

# PX-Edit & PX-Job 3.3

## What's New?

# PX-Edit Mission

- PX-Edit **should** be able to
  - read every **valid** px table
  - detect most of all the possible errors (ie. all real-life ones)
  - repair majority of the detected problems
  - save standardized px tables which can be opened with other px family products
- New features are added when they are considered **feasible**, sensible and necessary
- PX-Edit is **feedbackware**

# Problems, Ideas, Feedback?

- I cannot fix a problem, or add a new feature without **knowing** it!
- When a problem occurs, no screenshots, please
  - **Ctrl+C** will copy the texts to clipboard
  - if the problem is repeatable with a certain file, fine!
  - send me the *EdBug\*.dws* (or *CONTINUE.dws*) files (zipped)
  - *px-edit\_33.main.ini* file (as .txt) might be needed as well
  - the PX-Job command (can be found in the log files)

# New features in PX-Edit 3.3

- Normal application evolution: speedups & bug fixes
- 64-bit version
- Value ordering can be changed with metadata *Import*
- *Value Find* [**Ctrl+Shift+F**]
- Support for huge Excel tables
- Keyword replace: the message shows the proposed new value
- Metaeditor|Keywords: *Apply for all values* switch in the Date window
- Metaeditor|Edit: *Save classification...* selection
- Use classification files when creating new tables
- *File/Save to/PX-Job-CSV*
- Support for *ISO-8859* codings
- *Settings/Check for Unicode*
- *Settings/Character conversion* menu

# New features in PX-Job 3.3

- 64-bit version
- Set html field width: `-w`
- Change path separator in report: `-b/`
- Character coding options: `-c10` .. `-c22`
- All variables to rows: `-o6`
- Use the system language for conversions: `!1`
- Update all values and languages for *CONTVARIABLE* with `-u`
- With px job and csv control file, the `-a` option may be omitted
  - the metadata change functionality has been revisited, e.g. value ordering

# Short and simple introduction to character encodings

- One character = octet (one byte) = 8 bits > 256 different values
- ASCII (values 0-127) has become a standard:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht	lf	vt	ff	cr	so	si
1	dle	dc1	dc2	dc3	dc4	nak	syn	etb	can	em	sub	esc	fs	gs	rs	us
2	sp	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	del

# The Standard: ISO-8859 coding

- Values 128-255 (“upper half”) depends on language
  - the *internal values* are the same, the *interpretation* is different
- The upper half area of ISO-8859-15 (Updated *Latin-1*):

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
8																
9																
a	nbsp	ı	ç	£	€	¥	Š	§	š	©	ª	«	¬	shy	®	¯
b	°	±	²	³	Ž	µ	¶	·	ž	¹	º	»	Œ	œ	Ÿ	ı
c	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
d	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
e	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
f	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

# Windows: WinANSI

- WinANSI codings are based on ISO standards, but not fully ☹
- The upper ASCII area of Windows-1252 (Western):

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
8	€		,	f	..	...	†	‡	^	%	Š	<	Œ		Ž	
9		‘	’	“	”	•	—	—	~	™	š	>	œ		ž	ÿ
a	nbsp	ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	shy	®	¯
b	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
c	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
d	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
e	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
f	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ



# Remembering the Bronze age: DOS codepages

- The upper half of the codepage-858 (Western):

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
8	Ç	ü	é	â	ã	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	õ	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	f
a	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
b					†	Á	Â	À	©	¶		¶	¶	¢	¥	¶
c	ℓ	⊥	⊥	⊥	—	†	â	Ã	ℓ	¶	⊥	⊥	⊥	=	⊥	□
d	ø	Ð	Ê	Ë	È	€	Í	Î	Ï	¶	¶	■	■	¶	Ì	■
e	Ó	β	Ô	Ò	ø	Õ	μ	þ	Ɔ	Ú	Û	Ü	ý	Ý	¬	'
f	shy	±	≡	¾	¶	§	÷	,	°	ˆ	.	¹	³	²	■	nbsp

# Unicode

- All the characters have a unique code point (max 21 bits)
- The bigger the code point, the less used character is
- Five file encodings: UTF-8 is the most common
- The Byte Order Mark heading (BOM) **should** be included
- The upper half of UTF-8 coding:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
a	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
b	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
c	2	2	latin	latin	latin	latin	latin	latin	latin	ipa	ipa	ipa	accent	accent	greek	greek
d	cyril	cyril	cyril	cyril	cyril	armen	hebrew	hebrew	arabic	arabic	arabic	arabic	syriac	arabic	thaana	n'ko
e	indic	misc	symbol	kana	cjk	cjk	cjk	cjk	cjk	cjk	asian	hangul	hangul	hangul	pua	forms
f	smp	smp	smp	ssp	spu	4	4	4	5	5	5	5	6	6		

# How to convert a px file during opening

- If the file contains the *Byte Order Mark* (BOM) header → **Unicode**
- Selecting *PC-Axis Unicode files* in the *File Open* window and the Unicode check is passed → **Unicode**
- *Character conversion/Ignore coding* is not set, and *CODEPAGE* is
  - `utf-8` and the Unicode check is passed → **Unicode**
  - valid coding → **ISO-8859, WinANSI or DOS**
- *Character conversion/Ignore coding* is set, or *CODEPAGE* is not recognised
  - *Settings/Check for Unicode* is set, and the check is passed → **Unicode**
- The language dependent coding
  - *Character conversion/ISO-8859* → **ISO-8859**
  - *Character conversion/DOS coding & CHARSET* is not `ANSI` → **DOS**
  - Otherwise → **WinANSI**

# How to convert a px file when saving

- *Settings/Save in Unicode* is set, or selecting *PC-Axis Unicode files* in the *File Save* window → **Unicode**
- If *CODEPAGE* exists and no *Character conversion* is set
  - `utf-8` → **Unicode**
  - a valid iso-8859 coding → **ISO-8859**
  - a valid WinANSI coding → **WinANSI**
- The language dependent coding:
  - *Character conversion* is set to *iso-8859* → **ISO-8859**
  - Otherwise → **WinANSI**

# Conversion notes

- The language dependent codes are based on the main table language
  - the system language will be used if *Character conversion/System language coding* is set, or there is no *LANGUAGE* setting in the px file
  - if the code cannot be deducted, the corresponding Western code will be used instead
- The conversion codes may be checked with **Ctrl+double click** on the lower right field
- When saving
  - the Unicode files will be saved with the Byte Order Mark (BOM) header
  - *CODEPAGE* will be set as the used conversion code
  - *CHARSET* will be set as ANSI
  - DOS conversion is not supported