal community and others to understand the tools' capabilities. The CFTT approach to testing computer forensics tools is based on well-recognized methodologies for conformance and quality testing. The CFTT approach tests features that forensic labs are likely to use on a regular basis. Interested parties in the computer forensics community can review and comment on the specifications and test methods posted on the CFTT website.

This document reports the results from testing the string search function of Autopsy Version 4.6.0 using the CFTT Federated Testing Test Suite Version 4.0 (beta version, final to be released in 2018) using String Searching data set Version 1.1.

Federated Testing is an expansion of the CFTT program to provide forensic investigators and labs with test materials for tool testing and to support shared test reports. The goal of Federated Testing is to help forensic investigators to test the tools that they use in their labs and to enable sharing of tool test results. CFTT's Federated Testing Forensic Tool Testing Environment and included test suites can be downloaded by visiting CFTT website and selecting Federated Testing. The results can be optionally shared with CFTT, reviewed by CFTT staff, and then shared with the community.

Test results from this and other tools can be found on DHS's computer forensics web page.

Table of Contents

List of Tables

How to Read This Report

This report is organized into these sections:

1. Tested Tool Description. The tool name, version, and vendor information are listed.
2. Results Summary. This section identifies any significant anomalies observed in the test runs. This section provides a narrative of key findings identifying where the tool meets expectations and provides a summary of tool behaviors that did not meet expectations.
3. Test Environment & Selected Cases. Description of hardware and software used in tool testing and a list identifying the applicable test cases from the Federated Testing String Search Test Suite.
4. Test Result Details by Case. Automatically generated test results that identify anomalies.

Test Results for String Search Tool: Autopsy Version 4.6.0

1 Tested Tool Description

Tool Name: Autopsy
Tool Version: 4.6.0
Vendor: Open Source Tool from https://www.sleuthkit.org/

2 Results Summary

The test data set and test cases used to create this test report are limited to frequently encountered aspects of searching for text. Trying to cover every feature is not practical, but these test cases do cover a broad range of features. The features that are addressed in the full test data set (including features that Autopsy does not support) are listed below:

- File System: MS Windows (FAT, exFAT, NTFS) and UNIX-like (Ext4, OSXJ -- Mac OS Extended (Journaled), OSXC -- Mac OS Extended (Case-sensitive, Journaled) and APFS – Apple File System).
- String Location: Active File, Deleted (but recoverable) file, Unallocated Space, and Meta-Data.
- Search Method (aka search engine): Indexed, Live or Physical.
- String Encoding: ASCII, UTF-8, UTF-16BE and UTF-16LE with and without a **byte order mark**.
- Normalized Unicode: Match alternative forms of character representation, e.g., the substring “if” of the string “infinity” could be represented by a single ligature character or two separate characters, a letter with a diacritic mark could be represented by either one or two characters. A search for any one representation should match either representation.
- Language: In addition to English, strings representative of diacritical marks (German, French, Spanish), non-Latin characters (Russian), right-to-left presentation (Arabic), and Asian languages (Chinese, Japanese and Korean).
- Fragmented File: String that spans two disjoint file fragments.
- Logical Operations: Combine search results with logical operators **and**, **or** and **not**.
- Stemming: Match inflected forms derived from a word stem, e.g., a search for *run* should also match *runs*, *running* and *ran*.
- Embedded Formatting: String with embedded formatting. MS Word and HTML.

The following features are not supported by Autopsy:

- Live or physical search engines.
- Normalized Unicode string searching.
- Logical operations in searches.
- Stemming search.

- Apple File System is not supported, but it is treated as unallocated space.

The following results were observed (more details in Section: 4):

- With a few exceptions, all search targets located in active and deleted files were found. The exceptions where strings were missed were:
    - UTF-8 encoded strings, “flintlock” and “rifle,” contained in a MS Word .doc file were not found.
    - The phone number “(901)555-1111” was not found by the built-in phone number search. The string was found if searched for directly.
    - For Unicode strings with multiple representations, e.g., diacritic marks such as Spanish “tilde” or ligatures such as “if” the search is not normalized, i.e., only the representation that is an exact match to the search string is returned rather than all the equivalent strings.
    - The Apple File System was treated as unallocated space.
- Some UTF search strings located in unallocated space were missed.
    - UTF-16 strings encoded with a byte-order-mark for Chinese/Japanese Kanji, Korean, Japanese Katakana, French, and German were not reported, but Japanese Hiragana was reported.
    - Chinese/Japanese Kanji UTF-16 strings encoded with a byte-order-mark were not reported, but strings without a byte-order-mark were reported.
    - The Kanji for “□ □ ″ located in unallocated space was always missed, regardless of encoding.
    - UTF-16 strings for Russian and Arabic were reported regardless of encoding with a byte-order-mark.
- The **Global Keyword Search Setting** exhibited unexpected behavior for certain settings.
    - When both UTF-8 and UTF-16 were not selected. ASCII text in unallocated storage was missed. If either UTF-16 or UTF-8 was selected, the missing text was found.
    - When the **Han** language was selected. Some non-**Han** text (in various encodings including ASCII) in unallocated space was missed. Non-**Han** language selections had no observed impact on other languages.
- Korean text was not displayed correctly in the user interface but was correctly rendered in generated reports.

3 Test Environment & Selected Cases

This section describes test hardware, software, test data sets and test cases.

3.1 Test Hardware and Software

Autopsy 4.6.0 was installed on a Dell OptiPlex 7050 with 32GB installed RAM, running Microsoft Windows 10 Enterprise, Version 1607, OS Build 14393.2068.

Testing was performed using CFTT Federated Testing Test Suite Version 4.0 (beta version, final to be released in 2018).

3.2 Test Data Sets and Test Cases

This section discusses the test data sets and the test cases used in testing.

3.2.1 Test Data Sets

String search test data set package Version 1.1 was used. The package can be downloaded from either the CFTT web site (www.cftt.nist.gov then select String Searching) or the CFReDS web site (www.cfreds.nist.gov). The package includes two dd files with known content. One of the dd test images contains target strings within FAT, ExFAT and NTFS file systems (Windows), the other dd test image contains target strings from HFS+ journaled, case insensitive (OSXJ), HFS+ journaled, case sensitive (OSXC), ext4 file system and APFS (Apple file system) (UNIX-like).

In general, each target string is encoded in ASCII and located in both an active file and a recoverable deleted file in each partition of the test image. The Windows dd image also has a block of unallocated storage that contains the target strings without a file system. Some of the target strings are also encoded in Unicode UTF-8, UTF-16BE and UTF-16LE with a byte-order-mark. Test case FT-SS-07 is organized to test language and Unicode specific situations such as Unicode UTF-16 without a byte-order-mark, Unicode text with and without combining characters (diacritic marks), Unicode text with and without ligatures ("fi" as two characters and as one character) Test case FT-SS-09 is organized to test specific situations such as formatted strings, strings spanning file fragments, and strings located in inaccessible areas. Each instance of a target string also has a unique associated string ID located immediately after the target string. The string ID helps identify the specific string matched by the search tool.

3.2.2 Test Case Descriptions

The following table gives a brief description of available test cases in the data sets. Not all test cases are used for all data sets.

Table 1 Test Cases

Case	Case Description
FT-SS-01	Search ASCII
FT-SS-02	Search Ignore Case
FT-SS-03	Search for Words
FT-SS-04	Search Logical AND
FT-SS-05	Search Logical OR
FT-SS-06	Search Logical NOT

Case	Case Description
FT-SS-07-CJK-char	Search Unicode Chinese/Japanese ideograms (Asian)
FT-SS-07-CJK-hangul	Search Unicode CJK Korean Hangul (Asian)
FT-SS-07-CJK-kana	Search Unicode CJK Japanese phonetic Kana (Asian)
FT-SS-07-Cyrillic	Search Unicode Cyrillic (Russian)
FT-SS-07-Latin	Search Unicode Latin (French & German)
FT-SS-07-NoBOM	Search Unicode 16 without a byte-order-mark
FT-SS-07-Norm	Search Unicode 16 for normalized diacritic marks (NFC & NFD) and ligatures (NFKC & NFKD)
FT-SS-07-RTL	Search Unicode RTL (Arabic)
FT-SS-08-Email	Search Tool-defined Queries -- Email Address
FT-SS-08-Phone	Search Tool-defined Queries -- Telephone Number
FT-SS-08-SS	Search Tool-defined Queries -- Social Security
FT-SS-09-Doc	Search Formatted Document Text
FT-SS-09-Frag*	Search Fragmented File
FT-SS-09-Lost*	Search Inaccessible (lost) Areas
FT-SS-09-MFT*	Search File in NTFS MFT
FT-SS-09-Meta	Search file name substring in Meta-data
FT-SS-09-Stem	Search for matches to word stem
FT-SS-10-Hex	Search Hexadecimal Character Match
FT-SS-10-Regex	Search Pattern Character Match

Some test cases are for specific features, e.g., logical conditions (**and**, **or**, **not**), built in searches (email, telephone numbers), etc. Three test cases (marked with "*"), FT-SS-09-Frag, FT-SS-09-Lost & FT-SS-09-MFT are only applied to the Windows data set.

4 Test Result Details by Case (per Data Set)

A string search tool may implement more than one search algorithm (also known as a search engine) for searching text. The two most common search engines are *indexed search* and *live search*. An indexed search reads all the acquired data once before doing any searching and builds an index to all words found. Each query can be looked up quickly in the index. A Live search reads all the acquired data for each query.

This section presents test results by test image: windows file systems, or UNIX-like file systems. For each test image, there is a result table for each search engine tested. Each table shows results by test case of the number of expected search hits, the number of actual search hits and the number of strings missed (i.e., expected hits minus actual hits) for allocated files, deleted files and unallocated space.

The following search engines were tested: Indexed.

4.1 Results for Data Set: Windows

This section provides results for the Windows data set.

4.1.1 Results for Indexed Search of Windows Data Set

The table columns contain the following information:

- **Case** The test case identifier.
- **Expected String** The expected strings that should be reported by the search.
- **Active Files** A group of three columns (**Expected, Hits and Misses**) giving the number of hits and misses when searching for the expected string in an active file.
- **Deleted Files** A group of three columns (**Expected, Hits and Misses**) giving the number of hits and misses when searching for the expected string in a deleted file.
- **Unallocated Space** A group of three columns (**Expected, Hits and Misses**) giving the number of hits and misses when searching for the expected string in unallocated space.
- **Expected** The number of instances of the expected string found in the group (i.e., Active files, Deleted files or Unallocated space).
- **Hits** The number of times the expected string was found in the group.
- **Misses** The number of times the expected string was missed (not found) in the group.

Notes: If the row identifies a test case, then the results are a summary for all the strings that should be found.

In the Expected String column for test case FT-SS-09-DOC each string is labeled to indicate features of the expected string. The labels include the file type (.doc, .docx or .html), the encoding of the string in the .doc file and if the string has embedded formatting, labeled as *Formatted*, e.g., the string *crossbow* has the substring *cross* formatted as bold and underlined, i.e., **cross**bow.

Table 2 Test Results for Indexed Search of Windows Data Set

Case	Expected String	Active Files			Deleted Files			Unallocated Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-01		3	3	0	3	3	0	1	1	0
	DireWolf	3	3	0	3	3	0	1	1	0
FT-SS-02		15	15	0	15	15	0	5	5	0
	WOLF	3	3	0	3	3	0	1	1	0
	wolf	3	3	0	3	3	0	1	1	0
	Wolf	3	3	0	3	3	0	1	1	0
	DireWolf	3	3	0	3	3	0	1	1	0
	WereWolf	3	3	0	3	3	0	1	1	0
FT-SS-03		9	9	0	9	9	0	3	3	0
	WOLF	3	3	0	3	3	0	1	1	0
	wolf	3	3	0	3	3	0	1	1	0
	Wolf	3	3	0	3	3	0	1	1	0
FT-SS-07-CJK-char		18	18	0	18	18	0	6	2	4
	中国	9	9	0	9	9	0	3	1	2
	東京	9	9	0	9	9	0	3	1	2
FT-SS-07-CJK-hangul		9	9	0	9	9	0	3	1	2
	서울	9	9	0	9	9	0	3	1	2
FT-SS-07-CJK-kana		18	18	0	18	18	0	6	4	2
	スバル	9	9	0	9	9	0	3	1	2
	みつびし	9	9	0	9	9	0	3	3	0

Case	Expected String	Active Files			Deleted Files			Unallocated Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-07-Cyrillic		9	9	0	9	9	0	3	3	0
	Сибирь	9	9	0	9	9	0	3	3	0
FT-SS-07-Latin		18	18	0	18	18	0	6	2	4
	garçon	9	9	0	9	9	0	3	1	2
	Schönheit	9	9	0	9	9	0	3	1	2
FT-SS-07-NoBOM		39	39	0	39	39	0	13	13	0
	Россия	9	9	0	9	9	0	3	3	0
	فلافل	9	9	0	9	9	0	3	3	0
	中國	9	9	0	9	9	0	3	3	0
	QuarterHorse	12	12	0	12	12	0	4	4	0
FT-SS-07-Norm		75	75	0	75	75	0	25	9	16
	mañana (NFD)	9	9	0	9	9	0	3	0	3
	infinity (No Ligature)	12	12	0	12	12	0	4	4	0
	Mäuse (NFD)	9	9	0	9	9	0	3	0	3
	infinity (Ligature)	9	9	0	9	9	0	3	0	3
	Mäuse (NFC)	9	9	0	9	9	0	3	3	0
	libertà (NFC)	9	9	0	9	9	0	3	1	2
	libertà (NFD)	9	9	0	9	9	0	3	0	3
	mañana (NFC)	9	9	0	9	9	0	3	1	2
FT-SS-07-RTL		9	9	0	9	9	0	3	3	0
	الكسكس	9	9	0	9	9	0	3	3	0
FT-SS-08-Email		21	21	0	21	21	0	7	7	0
	iron.man@marvel.com	12	12	0	12	12	0	4	4	0

Case	Expected String	Active Files			Deleted Files			Unallocated Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
	potus@capitol.gov	3	3	0	3	3	0	1	1	0
	berlin@deutchland.net	3	3	0	3	3	0	1	1	0
	kgb@moscow.red.square.ru	3	3	0	3	3	0	1	1	0
FT-SS-08-Phone		21	18	3	21	18	3	7	6	1
	(901)555-1111	3	0	3	3	0	3	1	0	1
	301.555-9009	12	12	0	12	12	0	4	4	0
	800-555-1122	3	3	0	3	3	0	1	1	0
	202.555.3270	3	3	0	3	3	0	1	1	0
FT-SS-09-Doc		16	16	0	0	0	0	16	13	3
	longbow .html	2	2	0	0	0	0	2	2	0
	shotgun Formatted .doc UTF-16	2	2	0	0	0	0	2	2	0
	revolver .doc UTF-16	2	2	0	0	0	0	2	2	0
	peroxide .docx	2	2	0	0	0	0	2	1	1
	nitroglycerin Formatted .docx	2	2	0	0	0	0	2	1	1
	rifle .doc UTF-8	2	2	0	0	0	0	2	2	0
	crossbow Formatted .html	2	2	0	0	0	0	2	1	1
	flintlock Formatted .doc UTF-8	2	2	0	0	0	0	2	2	0

Case	Expected String	Active Files			Deleted Files			Unallocated Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-09-Frag		2	2	0	0	0	0	0	0	0
	Washington	1	1	0	0	0	0	0	0	0
	California	1	1	0	0	0	0	0	0	0
FT-SS-09-Lost		0	0	0	0	0	0	4	4	0
	SecretKey	0	0	0	0	0	0	2	2	0
	disconnected	0	0	0	0	0	0	2	2	0
FT-SS-09-MFT		4	4	0	4	4	0	0	0	0
	bear	4	4	0	4	4	0	0	0	0
FT-SS-09-Meta		6	6	0	6	6	0	2	2	0
	cañón	3	3	0	3	3	0	1	1	0
	thunderbird	3	3	0	3	3	0	1	1	0
FT-SS-10-Regex		6	6	0	6	6	0	2	2	0
	DireWolf	3	3	0	3	3	0	1	1	0
	WereWolf	3	3	0	3	3	0	1	1	0

4.1.2 Meta-Data results for Indexed Search of Windows Data Set

The following table presents search results for strings located in file system meta-data. The **Case** column identifies the test case, the **String** column identifies the search string, the **Partition** column identifies the partition (file system) where the string is located and the **Seen** column records if the search tool reported at least one instance of the string (yes or no) in meta-data.

Table 3 Meta-data Results for Indexed Search of Windows Data Set

Case	String	Partition	Seen
FT-SS-09-Meta			
	thunderbird	ntfs	Yes
	cañón	fat32	Yes
	cañón	exfat	Yes
	cañón	ntfs	Yes

4.1.3 Comments on Indexed Search of Windows Data Set

The following table presents any comments recorded during testing for a test case.

Table 4 Comments on Indexed Search of Windows Data Set

Case	Comments
FT-SS-01	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-02	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-03	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-07-CJK-char	Search target 中国 strings encoded as UTF-8 and located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-07-CJK-hangul	Search target strings encoded as UTF-8 and located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-07-CJK-kana	Search target string, スバル, encoded as UTF-8 and located in deleted files is reported twice, once from the deleted file and again as from unallocated storage. Search target string, みつびし, located in deleted files is reported twice, once from the deleted file and again as from unallocated storage.

Case	Comments
FT-SS-07-Cyrillic	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-07-Latin	Search target strings encoded as UTF-8 and located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-07-NoBOM	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-07-Norm	Tool did not normalize the search string. Search strings entered in NFC form found all targets in active files and deleted files and all targets located in unallocated space if encoded in UTF-8, but sometimes targets encoded in UTF-16 were missed. Search strings entered in NFD form found all targets in active files and deleted files and no targets located in unallocated space. Search results were the same if the combining characters (or ligature) are replaced with a regular expression of match any character.
FT-SS-07-RTL	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-08-Email	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-08-Phone	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.
FT-SS-09-MFT	All the strings are listed as being in the $MFT, except for string ID 7011. However, the string contains the fix-up-byte and is skipped in the "indexed text" window. The string ID for string 7010 contains a fix-up-byte and appears corrupted in the "indexed text" window.
FT-SS-09-Meta	Hits on the string "thunderbird" are reported in the files $MFT and $LogFile.
FT-SS-10-Regex	Search target strings located in deleted files are reported twice, once from the deleted file and again as from unallocated storage.

4.2 Results for Data Set: UNIX

This section provides results for the UNIX data set.

4.2.1 Results for Indexed Search of UNIX Data Set

The table columns contain the following information:

- **Case** The test case identifier.

- **Expected String** The strings that should be reported by the search.
- **Active Files** A group of three columns (**Expected, Hits and Misses**) giving the number of hits and misses when searching for the expected string in an active file.
- **Deleted Files** A group of three columns (**Expected, Hits and Misses**) giving the number of hits and misses when searching for the expected string in a deleted file.
- **Unallocated Space** A group of three columns (**Expected, Hits and Misses**) giving the number of hits and misses when searching for the expected string in unallocated space.
- **Expected** The number of instances of the expected string found in the group (i.e., Active files, Deleted files or Unallocated space).
- **Hits** The number of times the expected string was found in the group.
- **Misses** The number of times the expected string was missed (not found) in the group.

Notes: If the row identifies a test case, then the results are a summary for all the strings that should be found.

In the Expected String column for test case FT-SS-09-DOC each string is labeled to indicate features of the expected string. The labels include the file type (.doc, .docx or .html), the encoding of the string in the .doc file and if the string has embedded formatting, labeled as *Formatted*, e.g., the string *crossbow* has the substring *cross* formatted as bold and underlined, i.e., **cross**bow.

Table 5 Test Results for Indexed Search of UNIX Data Set

Case	Expected String	Active Files			Deleted Files		
		Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-01		4	4	0	4	4	0
	DireWolf	4	4	0	4	4	0
FT-SS-02		20	20	0	20	20	0
	WOLF	4	4	0	4	4	0
	wolf	4	4	0	4	4	0
	Wolf	4	4	0	4	4	0
	DireWolf	4	4	0	4	4	0
	WereWolf	4	4	0	4	4	0
FT-SS-03		12	12	0	12	12	0
	WOLF	4	4	0	4	4	0
	wolf	4	4	0	4	4	0
	Wolf	4	4	0	4	4	0
FT-SS-07-CJK-char		24	20	4	24	4	20
	中国	12	10	2	12	4	8

Case	Expected String	Active Files			Deleted Files		
		Expected	Hits	Misses	Expected	Hits	Misses
	東京	12	10	2	12	0	12
FT-SS-07-CJK-hangul		12	10	2	12	4	8
	서울	12	10	2	12	4	8
FT-SS-07-CJK-kana		24	22	2	24	16	8
	スバル	12	10	2	12	4	8
	みつびし	12	12	0	12	12	0
FT-SS-07-Cyrillic		12	12	0	12	12	0
	Сибирь	12	12	0	12	12	0
FT-SS-07-Latin		24	24	0	24	8	16
	garçon	12	12	0	12	4	8
	Schönheit	12	12	0	12	4	8
FT-SS-07-NoBOM		52	52	0	52	52	0
	Россия	12	12	0	12	12	0
	فلافل	12	12	0	12	12	0
	中國	12	12	0	12	12	0
	QuarterHorse	16	16	0	16	16	0
FT-SS-07-Norm		100	84	16	100	36	64
	mañana (NFD)	12	9	3	12	0	12
	infinity (No Ligature)	16	16	0	16	16	0
	Mäuse (NFD)	12	9	3	12	0	12
	infinity (Ligature)	12	9	3	12	0	12
	Mäuse (NFC)	12	10	2	12	4	8
	libertà (NFC)	12	12	0	12	12	0
	libertà (NFD)	12	9	3	12	0	12
	mañana (NFC)	12	10	2	12	4	8
FT-SS-07-RTL		12	12	0	12	12	0
	الكسكس	12	12	0	12	12	0
FT-SS-08-Email		28	28	0	28	28	0
	iron.man@marvel.com	16	16	0	16	16	0
	potus@capitol.gov	4	4	0	4	4	0

Case	Expected String	Active Files			Deleted Files		
		Expected	Hits	Misses	Expected	Hits	Misses
	berlin@deutchland.net	4	4	0	4	4	0
	kgb@moscow.red.square.ru	4	4	0	4	4	0
FT-SS-08-Phone		28	24	4	28	24	4
	(901)555-1111	4	0	4	4	0	4
	301.555-9009	16	16	0	16	16	0
	800-555-1122	4	4	0	4	4	0
	202.555.3270	4	4	0	4	4	0
FT-SS-09-Doc		16	16	0	0	0	0
	longbow .html	2	2	0	0	0	0
	shotgun Formatted .doc UTF-16	2	2	0	0	0	0
	revolver .doc UTF-16	2	2	0	0	0	0
	peroxide .docx	2	2	0	0	0	0
	nitroglycerin Formatted .docx	2	2	0	0	0	0
	rifle .doc UTF-8	2	2	0	0	0	0
	crossbow Formatted .html	2	2	0	0	0	0
	flintlock Formatted .doc UTF-8	2	2	0	0	0	0
FT-SS-09-Meta		8	8	0	8	8	0
	cañón	4	4	0	4	4	0
	thunderbird	4	4	0	4	4	0
FT-SS-10-Regex		8	8	0	8	8	0
	DireWolf	4	4	0	4	4	0
	WereWolf	4	4	0	4	4	0

4.2.2 Meta-Data results for Indexed Search of UNIX Data Set

The following table presents search results for strings located in file system meta-data. The **Case** column identifies the test case, the **String** column identifies the search string, the **Partition** column identifies the partition (file system) where the string is located and the **Seen** column records if the search tool reported at least one instance of the string (yes or no) in meta-data.

Table 6 Meta-data Results for Indexed Search of UNIX Data Set

Case	String	Partition	Seen
FT-SS-09-Meta			
	thunderbird	osxj	Yes
	thunderbird	osxc	Yes
	thunderbird	apfs	Yes
	thunderbird	ext4	Yes
	cañón	ext4	Yes

4.2.3 Comments on Indexed Search of UNIX Data Set

The following table presents any comments recorded during testing for a test case.

Table 7 Comments on Indexed Search of UNIX Data Set

Case	Comments
FT-SS-07-CJK-char	No UTF-16 search target string 中国 found in unallocated space or APFS. No search target string 東京 found in unallocated space or APFS.
FT-SS-07-CJK-hangul	No search target strings encoded UTF-16 in unallocated space or APFS were reported.
FT-SS-07-CJK-kana	No search target strings of Katakana (スバル), encoded UTF-16 in unallocated space or APFS were reported.
FT-SS-07-Latin	No search target strings encoded UTF-16 in unallocated space or APFS were reported.
FT-SS-07-Norm	Tool did not normalize the search string. Search strings entered in NFC form found all targets in active files and deleted files and all targets located in unallocated space if encoded in UTF-8, but sometimes targets encoded in UTF-16 were missed. Search strings entered in NFD form found all targets in active files, no targets located in unallocated space were found.

Case	Comments
	Search results were the same if the combining characters (or ligature) are replaced with a regular expression of mathc any character.
FT-SS-09-Meta	For the string "thunderbird" hits were reported in "0", ".journal", and "$catalog" files in the osxj and osxc (HFS+) file systems.