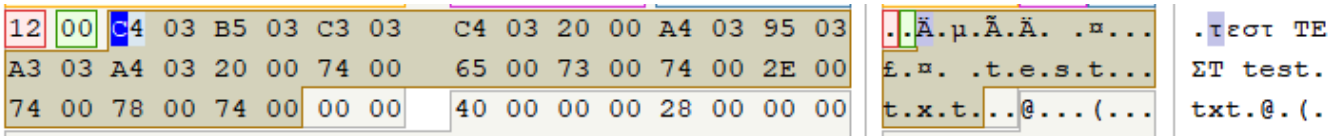
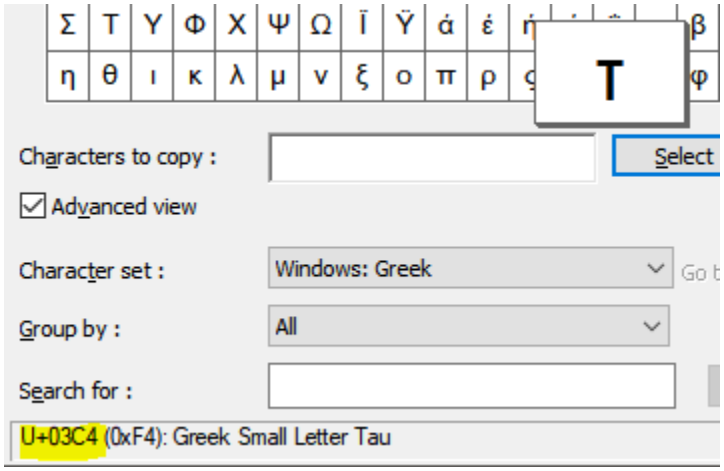
NTFS $FILE_NAME Attribute:

How ntfs translates Unicode characters in file names :
Here's a file I just created:



0xC403 represents the small greek letter "t" for example. How is that read/translated? In little endian it's 0x03C4 . Character map shows this:



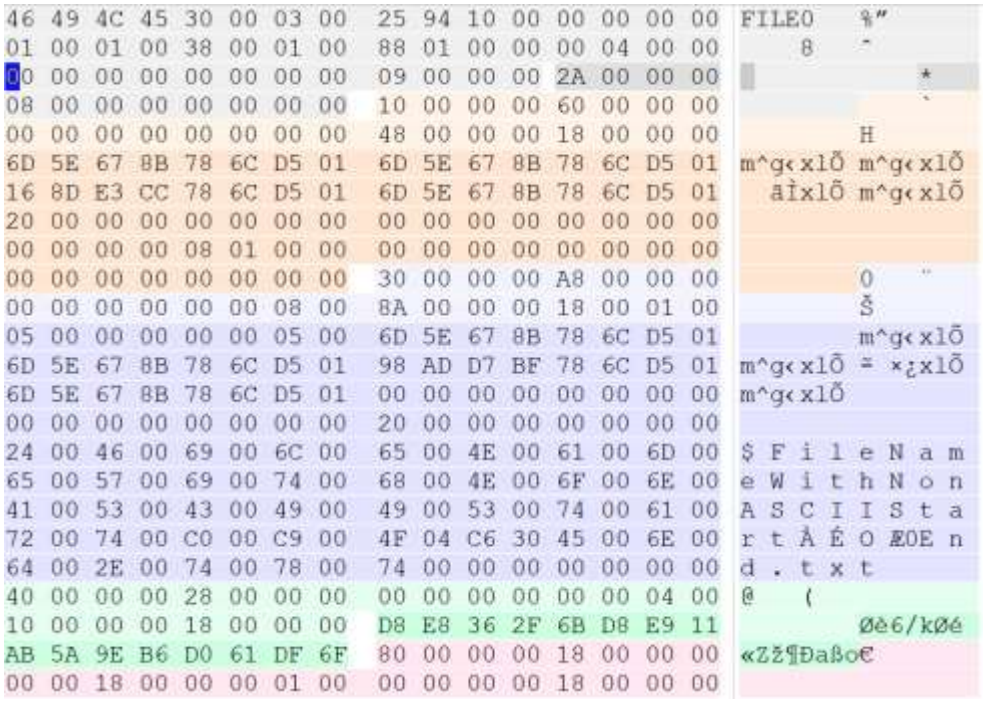
Every 2 bytes of the file name represent the Unicode escape sequence for a single Unicode character.

Or in other words, each file name character is represented by 2 bytes so

0x6500 is \u0065 or "e"
0x7400 is \u0074 or "t"
0x1920 is \u2019 or the character "᤺"
0xA303 is \u03A3 or the character "Σ" (*Greek capital letter S*)
etc ..

Here's a page for such conversions:
https://www.mobilefish.com/services/unicode_escape_sequence_converter/unicode_escape_sequence_converter.php

Let’s examine another file record:



```
46 49 4C 45 30 00 03 00  25 94 10 00 00 00 00 00  FILE0    %”
01 00 01 00 38 00 01 00  88 01 00 00 00 04 00 00      8    ˆ
00 00 00 00 00 00 00 00  09 00 00 00 2A 00 00 00                *
08 00 00 00 00 00 00 00  10 00 00 00 60 00 00 00                `
00 00 00 00 00 00 00 00  48 00 00 00 18 00 00 00            H
6D 5E 67 8B 78 6C D5 01  6D 5E 67 8B 78 6C D5 01  m^g‹xlÕ m^g‹xlÕ
16 8D E3 CC 78 6C D5 01  6D 5E 67 8B 78 6C D5 01    ãÌxlÕ m^g‹xlÕ
20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 08 01 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  30 00 00 00 A8 00 00 00            0    ¨
00 00 00 00 00 00 08 00  8A 00 00 00 18 00 01 00            Š
05 00 00 00 00 00 05 00  6D 5E 67 8B 78 6C D5 01            m^g‹xlÕ
6D 5E 67 8B 78 6C D5 01  98 AD D7 BF 78 6C D5 01  m^g‹xlÕ ˜ ×¿xlÕ
6D 5E 67 8B 78 6C D5 01  00 00 00 00 00 00 00 00  m^g‹xlÕ
00 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00
24 00 46 00 69 00 6C 00  65 00 4E 00 61 00 6D 00  $ F i l e N a m
65 00 57 00 69 00 74 00  68 00 4E 00 6F 00 6E 00  e W i t h N o n
41 00 53 00 43 00 49 00  49 00 53 00 74 00 61 00  A S C I I S t a
72 00 74 00 C0 00 C9 00  4F 04 C6 30 45 00 6E 00  r t À É O Æ0E n
64 00 2E 00 74 00 78 00  74 00 00 00 00 00 00 00  d . t x t
40 00 00 00 28 00 00 00  00 00 00 00 00 00 04 00  @    (
10 00 00 00 18 00 00 00  D8 E8 36 2F 6B D8 E9 11            Øè6/kØé
AB 5A 9E B6 D0 61 DF 6F  80 00 00 00 18 00 00 00  «Zž¶Ðaßo€
00 00 18 00 00 00 01 00  00 00 00 00 18 00 00 00
```

The Hex for the filename is:
0x460069006C0065004E0061006D00650057006900740068004E006F006E00410053004300490049005300740061007200740C000C9004F04C63045006E006400

Let’s split that into 2 byte groups, and convert it to Little Endian, and then to Unicode in Excel to see what the name is:

Commands used in Excel:

HEX2DEC (to convert from Hex to Decimal)
UNICHAR (to convert Decimal to Unicode character)

HEX	HEX (LE)	Unicode	Character
4600	0046	70	F
6900	0069	105	i
6C00	006C	108	l
6500	0065	101	e
4E00	004E	78	N
6100	0061	97	a
6D00	006D	109	m
6500	0065	101	e
5700	0057	87	W
6900	0069	105	i
7400	0074	116	t
6800	0068	104	h
4E00	004E	78	N
6F00	006F	111	o
6E00	006E	110	n
4100	0041	65	A
5300	0053	83	S
4300	0043	67	C
4900	0049	73	I
4900	0049	73	I
5300	0053	83	S
7400	0074	116	t
6100	0061	97	a
7200	0072	114	r
7400	0074	116	t
C000	00C0	192	À
C900	00C9	201	É
4F04	044F	1103	я
C630	30C6	12486	〒
4500	0045	69	E
6E00	006E	110	n
6400	0064	100	d

And it is correct:

Cluster No.: 21.343
$MFT (#42)
\FileNameWithNonASCIIStartÀÉя〒End.txt

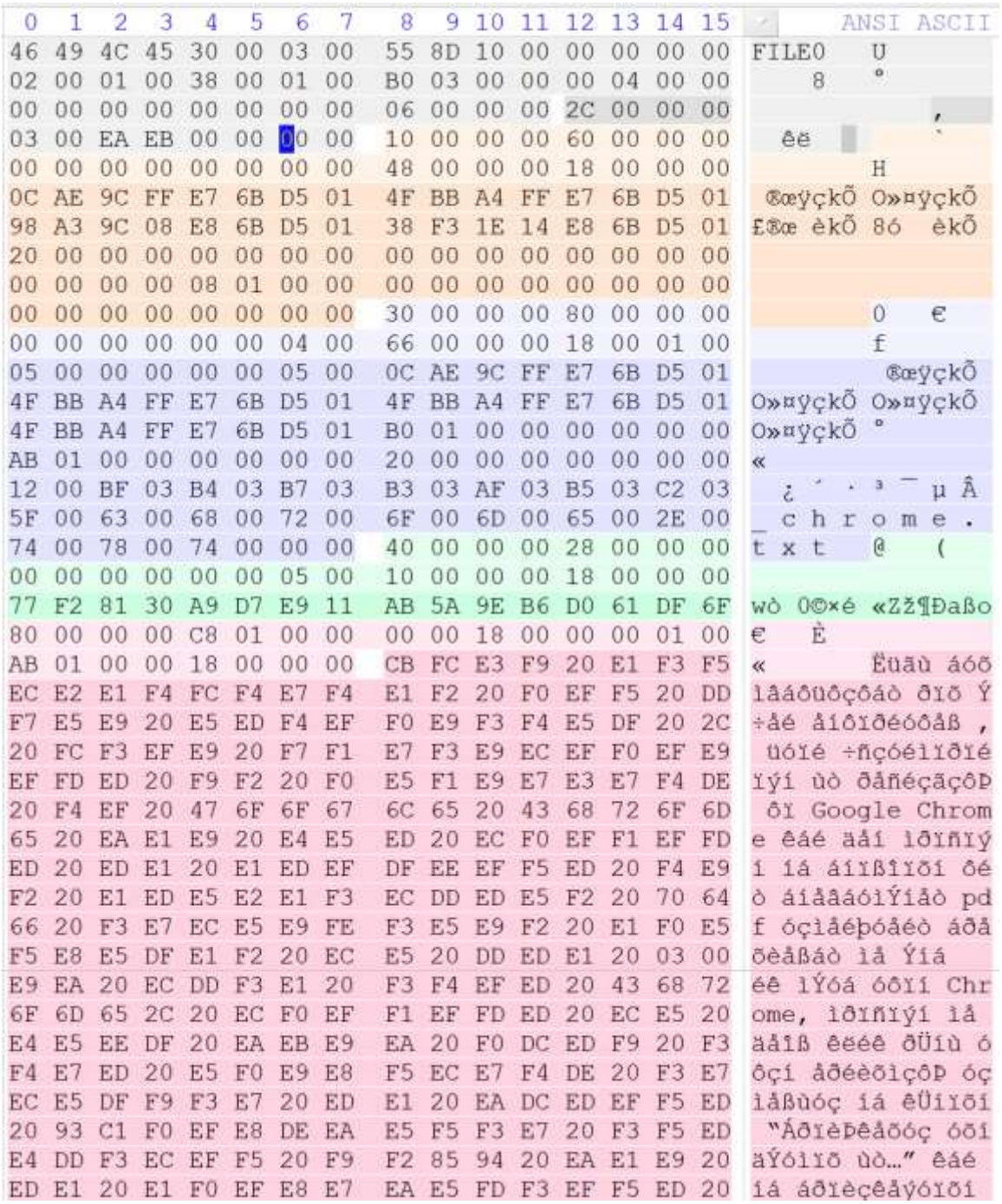
```
24 00 46 00 69 00 6C 00  65 00 4E 00 61 00 6D 00  $ F i l e N a m
65 00 57 00 69 00 74 00  68 00 4E 00 6F 00 6E 00  e W i t h N o n
41 00 53 00 43 00 49 00  49 00 53 00 74 00 61 00  A S C I I S t a
72 00 74 00 C0 00 C9 00  4F 04 C6 30 45 00 6E 00  r t À É я 〒 E n
64 00 2E 00 74 00 78 00  74 00 00 00 00 00 00 00  d . t x t
```

Same applies to ExFAT directory entries:

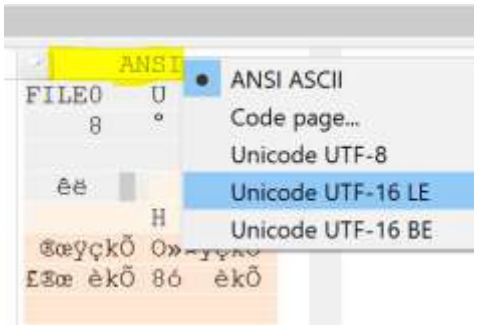
```
C1 00 46 00 69 00 6C 00  65 00 4E 00 61 00 6D 00  Á F i l e N a m
65 00 57 00 69 00 74 00  68 00 4E 00 6F 00 6E 00  e W i t h N o n
C1 00 41 00 53 00 43 00  49 00 49 00 53 00 74 00  Á A S C I I S t
61 00 72 00 74 00 C0 00  C9 00 4F 04 C6 30 45 00  a r t À É я 〒 E
C1 00 6E 00 64 00 2E 00  74 00 78 00 74 00 00 00  Á n d . t x t
```

The reason WinHex, HxD and other such programs display only Latin alphanumeric characters in the Hex/ASCII window, is that they read each bytes' s hexadecimal value and convert that to ASCII (first 256 Unicode characters in the form of \u00XX).
However, there is a way to display ASCII beyond the first 256 characters (extended set) in WinHex's Hexview.

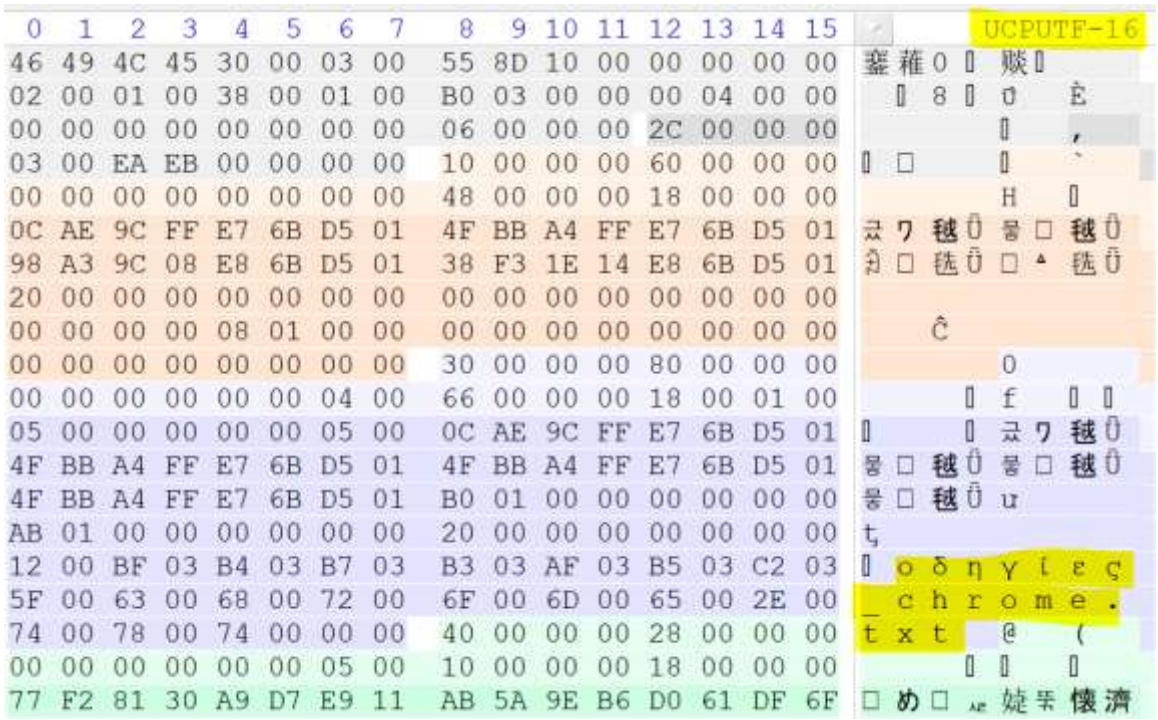
Let's say we want to see the FILENAME and contents of this resident FILE record:



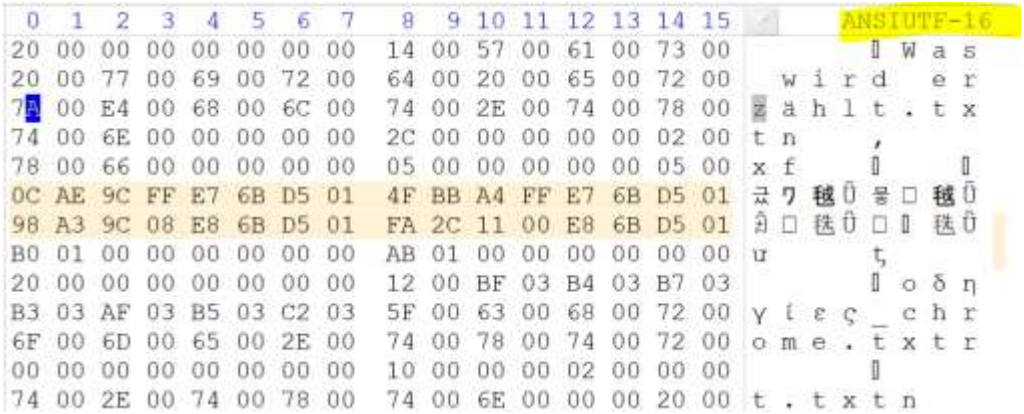
All we have to do is click on “ANSI ASCII” in the upper right corner:



And select “Unicode UTF-16 LE” (*LE for Little Endian*). The result will be that instead of the 0-256 char ASCII we now see the extended set and the actual filename:

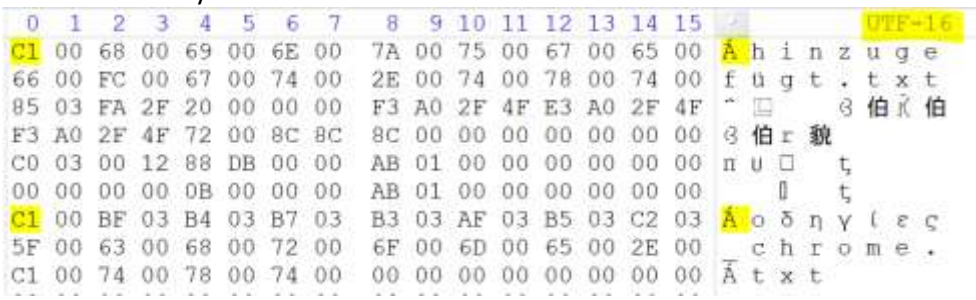


Same applies to NTFS directory entries:



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI	UTF-16
20	00	00	00	00	00	00	00	14	00	57	00	61	00	73	00		W a s
20	00	77	00	69	00	72	00	64	00	20	00	65	00	72	00	w i r d e r
7A	00	E4	00	68	00	6C	00	74	00	2E	00	74	00	78	00	z ä h l t . t x
74	00	6E	00	00	00	00	00	2C	00	00	00	00	00	02	00	t n ,
78	00	66	00	00	00	00	00	05	00	00	00	00	00	05	00	x f
0C	AE	9C	FF	E7	6B	D5	01	4F	BB	A4	FF	E7	6B	D5	01	碌 碌
98	A3	9C	08	E8	6B	D5	01	FA	2C	11	00	E8	6B	D5	01	碌 碌
B0	01	00	00	00	00	00	00	AB	01	00	00	00	00	00	00	ư ţ
20	00	00	00	00	00	00	00	12	00	BF	03	B4	03	B7	03	ο δ η
B3	03	AF	03	B5	03	C2	03	5F	00	63	00	68	00	72	00	γ ί ε ç _ c h r
6F	00	6D	00	65	00	2E	00	74	00	78	00	74	00	72	00	o m e . t x t r
00	00	00	00	00	00	00	00	10	00	00	00	02	00	00	00	
74	00	2E	00	74	00	78	00	74	00	6E	00	00	00	20	00	t . t x t n

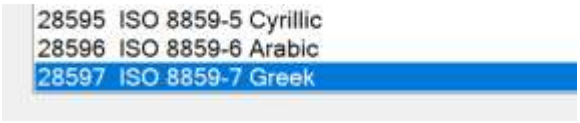
& ExFat directory entries



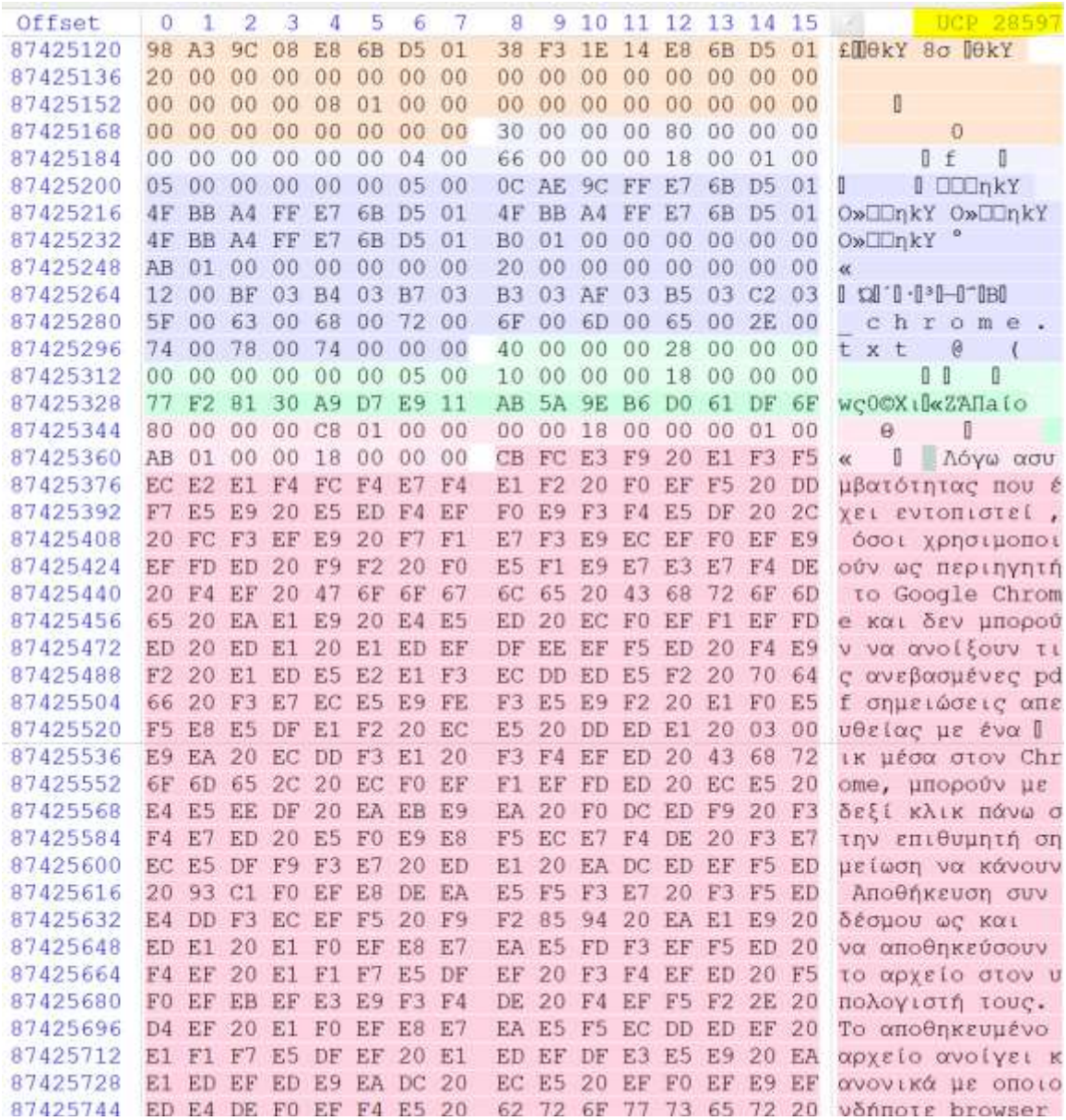
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	UTF-16
C1	00	68	00	69	00	6E	00	7A	00	75	00	67	00	65	00	h i n z u g e
66	00	FC	00	67	00	74	00	2E	00	74	00	78	00	74	00	f ü g t . t x t
85	03	FA	2F	20	00	00	00	F3	A0	2F	4F	E3	A0	2F	4F	伯 伯
F3	A0	2F	4F	72	00	8C	8C	8C	00	00	00	00	00	00	00	伯 r 貌
C0	03	00	12	88	DB	00	00	AB	01	00	00	00	00	00	00	ư ţ
00	00	00	00	0B	00	00	00	AB	01	00	00	00	00	00	00	ţ
C1	00	BF	03	B4	03	B7	03	B3	03	AF	03	B5	03	C2	03	ο δ η γ ί ε ç
5F	00	63	00	68	00	72	00	6F	00	6D	00	65	00	2E	00	_ c h r o m e .
C1	00	74	00	78	00	74	00	00	00	00	00	00	00	00	00	t x t

If instead we select Code page

and ISO 8859-7 for this file:



We can see the actual contents (*Resident text file in Greek*):



Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	UCP 28597
87425120	98	A3	9C	08	E8	6B	D5	01	38	F3	1E	14	E8	6B	D5	01	£ θkY 8σ θkY
87425136	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
87425152	00	00	00	00	08	01	00	00	00	00	00	00	00	00	00	00	
87425168	00	00	00	00	00	00	00	00	30	00	00	00	80	00	00	00	0
87425184	00	00	00	00	00	00	04	00	66	00	00	00	18	00	01	00	f
87425200	05	00	00	00	00	00	05	00	0C	AE	9C	FF	E7	6B	D5	01	ηkY
87425216	4F	BB	A4	FF	E7	6B	D5	01	4F	BB	A4	FF	E7	6B	D5	01	O»ηkY O»ηkY
87425232	4F	BB	A4	FF	E7	6B	D5	01	B0	01	00	00	00	00	00	00	O»ηkY °
87425248	AB	01	00	00	00	00	00	00	20	00	00	00	00	00	00	00	«
87425264	12	00	BF	03	B4	03	B7	03	B3	03	AF	03	B5	03	C2	03	¿´·³¯B
87425280	5F	00	63	00	68	00	72	00	6F	00	6D	00	65	00	2E	00	_ c h r o m e .
87425296	74	00	78	00	74	00	00	00	40	00	00	00	28	00	00	00	t x t @ (
87425312	00	00	00	00	00	00	05	00	10	00	00	00	18	00	00	00	
87425328	77	F2	B1	30	A9	D7	E9	11	AB	5A	9E	B6	D0	61	DF	6F	wς0©X«ZΑΠaίo
87425344	80	00	00	00	C8	01	00	00	00	00	18	00	00	00	01	00	θ
87425360	AB	01	00	00	18	00	00	00	CB	FC	E3	F9	20	E1	F3	F5	« Λόγω ασυ
87425376	EC	E2	E1	F4	FC	F4	E7	F4	E1	F2	20	F0	EF	F5	20	DD	μβατότητας που έ
87425392	F7	E5	E9	20	E5	ED	F4	EF	F0	E9	F3	F4	E5	DF	20	2C	χει εντοπιστεί ,
87425408	20	FC	F3	EF	E9	20	F7	F1	E7	F3	E9	EC	EF	F0	EF	E9	όσοι χρησιμοποι
87425424	EF	FD	ED	20	F9	F2	20	F0	E5	F1	E9	E7	E3	E7	F4	DE	ούν ως περιηγητή
87425440	20	F4	EF	20	47	6F	6F	67	6C	65	20	43	68	72	6F	6D	το Google Chrom
87425456	65	20	EA	E1	E9	20	E4	E5	ED	20	EC	F0	EF	F1	EF	FD	e και δεν μπορού
87425472	ED	20	ED	E1	20	E1	ED	EF	DF	EE	EF	F5	ED	20	F4	E9	ν να ανοίξουν τι
87425488	F2	20	E1	ED	E5	E2	E1	F3	EC	DD	ED	E5	F2	20	70	64	ς ανεβασμένες pd
87425504	66	20	F3	E7	EC	E5	E9	FE	F3	E5	E9	F2	20	E1	F0	E5	f σημειώσεις απε
87425520	F5	E8	E5	DF	E1	F2	20	EC	E5	20	DD	ED	E1	20	03	00	υθείας με ένα
87425536	E9	EA	20	EC	DD	F3	E1	20	F3	F4	EF	ED	20	43	68	72	ικ μέσα στον Chr
87425552	6F	6D	65	2C	20	EC	F0	EF	F1	EF	FD	ED	20	EC	E5	20	ome, μπορούν με
87425568	E4	E5	EE	DF	20	EA	EB	E9	EA	20	F0	DC	ED	F9	20	F3	δεξί κλικ πάνω σ
87425584	F4	E7	ED	20	E5	F0	E9	E8	F5	EC	E7	F4	DE	20	F3	E7	την επιθυμητή ση
87425600	EC	E5	DF	F9	F3	E7	20	ED	E1	20	EA	DC	ED	EF	F5	ED	μείωση να κάνουν
87425616	20	93	C1	F0	EF	E8	DE	EA	E5	F5	F3	E7	20	F3	F5	ED	Αποθήκευση συν
87425632	E4	DD	F3	EC	EF	F5	20	F9	F2	85	94	20	EA	E1	E9	20	δέσμου ως και
87425648	ED	E1	20	E1	F0	EF	E8	E7	EA	E5	FD	F3	EF	F5	ED	20	να αποθηκεύσουν
87425664	F4	EF	20	E1	F1	F7	E5	DF	EF	20	F3	F4	EF	ED	20	F5	το αρχείο στον υ
87425680	F0	EF	EB	EF	E3	E9	F3	F4	DE	20	F4	EF	F5	F2	2E	20	πολογιστή τους.
87425696	D4	EF	20	E1	F0	EF	E8	E7	EA	E5	F5	EC	DD	ED	EF	20	Το αποθηκευμένο
87425712	E1	F1	F7	E5	DF	EF	20	E1	ED	EF	DF	E3	E5	E9	20	EA	αρχείο ανοίγει κ
87425728	E1	ED	EF	ED	E9	EA	DC	20	EC	E5	20	EF	F0	EF	E9	EF	ανονικά με οποιο
87425744	ED	E4	DE	F0	EF	F4	E5	20	62	72	6F	77	73	65	72	20	νδήποτε browser

References:

"..the name of the file is encoded in UTF-16 Unicode.." (p. 318, File System Forensic Analysis, Brian Carrier).

6.4. **The file name attribute**
"..Name string Contains an UTF-16 little-endian without an end-of-string character.."

6.4.1. **Namespace**

Value	Identifier	Description
0	POSIX	Case sensitive character set that consists of all Unicode characters except for: \0 (zero character), / (forward slash). The : (colon) is valid for NTFS but not for Windows.
1	FILE_NAME_NTFS (or WINDOWS)	A case insensitive sub set of the POSIX character set that consists of all Unicode characters except for: " * / : < > ? \ | Note that names cannot end with a . (dot) or ' ' (space).
2	FILE_NAME_DOS (or DOS)	A case insensitive sub set of the WINDOWS character set that consists of all upper case ASCII characters except for: " * + , / : ; < = > ? \ Note the name must follow the 8.3 format.
3	DOS_WINDOWS	Both the DOS and WINDOWS names are identical **Which is the same as the DOS character set, with the exception that lower case is used as well.**

(https://github.com/libyal/libfsntfs/blob/master/documentation/New%20Technologies%20File%20System%20(NTFS).asciidoc)

Complete Character List for UTF-16
http://www.fileformat.info/info/charset/UTF-16/list.htm

Basic Multilingual Plane
"…One advantage of Unicode over other possible sets is that the first 256 code points are identical to ISO-8859-1, and hence also ASCII. In addition, the vast majority of commonly used characters are representable by only two bytes, in a region called the Basic Multilingual Plane (BMP). Now a character encoding is needed to access this character set, and as the question asks, I will concentrate on UTF-8 and UTF-16.."

https://en.wikipedia.org/wiki/Plane_%28Unicode%29#Basic_Multilingual_Plane