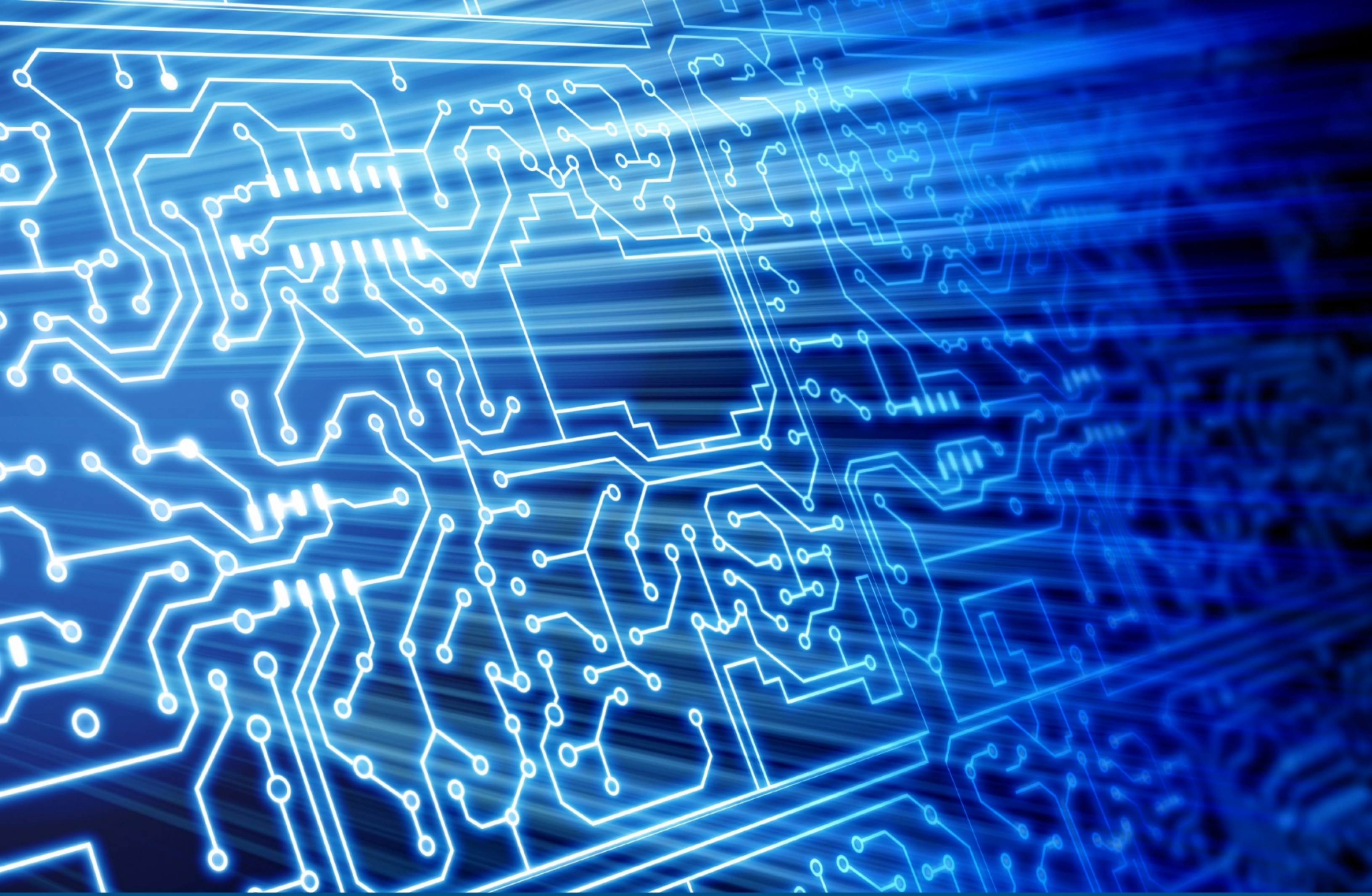


Test Results for String Search Tool:
EnCase Version 8.09.00.192

June 2020

Homeland Security

Science and Technology

This report was prepared for the Department of Homeland Security (DHS) Science and Technology Directorate (S&T) by the Office of Law Enforcement Standards of the National Institute of Standards and Technology.

For additional information about ongoing DHS S&T cybersecurity projects, please visit https://www.dhs.gov/science-and-technology/cybersecurity

June 2020

Test Results for String Search Tool: EnCase Version 8.09.00.192

Introduction

The Computer Forensics Tool Testing (CFTT) program is a joint project of the Department of Homeland Security, Science and Technology Directorate (DHS S&T), 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 Web site (https://www.cftt.nist.gov).

This document reports the results from testing the string search function of EnCase Version 8.09.00.192 using the CFTT Federated Testing Test Suite Version 4.0 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 https://www.cftt.nist.gov 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, https://www.dhs.gov/science-and-technology/nist-cftt-reports.

Table of Contents

How to Read This Report

This report is organized into the following 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 any ways the tool did not meet expectations. The section also provides any observations of interest about the tool or about testing the tool, including any observed limitations.
3. **Test Environment & Selected Test Cases:** Description of hardware, software and support environment (e.g., version of Federated Testing used, device firmware version, etc.) used in tool testing and a list identifying the applicable test cases selected from the Federated Testing String Search Test Suite.
4. **Test Result Details by Case:** Automatically generated test results that identify anomalies.

1 Tested Tool Description

Tool Name: EnCase
Tool Version: 8.09.00.192
Vendor:

OpenText
275 Frank Tompa Drive
Waterloo, ON
N2L 0A1
Canada

Phone: 519-888-7111
Fax: 519-888-0677

This test report was generated using CFTT's Federated Testing Forensic Tool Testing Environment, see Federated Testing Home Page.

2 Results Summary

This section provides an overview of string search testing and a list of observations from testing the tool under test.

2.1 Testing Overview

The test data sets 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 EnCase 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 or Live.

- 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 “fi” 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 that are representative of diacritical marks (German, French, Spanish), non-Latin characters (Russian), right-to-left presentation (Arabic), and Asian languages (Chinese, Japanese and Korean) are search targets.
- 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.

Two search engines were tested: Live Search and Indexed Search.

The following features are not supported by EnCase Version 8.09.00.192:

2.1.1 Features not supported by Live Search

- Logical combinations (**and, or** and **not**); test cases 04, 05 & 06.
- Normalized Unicode string searching. Each Unicode Normal Form must be searched for explicitly; test case 07-Norm.
- Built-in searches for phone numbers, email addresses and social security numbers; test cases 08-phone, 08-Email and 08-SS (of course, you can use your own regular expression to search for these strings).
- Stemming search is not supported; test case 09-Stem.

2.1.2 Features not supported by Indexed Search

- *Substring* and *case sensitive* searching.
- Normalized Unicode string searching. Each Unicode Normal Form must be searched for explicitly; test case 07-Norm.
- Built-in searches for phone numbers, email addresses and social security numbers; test cases 08-phone, 08-Email and 08-SS. (Although the user guide states that built-in searching is supported, when we could not get the expected hits for these searches, the vendor responded to our query: “Starting with EnCase v8.06 the [built-in search feature] was removed.”)
- Stemming search is not supported; test case 09-Stem.

2.2 Test Observations

We have the following observations*:*

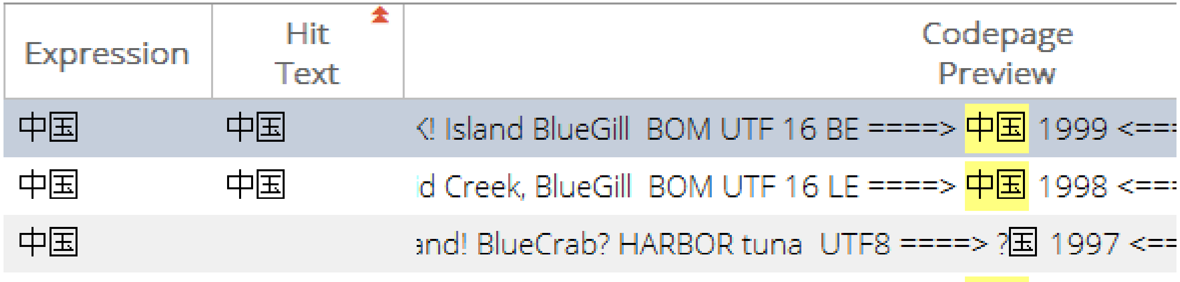
2.2.1 Live Search Observations

- Missed strings in Microsoft Office DOCX file located in unallocated space.
- Missed string from HTML file with embedded HTML formatting tags in the target string.

2.2.2 Indexed Search Observations

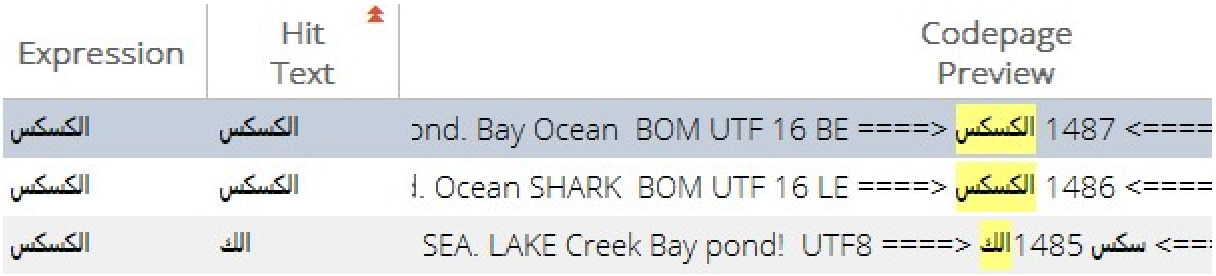
- Missed UTF-8 strings for non-English text.
- Missed UTF-16-BE strings for non-English text in unallocated space.
- Missed UTF strings for non-English text in files without a *byte-order-mark*.
- Returned false positives for multi-character Asian language search strings by only matching the last character in the string. For example, a search for *China,* represented as 中国, also returned hits for *America*, 美国.
- Double hits reported for ambiguous UNICODE UTF-16 strings in these languages: German, Spanish, French, English and Italian. Result is likely to generalize to any Latin-based text, e.g., Swedish, Dutch, Vietnamese, etc. However, if diacritical marks (or other multibyte code points) are included then only a single hit is reported.
- Some text rendering problems were also observed:

Rendering problem for Asian text: UTF-8 text is not displayed in the "Hit Text" column of reported results.



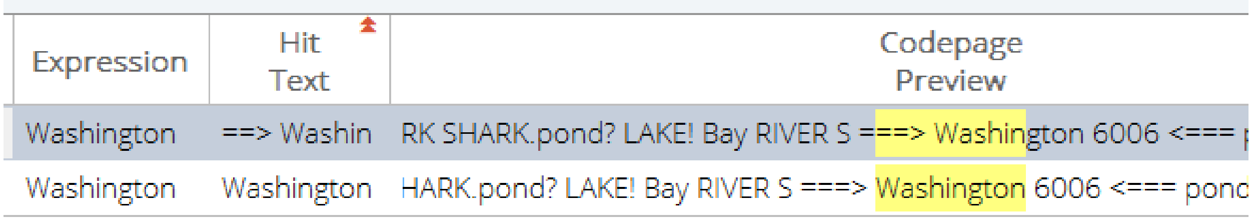
Expression	Hit Text	Codepage Preview
中国	中国	<! Island BlueGill BOM UTF 16 BE ====> 中国 1999 <==
中国	中国	d Creek, BlueGill BOM UTF 16 LE ====> 中国 1998 <==
中国		and! BlueCrab? HARBOR tuna UTF8 ====> ?国 1997 <==

Rendering problem for Arabic text, text is presented out of order for UTF-8. see below:

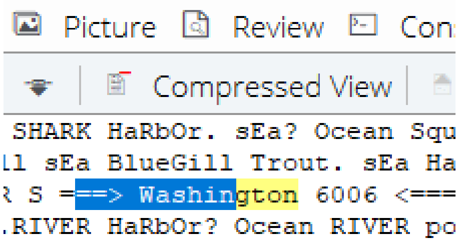


Expression	Hit Text	Codepage Preview
الكسكس	الكسكس	ɔnd. Bay Ocean BOM UTF 16 BE ====> الكسكس 1487 <====
الكسكس	الكسكس	ᛧ. Ocean SHARK BOM UTF 16 LE ====> الكسكس 1486 <====
الكسكس	الك	SEA. LAKE Creek Bay pond! UTF8 ====> الك1485 سكس <==

The highlighting of this search hit that spans a file fragment is unclear.

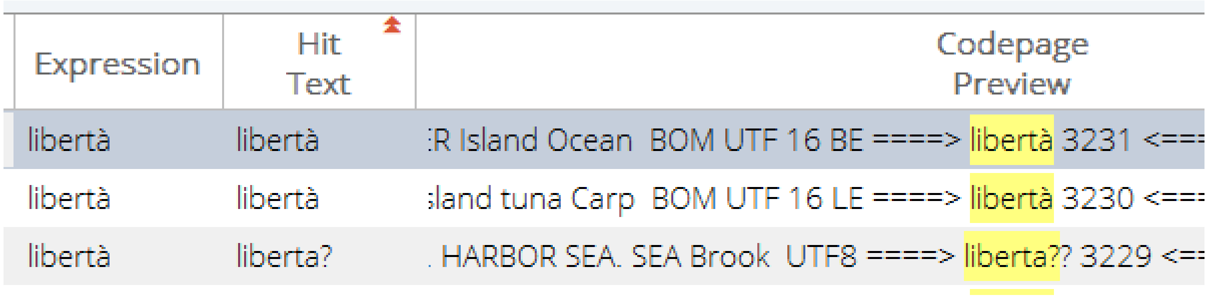


Expression	Hit Text	Codepage Preview
Washington	==> Washin	RK SHARK.pond? LAKE! Bay RIVER S ===> Washington 6006 <=== ɼ
Washington	Washington	HARK.pond? LAKE! Bay RIVER S ===> Washington 6006 <=== ponc

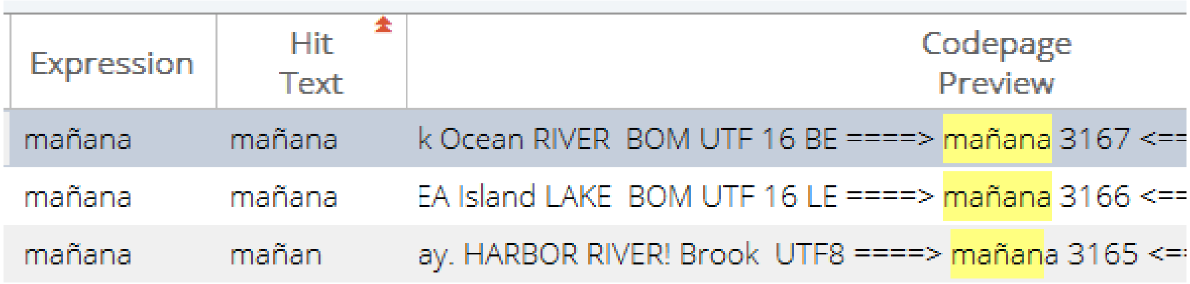


```
 SHARK HaRbOr. sEa? Ocean Squ
ll sEa BlueGill Trout. sEa Ha
R S ===> Washington 6006 <===
.RIVER HaRbOr? Ocean RIVER po
```

The search hit highlighting seems off, string length seems to be off by one for UTF-8 in Spanish and Italian with a diacritic mark:

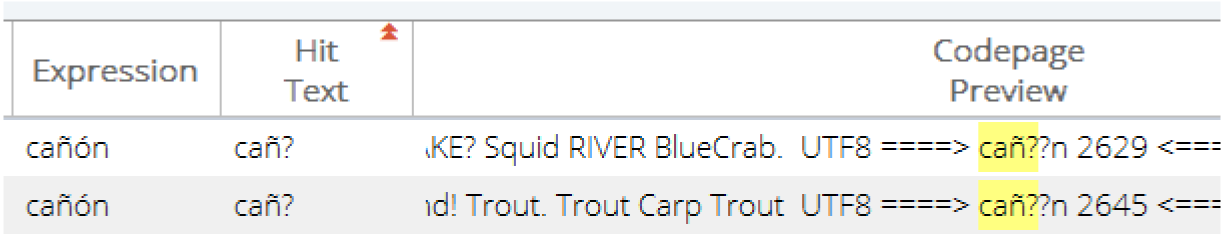


Expression	Hit Text	Codepage Preview
libertà	libertà	꞉R Island Ocean BOM UTF 16 BE ====> libertà 3231 <===
libertà	libertà	꞉land tuna Carp BOM UTF 16 LE ====> libertà 3230 <===
libertà	liberta?	. HARBOR SEA. SEA Brook UTF8 ====> liberta?? 3229 <==



Expression	Hit Text	Codepage Preview
mañana	mañana	k Ocean RIVER BOM UTF 16 BE ====> mañana 3167 <==
mañana	mañana	EA Island LAKE BOM UTF 16 LE ====> mañana 3166 <==
mañana	mañan	ay. HARBOR RIVER! Brook UTF8 ====> mañana 3165 <=

Note that the above items have a single diacritic and are off by one, but the string below has two diacritics and is off by two:



Expression	Hit Text	Codepage Preview
cañón	cañ?	KE? Squid RIVER BlueCrab. UTF8 ====> cañ??n 2629 <===
cañón	cañ?	id! Trout. Trout Carp Trout UTF8 ====> cañ??n 2645 <===

3 Test Environment & Selected Test Cases

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

3.1 Test Hardware and Software

The tool under test (EnCase Version 8.09.000.192) 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 4.1.

3.2 Test Data Sets and Test Cases

This section describes the test data sets and test cases that were used.

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.

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
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	Normalized Search of Unicode text with 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 Master File Table (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 and Live.

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 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: The first row of results for a test case is a summary for all the strings that should be found for that case.

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) and the encoding of the string (if a .doc file). If the string has embedded formatting it is labeled as *Formatted*, e.g., the string *crossbow* has the substring *cross* formatted as bold and underlined, i.e., **cross**bow.

Results for Indexed Search of Windows Data Set

Case	Expected String	Active Files			Deleted Files			Unalloc 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	9	6	15	9	6	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	0	3	3	0	3	1	1	0
	WereWolf	3	0	3	3	0	3	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-04		3	3	0	3	3	0	0	0	0
	panda and fox	3	3	0	3	3	0	0	0	0
FT-SS-05		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
FT-SS-06		12	12	0	12	12	0	0	0	0
	fox and not tiger	12	12	0	12	12	0	0	0	0
FT-SS-07-CJK-char		18	12	6	18	12	6	6	2	4
	中国	9	6	3	9	6	3	3	1	2
	東京	9	6	3	9	6	3	3	1	2

Results for Indexed Search of Windows Data Set										
Case	Expected String	Active Files			Deleted Files			Unalloc Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-07-CJK-hangul		9	6	3	9	6	3	3	1	2
	서울	9	6	3	9	6	3	3	1	2
FT-SS-07-CJK-kana		18	12	6	18	12	6	6	2	4
	スバル	9	6	3	9	6	3	3	1	2
	みつびし	9	6	3	9	6	3	3	1	2
FT-SS-07-Cyrillic		9	6	3	9	6	3	3	1	2
	Сибирь	9	6	3	9	6	3	3	1	2
FT-SS-07-Latin		18	12	6	18	12	6	6	6	0
	garçon	9	6	3	9	6	3	3	3	0
	Schönheit	9	6	3	9	6	3	3	3	0
FT-SS-07-NoBOM		39	12	27	39	12	27	13	2	11
	Россия	9	0	9	9	0	9	3	0	3
	لف الف	9	0	9	9	0	9	3	0	3
	中國	9	0	9	9	0	9	3	0	3
	QuarterHorse	12	12	0	12	12	0	4	2	2
FT-SS-07-Norm		75	54	21	75	54	21	25	8	17
	mañana (NFD)	9	6	3	9	6	3	3	1	2
	infinity (No Ligature)	12	12	0	12	12	0	4	1	3
	Mäuse (NFD)	9	6	3	9	6	3	3	1	2
	infinity (Ligature)	9	6	3	9	6	3	3	1	2
	Mäuse (NFC)	9	6	3	9	6	3	3	1	2
	libertà (NFC)	9	6	3	9	6	3	3	1	2
	libertà (NFD)	9	6	3	9	6	3	3	1	2
	mañana (NFC)	9	6	3	9	6	3	3	1	2

Results for Indexed Search of Windows Data Set										
Case	Expected String	Active Files			Deleted Files			Unalloc Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-07-RTL		9	6	3	9	6	3	3	1	2
	سكسكل ا	9	6	3	9	6	3	3	1	2
FT-SS-09-Doc		16	16	0	0	0	0	16	16	0
	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	2	0
	nitroglycerin Formatted .docx	2	2	0	0	0	0	2	2	0
	rifle .doc UTF-8	2	2	0	0	0	0	2	2	0
	crossbow Formatted .html	2	2	0	0	0	0	2	2	0
	flintlock Formatted .doc UTF-8	2	2	0	0	0	0	2	2	0
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	3	1
	SecretKey	0	0	0	0	0	0	2	2	0
	disconnected	0	0	0	0	0	0	2	1	1
FT-SS-09-MFT		4	4	0	4	4	0	0	0	0
	Bear	4	4	0	4	4	0	0	0	0

Results for Indexed Search of Windows Data Set										
Case	Expected String	Active Files			Deleted Files			Unalloc Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-09-Meta		6	6	0	6	6	0	2	1	1
	cañón	3	3	0	3	3	0	1	0	1
	thunderbird	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.

Meta-Data Results for Indexed Search of Windows Data Set			
Case	String	Partition	Seen
FT-SS-09-Meta			
	thunderbird	ntfs	Yes
	cañón	fat32	No
	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.

Case	Comments on Indexed Search of Windows Data Set
FT-SS-04	Also reported a hit from unallocated space
FT-SS-07-CJK-char	In addition to reporting hits for 中 国 (China), hits were also reported for America (another two character Chinese word that matches the second character (国) of the search string. The same issue occurs with Tokyo (東京).

4.1.4 Results for Live Search of Windows 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: The first row of results for a test case is a summary for all the strings that should be found for that case.

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) and the encoding of the string (if a .doc file). If the string has embedded formatting it is labeled as *Formatted*, e.g., the string *crossbow* has the substring *cross* formatted as bold and underlined, i.e., **cross**bow.

Results for Live Search of Windows Data Set										
Case	**Expected String**	**Active Files**			**Deleted Files**			**Unalloc 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

Results for Live Search of Windows Data Set										
Case	Expected String	Active Files			Deleted Files			Unalloc Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-07-CJK-char		18	18	0	18	18	0	6	6	0
	中国	9	9	0	9	9	0	3	3	0
	東京	9	9	0	9	9	0	3	3	0
FT-SS-07-CJK-hangul		9	9	0	9	9	0	3	3	0
	서울	9	9	0	9	9	0	3	3	0
FT-SS-07-CJK-kana		18	18	0	18	18	0	6	6	0
	スバル	9	9	0	9	9	0	3	3	0
	みつびし	9	9	0	9	9	0	3	3	0
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	6	0
	garçon	9	9	0	9	9	0	3	3	0
	Schönheit	9	9	0	9	9	0	3	3	0
FT-SS-07-NoBOM		39	39	0	39	39	0	13	13	0
	Россия	9	9	0	9	9	0	3	3	0
	لف الف	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	25	0
	mañana (NFD)	9	9	0	9	9	0	3	3	0
	infinity (No Ligature)	12	12	0	12	12	0	4	4	0
	Mäuse (NFD)	9	9	0	9	9	0	3	3	0
	infinity (Ligature)	9	9	0	9	9	0	3	3	0

Results for Live Search of Windows Data Set										
Case	Expected String	Active Files			Deleted Files			Unalloc Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
	Mäuse (NFC)	9	9	0	9	9	0	3	3	0
	libertà (NFC)	9	9	0	9	9	0	3	3	0
	libertà (NFD)	9	9	0	9	9	0	3	3	0
	mañana (NFC)	9	9	0	9	9	0	3	3	0
FT-SS-07-RTL		9	9	0	9	9	0	3	3	0
	سكسكل ا	9	9	0	9	9	0	3	3	0
FT-SS-09-Doc		16	15	1	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	1	1	0	0	0	2	1	1
	flintlock Formatted .doc UTF-8	2	2	0	0	0	0	2	2	0
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

Results for Live Search of Windows Data Set										
Case	Expected String	Active Files			Deleted Files			Unalloc Space		
		Expected	Hits	Misses	Expected	Hits	Misses	Expected	Hits	Misses
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-Hex		3	3	0	3	3	0	1	1	0
	Panda	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

4.1.5 Meta-Data results for Live 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.

Meta-Data Results for Live Search of Windows Data Set			
Case	String	Partition	Seen
FT-SS-09-Meta			
	thunderbird	ntfs	Yes
	cañón	fat32	No
	cañón	exfat	Yes
	cañón	ntfs	Yes

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: The first row of results for a test case is a summary for all the strings that should be found for that case.

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) and the encoding of the string (if a .doc file). If the string has embedded formatting it is labeled as *Formatted*, e.g., the string *crossbow* has the substring *cross* formatted as bold and underlined, i.e., **cross**bow.

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	14	6	20	5	15
	WOLF	4	4	0	4	1	3
	wolf	4	4	0	4	1	3
	Wolf	4	4	0	4	1	3
	DireWolf	4	1	3	4	1	3
	WereWolf	4	1	3	4	1	3
FT-SS-03		12	12	0	12	3	9
	WOLF	4	4	0	4	1	3
	Wolf	4	4	0	4	1	3
	Wolf	4	4	0	4	1	3

Results for Indexed Search of UNIX Data Set							
Case	Expected String	Active Files			Deleted Files		
		Expected	Hits	Misses	Expected	Hits	Misses
FT-SS-04		4	4	0	4	1	3
	panda and fox	4	4	0	4	1	3
FT-SS-05		8	8	0	8	2	6
	DireWolf	4	4	0	4	1	3
	WereWolf	4	4	0	4	1	3
FT-SS-06		16	16	0	16	4	12
	fox and not tiger	16	16	0	16	4	12
FT-SS-07-CJK-char		24	16	8	24	8	16
	中国	12	8	4	12	4	8
	東京	12	8	4	12	4	8
FT-SS-07-CJK-hangul		12	8	4	12	4	8
	서울	12	8	4	12	4	8
FT-SS-07-CJK-kana		24	16	8	24	8	16
	スバル	12	8	4	12	4	8
	みつびし	12	8	4	12	4	8
FT-SS-07-Cyrillic		12	8	4	12	4	8
	Сибирь	12	8	4	12	4	8
FT-SS-07-Latin		24	16	8	24	16	8
	garçon	12	8	4	12	8	4
	Schönheit	12	8	4	12	8	4
FT-SS-07-NoBOM		52	0	52	52	0	52
	Россия	12	0	12	12	0	12
	لف الف	12	0	12	12	0	12
	中國	12	0	12	12	0	12
	QuarterHorse	16	0	16	16	0	16
FT-SS-07-Norm		100	72	28	100	56	44
	mañana (NFD)	12	8	4	12	4	8
	infinity (No Ligature)	16	16	0	16	16	0
	Mäuse (NFD)	12	8	4	12	4	8
	infinity (Ligature)	12	8	4	12	4	8

Results for Indexed Search of UNIX Data Set							
Case	Expected String	Active Files			Deleted Files		
		Expected	Hits	Misses	Expected	Hits	Misses
	Mäuse (NFC)	12	8	4	12	8	4
	libertà (NFC)	12	8	4	12	8	4
	libertà (NFD)	12	8	4	12	4	8
	mañana (NFC)	12	8	4	12	8	4
FT-SS-07-RTL		12	8	4	12	4	8
	سكسكل ا	12	8	4	12	4	8
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	4	4	8	4	4
	cañón	4	0	4	4	0	4
	thunderbird	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.

Meta-Data Results for Indexed Search of UNIX Data Set			
Case	**String**	**Partition**	**Seen**
FT-SS-07-CJK-char			
	中国	osxj	No
	中国	osxc	No
	中国	apfs	No
	東京	osxj	No
	東京	osxc	No
	東京	apfs	No
FT-SS-07-Cyrillic			
	Сибирь	osxj	No
	Сибирь	osxc	No
	Сибирь	apfs	No
FT-SS-07-NoBOM			
	لف الف	osxj	No
	لف الف	osxc	No
	لف الف	apfs	No
	Россия	osxj	No
	Россия	osxc	No
	Россия	apfs	No
	中國	osxj	No
	中國	osxc	No
	中國	apfs	No
FT-SS-07-RTL			
	سكسكل ا	osxj	No
	سكسكل ا	osxc	No
	سكسكل ا	apfs	No
FT-SS-09-Meta			
	thunderbird	osxj	No

Meta-Data Results for Indexed Search of UNIX Data Set			
Case	**String**	**Partition**	**Seen**
	thunderbird	osxc	No
	thunderbird	apfs	No
	thunderbird	ext4	No
	cañón	ext4	No

4.2.3 Results for Live 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: The first row of results for a test case is a summary for all the strings that should be found for that case.

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) and the encoding of the string (if a .doc file). If the string has embedded formatting it is labeled as *Formatted*, e.g., the string *crossbow* has the substring *cross* formatted as bold and underlined, i.e., **<u>cross</u>**bow.

Results for Live 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

Results for Live Search of UNIX Data Set							
Case	Expected String	Active Files			Deleted Files		
		Expected	Hits	Misses	Expected	Hits	Misses
	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	24	0	24	24	0
	中国	12	12	0	12	12	0
	東京	12	12	0	12	12	0
FT-SS-07-CJK-hangul		12	12	0	12	12	0
	서울	12	12	0	12	12	0
FT-SS-07-CJK-kana		24	24	0	24	24	0
	スバル	12	12	0	12	12	0
	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	24	0
	garçon	12	12	0	12	12	0
	Schönheit	12	12	0	12	12	0
FT-SS-07-NoBOM		52	52	0	52	52	0
	Россия	12	12	0	12	12	0
	لف الف	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	100	0	100	100	0
	mañana (NFD)	12	12	0	12	12	0
	infinity (No Ligature)	16	16	0	16	16	0
	Mäuse (NFD)	12	12	0	12	12	0

Results for Live Search of UNIX Data Set							
Case	Expected String	Active Files			Deleted Files		
		Expected	Hits	Misses	Expected	Hits	Misses
	infinity (Ligature)	12	12	0	12	12	0
	Mäuse (NFC)	12	12	0	12	12	0
	libertà (NFC)	12	12	0	12	12	0
	libertà (NFD)	12	12	0	12	12	0
	mañana (NFC)	12	12	0	12	12	0
FT-SS-07-RTL		12	12	0	12	12	0
	سكسكل ا	12	12	0	12	12	0
FT-SS-09-Doc		16	15	1	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	1	1	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-Hex		4	4	0	4	4	0
	panda	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

4.2.4 Meta-Data results for Live 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.

Meta-Data Results for Live Search of UNIX Data Set			
Case	**String**	**Partition**	**Seen**
FT-SS-07-CJK-char			
	中国	osxj	Yes
	中国	osxc	Yes
	中国	apfs	Yes
	東京	osxj	Yes
	東京	osxc	Yes
	東京	apfs	Yes
FT-SS-07-Cyrillic			
	Сибирь	osxj	Yes
	Сибирь	osxc	Yes
	Сибирь	apfs	Yes
FT-SS-07-NoBOM			
	لف الف	osxj	Yes
	لف الف	osxc	Yes
	لف الف	apfs	Yes
	Россия	osxj	Yes
	Россия	osxc	Yes
	Россия	apfs	Yes
	中國	osxj	Yes
	中國	osxc	Yes
	中國	apfs	Yes
FT-SS-07-RTL			
	سكسكل ا	osxj	Yes
	سكسكل ا	osxc	Yes
	سكسكل ا	apfs	Yes
FT-SS-09-Meta			

Meta-Data Results for Live Search of UNIX Data Set			
Case	String	Partition	Seen
	thunderbird	osxj	Yes
	thunderbird	osxc	Yes
	thunderbird	apfs	Yes
	thunderbird	ext4	Yes
	cañón	ext4	Yes

4.3 Unicode Normalization

The following is from "Unicode® Standard Annex #15, Unicode Normalization Forms." http://unicode.org/reports/tr15/

> Unicode Normalization Forms are formally defined normalizations of Unicode strings which make it possible to determine whether any two Unicode strings are equivalent to each other. Depending on the particular Unicode Normalization Form, that equivalence can either be a canonical equivalence or a compatibility equivalence.
>
> Essentially, the Unicode Normalization Algorithm puts all combining marks in a specified order and uses rules for decomposition and composition to transform each string into one of the Unicode Normalization Forms. A binary comparison of the transformed strings will then determine equivalence.
>
> The four Unicode Normalization Forms are summarized in *Table 1.*
>
> Table 1. Normalization Forms

Form	Description
Normalization Form D (NFD)	Canonical Decomposition
Normalization Form C (NFC)	Canonical Decomposition, followed by Canonical Composition
Normalization Form KD (NFKD)	Compatibility Decomposition

Normalization Form KC (NFKC)	Compatibility Decomposition, followed by Canonical Composition

END of REPORT