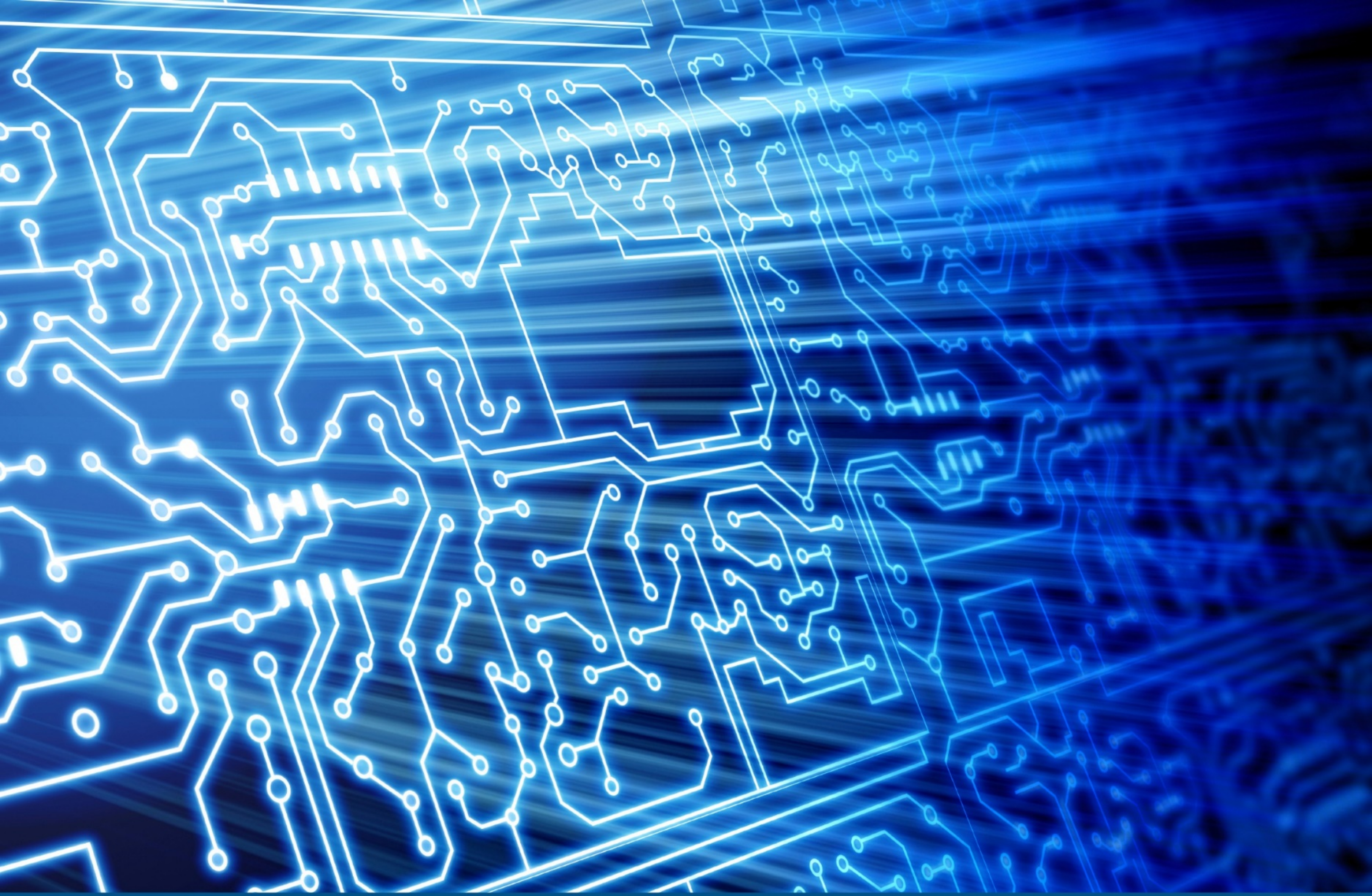

Test Results for String Search Tool:
Magnet Axiom Version 4.1.1.20153

September 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.

September 2020

Test Results for String Search Tool: Magnet Axiom Version 4.1.1.20153

Introduction

The Computer Forensics Tool Testing (CFTT) program is a joint project of the Department of Homeland Security (DHS) Science and Technology Directorate (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 Magnet Axiom Version 4.1.1.20153 using the CFTT Federated Testing Test Suite Version 5 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 test the tools that they use in their labs and 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&T'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. **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.

Test Results for String Search Tool: Magnet Axiom Version 4.1.1.20153

1 Tested Tool Description

Tool Name: Magnet Axiom
Tool Version: 4.1.1.20153
Vendor:

MAGNET Forensics
Herndon, VA 2250 Corporate Park Drive, Suite 130
Zip code: 20171
Phone: 1-844-638-7884

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 Magnet Axiom 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. One search engine was tested: Indexed Search.

The following features are not supported by Magnet Axiom Version 4.1.1.20153:

- Logical combinations (**and, or** and **not**); test cases 04, 05 & 06.
- *Substring*
- Built-in searches for phone numbers, email addresses and social security numbers; test cases 08-phone, 08-Email and 08-SS.
- Stemming search is not supported; test case 09-Stem.
- The tool does not support UTF-16 BE Big Endian.
- The tool does not support searching UNIX-like file system images.

2.2 Test Observations

We have the following observations*:*

- Only one anomaly was observed in testing:
  A string crossing a cluster boundary between non-contiguous clusters was not found.
- Observed misses for UTF-16 BE in this report do not reflect against the tool, they only appear in this report as a measure of test completeness.

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 (Magnet Axiom Version 4.1.1.20153) was installed on a Dell Latitude 7200 with 16GB installed RAM, running Microsoft Windows 10 Enterprise, Version 1903, OS Build 18362.

Testing was performed using CFTT Federated Testing Test Suite Version 5.0.

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 the https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/cftt-technical-1 CFReDS web site. The package includes two dd test image 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)

Case	Case Description
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 engine was 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 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**			**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

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
	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	12	6	18	12	6	6	4	2
	中国	9	6	3	9	6	3	3	2	1
	東京	9	6	3	9	6	3	3	2	1
FT-SS-07-CJK-hangul		9	6	3	9	6	3	3	2	1
	서울	9	6	3	9	6	3	3	2	1
FT-SS-07-CJK-kana		18	12	6	18	12	6	6	4	2
	スバル	9	6	3	9	6	3	3	2	1
	みつびし	9	6	3	9	6	3	3	2	1
FT-SS-07-Cyrillic		9	6	3	9	6	3	3	2	1
	Сибирь	9	6	3	9	6	3	3	2	1
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	30	9	39	30	9	13	10	3
	Россия	9	6	3	9	6	3	3	2	1
	فلافل	9	6	3	9	6	3	3	2	1
	中國	9	6	3	9	6	3	3	2	1
	QuarterHorse	12	12	0	12	12	0	4	4	0
FT-SS-07-Norm		75	51	24	75	51	24	25	17	8
	mañana (NFD)	9	6	3	9	6	3	3	2	1
	infinity (No Ligature)	12	9	3	12	9	3	4	3	1
	Mäuse (NFD)	9	6	3	9	6	3	3	2	1
	infinity (Ligature)	9	6	3	9	6	3	3	2	1

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
	Mäuse (NFC)	9	6	3	9	6	3	3	2	1
	libertà (NFC)	9	6	3	9	6	3	3	2	1
	libertà (NFD)	9	6	3	9	6	3	3	2	1
	mañana (NFC)	9	6	3	9	6	3	3	2	1
FT-SS-07-RTL		9	6	3	9	6	3	3	2	1
	س‌كسكلا	9	6	3	9	6	3	3	2	1
FT-SS-09-Doc		16	13	3	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	1	1	0	0	0	2	1	1
	nitroglycerin Formatted .docx	2	1	1	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	1	1	0	0	0	0	0	0
	Washington	1	0	1	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	2	2	4	2	2	0	0	0
	bear	4	2	2	4	2	2	0	0	0

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-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.

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.

Case	Comments on Indexed Search of Windows Data Set
FT-SS-01	All items Found
FT-SS-02	All items Found
FT-SS-03	All items Found
FT-SS-07-CJK-char	no utf-16-be Found
FT-SS-07-CJK-hangul	no utf-16-be found

Case	Comments on Indexed Search of Windows Data Set
FT-SS-07-CJK-kana	no utf-16-be found
FT-SS-07-Cyrillic	no utf-16-be found
FT-SS-07-Latin	All items Found
FT-SS-07-NoBOM	QuarterHorse all items found including 16-be no utf-16-be found for the other terms
FT-SS-07-Norm	no utf-16-be reported
FT-SS-07-RTL	no utf-16-be found
FT-SS-09-Doc	items not found: 9004 FMT-explosive-docx-win.docx 9005 FMT-dynamite-docxfmt-win.docx 9007 FMT-bolt-htmlfmt-win.html 9512 zipped zipped peroxide UTF 9513 zipped zipped nitroglycerin UTF 9515 500008338 976578 crossbow utf-8
FT-SS-09-Frag	item not Found: 6006 FRAG-fat-Olympia-split-32k.txt
FT-SS-09-Lost	All items Found
FT-SS-09-MFT	items not Found: 7010 (70È1 or 7a^@1) LIVE-MFT-Ursa-mft-utf-16-be.txt 7011 LIVE-MFT-Ursa-mft-utf-16-be.txt 7014 DELETED-MFT-Ursa-mft-utf-16-le.txt 7015 DELETED-MFT-Ursa-mft-utf-16-be.txt
FT-SS-09-Meta	All items found
FT-SS-10-Regex	All items found

4.2 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 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