



Handling Special Characters

Author : Aravindan Karunakaran

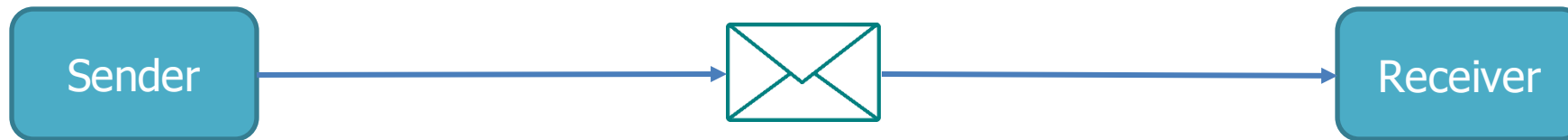
Date : 06-Mar-2023

Agenda

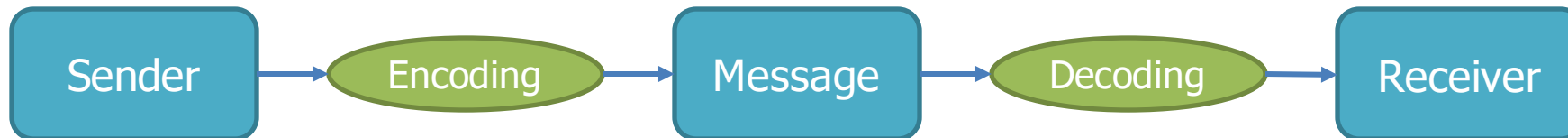
- Digital vs Physical Data
- Number systems
- What, why, how? – Handling Special characters
- Double-Byte Character sets (DBCS)

Digital vs Physical Data

Physical



Digital



Number systems

Decimal System	Binary System	Octal System	Hexadecimal System
0	00000000	000	0
1	00000001	001	1
2	00000010	002	2
3	00000011	003	3
4	00000100	004	4
5	00000101	005	5
6	00000110	006	6
7	00000111	007	7
8	00001000	010	8
9	00001001	011	9
10	00001010	012	A
11	00001011	013	B
12	00001100	014	C
13	00001101	015	D
14	00001110	016	E
15	00001111	017	F

Memory cell types

Cell types of storage

1 bit per cell	2 bits per cell	3 bits per cell	4 bits per cell	5 bits per cell
1	11	111	1111	
		110	1110	
		101	1101	
		100	1100	
			1011	
			1010	
			1001	
			1000	
			0111	
			0110	
			0101	
			0100	
			0011	
			0010	
			0001	
			0000	
SLC	MLC/DLC	TLC	QLC	PLC

What?

Basic Computer Keyboard – Special characters



Example of Special characters outside ASCII standard

“ ” ‘ ’ μ © ™ ®

ASCII standards

Control characters/
Non-printable
characters

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-`	63	3F	?	95	5F	_	127	7F	DEL

Acceptable ASCII
characters

Extended ASCII table

Character encoding

Windows-1252

UTF-8 (Unicode)

CP437

Windows-1252

Big5 (Chinese)

CP866 (Russian)

EUC-JP (Japanese)

EUC-KR (Korean)

GB 18030 (Chinese)

ISO-8859-1 (Latin1/Western European)

ISO-8859-2 (Latin2/Eastern European)

ISO-8859-3 (Latin3/South European)

ISO-8859-4 (Latin4/North European)

ISO-8859-5 (Latin/Cyrillic)

ISO-8859-6 (Latin/Arabic)

ISO-8859-7 (Latin/Greek)

ISO-8859-8 (Latin/Hebrew)

ISO-8859-10 (Latin6/Nordic)

ISO-8859-13 (Latin7/Baltic Rim)

ISO-8859-14 (Latin8/Celtic)

ISO-8859-15 (Latin9/Western European)

Character encoding

ISO-8859-1 (Latin1/Western European)

128 80 €	153 99 ™	179 B3 ³	205 CD Í	231 E7 ç
129 81 □	154 9A š	180 B4 ´	206 CE Î	232 E8 è
130 82 ,	155 9B >	181 B5 μ	207 CF Ĭ	233 E9 é
131 83 f	156 9C œ	182 B6 ¶	208 D0 Đ	234 EA ê
132 84 „	157 9D □	183 B7 ·	209 D1 Ñ	235 EB ë
133 85 ...	158 9E ž	184 B8	210 D2 Ò	236 EC ì
134 86 †	159 9F Ÿ	185 B9 ï	211 D3 Ó	237 ED í
135 87 ‡	160 A0	186 BA °	212 D4 Ô	238 EE î
136 88 ^	161 A1 ¡	187 BB »	213 D5 Õ	239 EF ï
137 89 ‰	162 A2 ¢	188 BC ¼	214 D6 Ö	240 F0 ð
138 8A Š	163 A3 £	189 BD ½	215 D7 ×	241 F1 ñ
139 8B <	164 A4 ¤	190 BE ¾	216 D8 Ø	242 F2 ò
140 8C Œ	165 A5 ¥	191 BF ¿	217 D9 Ù	243 F3 ó
141 8D □	166 A6 ¦	192 C0 À	218 DA Ú	244 F4 ô
142 8E Ž	167 A7 §	193 C1 Á	219 DB Û	245 F5 õ
143 8F □	168 A8 ¨	194 C2 Â	220 DC Ü	246 F6 ö
144 90 □	169 A9 ©	195 C3 Ã	221 DD Ý	247 F7 ÷
145 91 ’	170 AA ª	196 C4 Ä	222 DE Þ	248 F8 ø
146 92 ’	171 AB «	197 C5 Å	223 DF ß	249 F9 ù
147 93 ”	172 AC ¬	198 C6 Æ	224 E0 à	250 FA ú
148 94 ”	173 AD	199 C7 Ç	225 E1 á	251 FB û
149 95 •	174 AE ®	200 C8 È	226 E2 â	252 FC ü
150 96 –	175 AF ¯	201 C9 É	227 E3 ã	253 FD ý
151 97 —	176 B0 °	202 CA Ê	228 E4 ä	254 FE þ
152 98 ~	177 B1 ±	203 CB Ë	229 E5 å	255 FF ÿ
	178 B2 ²	204 CC Ì	230 E6 æ	

Why?

- Issue in line/page alignment, data spill over.
- Wrong statistics or count in output.
- Issue while executing programs.
- Issue while converting datasets to XPT files.
- P21 issue.

Example – why?

The diagram illustrates how special characters in data affect frequency counts. It shows a raw data table, a summary of the data, and the resulting frequency table from the FREQ procedure.

Obs	TESTTERM
1	Cough
2	Cough
3	Cough□
4	Cough
5	Cough□
6	Cough
7	Cough

Form Feed (points to the vertical line between observations 4 and 5)

Line Feed (points to the horizontal line between observations 4 and 5)

Special character seen as small box (points to the □ character in observations 3, 4, 5, and 6)

Incorrect frequency counts (points to the frequency counts in the FREQ table)

The FREQ Procedure

TESTTERM	Frequency	Percent
Cough	1	14.29
Cough	1	14.29
Cough□	1	14.29
Cough	2	28.57

How?

1. Deleting/Compressing all Non-printable characters (from 0 - 31 and 127).
2. Finding and replacing the Extended ASCII characters (from 128 - 256) with appropriate ASCII characters (from 32 - 126).
 - a) Scan and Print special characters in a file (special_char.txt) during first extraction.
 - b) Mention the replacement character in the file and use it to replace the special characters in consecutive extractions.

E.g. µg -> ug

Example – How?

- Compressing a specific Non-printable character.

```
labunit=compress(labunit,'09'x); *remove the horizontal-tab;
```

- Compressing most of the Non-printable character (some Non-printable characters can not be compressed with the following method E.g. 'A0').

```
varname=compress(varname,, 'kw');
```

- Replacing Extended ASCII character μ with u

```
labunit=tranwrd(labunit,'B5'x,'u'); *replace  $\mu$  with u;
```

Compressing all Non-printable and Extended ASCII characters

```
data &ds (drop=i npschars);  
  set &ds;  
  array charvars _character_;  
  length npschars $161; *161 is the total number of characters from 0 to 31, 127 to 255;  
  
  do i=0 to 31, 127 to 255;  
    if i=0 then npschars=byte(i);  
    else npschars=trim(npschars)||byte(i);  
  end;  
  
  do over charvars;  
    charvars=compress(charvars, npschars);  
  end;  
run;
```

Finding and Replacing all Extended ASCII Characters

```
data _td_&ds(drop=i);
  set &ds(keep=_character_);
  array charvars{*} _character_;
  length npschars $96; *96 is the total number of characters from 32 to 126;
  retain npschars;

  if _n_=1 then do;
    do i=32 to 126;
      if i=32 then npschars=byte(i);
      else npschars=trim(npschars)||byte(i);
    end;
  end;

  do i=1 to dim(charvars);
    char_vars(i)=compress(charvars(i),npschars);
  end;
run;
```

Finding and Replacing Extended ASCII Characters - Continued

```
ods exclude all;
ods output nlevels=_td_nlevels;
proc freq data=_td_&ds. nlevels;
    tables _character_ / noprint;
run;
ods exclude none;

%let varlist=; * list of variables with special characters;
proc sql noprint;
    select TableVar into :varlist separated by " "
    from _td_nlevels
    where nnonmisslevels ne 0;
quit;
```

Double-Byte Character sets (DBCS)

- Because East Asian languages have thousands of characters, double (two) bytes of information are needed to represent most characters.
- Proc contents can be used to determine the Encoding of the dataset.
- To work with datasets created in DBCS/MBCS encoding there are different versions of SAS editor available with different encoding options.
- UTF-8 is an DBCS/MBCS encoding that contains the Latin-script languages, Greek, Cyrillic, Arabic, and Hebrew. It also includes East Asian languages such as Japanese, Chinese, and Korean.



THANK YOU!

aravindan.karunakaran@ephicity.com

References

- <https://www.rapidtables.com/code/text/asciitable.html>
- <https://programmatically.com/understanding-memory-cells-in-computers>
- <https://www.lexjansen.com/pharmasug/2010/CC/CC13.pdf>