

Was Entwickler über Zeichensätze, Unicode & Co. wissen sollten

Windows-1252 oder ISO-8859-1?
UTF-8? Unicode? Jeder Entwickler
sollte die Möglichkeiten der
Zeichensätze kennen, damit es mit
dem Austausch und der
Speicherung von Informationen
klappt. Und damit Ihnen niemand
ein ô für ein ä vormacht.

Stefan Heymann

Bevor es losgeht

- Handy aus?
- Fragen? Fragen!
- www.consic.de > Downloads > Talks

Inhalt

- Warum?
- Schriften, Schriftzeichen
- ASCII & Co.
- Unicode
- Anwendung

Warum?

Mussten Sie schon mal ...

- Content-Type bei HTML?
- encoding bei XML?
- Character Set bei InterBase/Firebird?
- NLS_LANG bei Oracle?

Haben Sie schon ...

- Spam aus Russland oder Korea bekommen und nur Fragezeichen gesehen?
- Tschechischen Text verarbeiten müssen?
- gewusst, dass es von Mörfelden bis Tschechien nur ca. 300 km Luftlinie sind?

Wachsende Anforderungen

- ASCII reicht doch ...
- Internationalisierung der Oberfläche
- Internationalisierte Websites
- Umgang mit fremdsprachlichen Texten
- Unsere Tool-Lieferanten (vorw. USA) sind nicht sensibilisiert, da sie es noch einfacher haben als wir

Joel on Software

So I have an announcement to make: if you are a programmer working in 2003 and you don't know the basics of characters, character sets, encodings, and Unicode, and I catch you, I'm going to punish you by making you peel onions for 6 months in a submarine. I swear I will.

-- Joel Spolsky

Schriften, Schriftzeichen

Schriftzeichen in Europa



Schriftzeichen der Welt



Glyph vs. Zeichen

A

A

A

A

A

A

A

Glyph, Zeichen, Zeichensatz

- Die Darstellung der Zeichen als Glyphen ist Aufgabe der Grafikausgabe (Postscript, GDI, TrueType, usw.)
- Wir kümmern uns vorwiegend um die Verarbeitung von Zeichen (Characters)
- Ein Zeichensatz kodiert Zeichen als numerische Werte

A = 65

Glyphen

- Nicht alle Sprachen stellen Glyphen von links nach rechts, in aufeinanderfolgenden Rechtecken dar
- Rechts nach Links, Oben nach unten
- Mehrere aufeinanderfolgende Zeichen können zu einem Glyphen verschmelzen

ASCII & Co.

Die Mutter aller Zeichensätze

- American Standard Code for Information Interchange: ASCII, ISO-646
- 7 Bit breit, also Zeichen von 0 bis 127.
- 32 Unsichtbare Steuerzeichen (NUL, CR, LF, FF, BEL, ESC, ...)
- Alphabet, Ziffern, Sonderzeichen (,.-:?)
- Ausgelegt für Englisch
- Daher keine Umlaute und Akzente

ASCII für Europa

- Umbelegung von selten gebrauchten Zeichen
- [= Ä \ = Ö] = Ü
- Problem: Drucker und Bildschirm müssen die selbe Einstellung haben
- Keine Mischung in einem Text möglich:
„Amélie knackt gerne die Kruste von Crème Brulé mit dem Löffel“

Nutzung des 8. Bit

- Dadurch 128 neue Positionen belegbar
- Es haben sich viele solche Zeichensätze entwickelt
- $\text{ISO 8859-x} = \text{ASCII} + 160..255$
- $\text{ISO 8859-1} = \text{Latin-1}$ (Westeuropäische Sprachen)
- $\text{Windows 1252} = \text{ISO 8859-1} + 128..159$

ISO 8859

- Die Zeichen 0..127 sind identisch mit ASCII
- Die Zeichen 128..159 sind unbelegte Steuerzeichen
- Zeichen 160..255 individuell belegt

ISO 8859-1 / Latin-1

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	í	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	­	®	¯
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
ø	ñ	õ	ö	ö	ö	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Afrikaans, Albanisch, Baskisch, Dänisch, Deutsch, Englisch, Färöisch, Finnisch, Französisch, Isländisch, Italienisch, Katalanisch, Niederländisch, Norwegisch, Portugiesisch/Brasilianisch, Rätoromanisch, Schottisches Gälisch, Schwedisch, Spanisch, Suaheli
Also fast überall – daher große Bedeutung

ISO 8859-2 / Latin-2

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	À	Á	Â	Ã	Ä	Å	Š	Ŝ	Š	Ť	Ž	-	Ž	Ž	
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
	à	á	â	ã	ä	å	š	ŝ	ŝ	ť	ž			ž	ž
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
	Ř	Ā	Ā	Ā	Ā	Ĺ	Č	Č	Ě	Ě	Ě	Ě	Ī	Ī	Ď
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
	Đ	Ń	Ń	Ō	Ō	Ō	×	Ř	Ů	Ů	Ů	Ů	Ÿ	Ÿ	ß
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
	ř	ā	ā	ā	ā	ĺ	č	č	ě	ě	ě	ě	ī	ī	ď
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
	đ	ń	ń	ō	ō	ō	÷	ř	ů	ů	ů	ů	ý	ţ	.

Zentraleuropa, Osteuropa (Tschechien, Polen, usw)

ISO 8859-3 / Latin-3

A0	A1	A2	A3	A4		A6	A7	A8	A9	AA	AB	AC	AD		AF
	Ĥ	Ĵ	£	℥		Ĥ	Š	..	İ	Š	Ĝ	Ĵ	-		Ž
B0	ħ	2	3	ˆ	μ	ĥ	.	ˆ	ı	Š	ğ	ĵ	¼		Ž
C0	Ā	Ā	Ā		Ä	Ĉ	Ç	È	É	Ê	Ë	Ĭ	Ĩ	Î	Ï
	Ñ	Ō	Ó	Ô	Ġ	Ö	×	Ĝ	Ŭ	Ū	Ŭ	Ü	Ŭ	Ŝ	ß
E0	ā	ā	ā		ä	ĉ	ç	è	é	ê	ë	ĭ	ĩ	î	ï
	ñ	ō	ó	ô	ğ	ö	÷	ğ	ŭ	ū	ŭ	ü	ŭ	ŝ	.

Südeuropa, Malta, Esperanto

ISO 8859-4 / Latin-4

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	À	Á	Â	Ã	Ä	Å	Š	Ÿ	Ž	Ē	Ģ	Č	–	Ž	–
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
°	à	á	â	ã	ä	å	š	ŷ	ž	ē	ģ	č	ð	ž	ŋ
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
Ā	Ā	Ā	Ā	Ā	Ā	Æ	ı	Č	Ě	Ě	Ě	Ě	İ	İ	İ
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
Ð	Ŋ	Ō	Ɔ	Ô	Õ	Ö	×	Ø	Ů	Ů	Ů	Ü	Ů	Ů	ß
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
ā	ā	ā	ā	ā	ā	æ	ı	č	ě	ě	ě	ě	ı	ı	ı
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
đ	ŋ	ō	ƙ	ô	õ	ö	÷	ø	ů	ů	ů	ü	ů	ū	.

Nordeuropa, Balten, Grönländisch, Sami

ISO 8859-5 / Cyrillic

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	–	Ў	Ў
B0	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
C0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю
D0	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
E0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю
F0	ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ	–	ў	џ

Kyrillisch (Russland, Ukraine, usw)

In der Praxis wichtiger: KOI8-R (Russisch), KOI8-U (Ukrainisch)

ISO 8859-6 / Arabic

A0				A4	هـ							AC	ء	AD	ـ	
											BB	ة				BF
	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	
	ع	ف	ق	ك	ل	م	ن	و	ى	ي	ث	ج	ح	خ	د	
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA						
ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ						
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	
ـ	ف	ق	ك	ل	م	ن	و	ى	ي	ي	ء	ء	ء	ء	ء	ء
F0	F1	F2														
ـ	ـ	ـ														

Arabisch

(enthält nicht alle Zeichen, wird daher nicht häufig eingesetzt)

ISO 8859-7 / Greek

A0	A1 ¸	A2 ´	A3 £			A6 ¡	A7 §	A8 ¨	A9 ©		AB «	AC ¬	AD −		AF −
B0 °	B1 ±	B2 ²	B3 ³	B4 ´	B5 µ	B6 ¸	B7 ·	B8 È	B9 É	BA Ì	BB »	BC Ò	BD ¼	BE Ý	BF Ñ
C0 Ì	C1 Á	C2 Β	C3 Γ	C4 Δ	C5 Ε	C6 Ζ	C7 Η	C8 Θ	C9 Ι	CA Κ	CB Λ	CC Μ	CD Ν	CE Ξ	CF Ο
D0 Π	D1 Ρ		D3 Σ	D4 Τ	D5 Υ	D6 Φ	D7 Χ	D8 Ψ	D9 Ω	DA Ì	DB ÿ	DC Ò	DD É	DE Ñ	DF Ì
E0 Ò	E1 α	E2 β	E3 γ	E4 δ	E5 ε	E6 ζ	E7 η	E8 θ	E9 ι	EA κ	EB λ	EC μ	ED ν	EE ξ	EF ο
F0 π	F1 ρ	F2 ς	F3 σ	F4 τ	F5 υ	F6 φ	F7 χ	F8 ψ	F9 ω	FA ì	FB ü	FC ò	FD ù	FE ñ	

Neugriechisch, Mathematik

ISO 8859-8 / Hebrew

A0		A2	¢	A3	£	A4	¤	A5	¥	A6	¦	A7	§	A8	¨	A9	©	AA	×	AB	«	AC	¬	AD	–	AE	®	AF	—
B0	°	B1	±	B2	²	B3	³	B4	´	B5	µ	B6	¶	B7	·	B8	¸	BA	÷	BB	»	BC	¼	BD	½	BE	¾		
																											DF		
																											=		
E0		E1		E2		E3		E4		E5		E6		E7		EA		EB		EC		ED		EE		EF			
F0		F1		F2		F3		F4		F5		F6		F7		F8		F9		FA									

Hebräisch (Konsonanten)

ISO 8859-9 / Latin-5

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	­	®	¯
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
Ğ	Ñ	Ō	Ȫ	Ȫ	Ȫ	Ȫ	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
ğ	ñ	ö	ö	ö	ö	ö	÷	ø	ù	ú	û	ü	ı	ş	ü

Türkisch

Windows-Zeichensätze

- Oft Deckungsgleich zu ISO-8859
- Zusätzliche Belegung der Zeichen 128..159 mit sichtbaren Zeichen
- Gedankenstrich, typografische Anführungszeichen, usw.
- Die Windows-Zeichensätze sind offiziell bei der IANA registriert

windows-1252

- Deckungsgleich mit ISO-8859-1
- Zusätzlich auf den Positionen 128..159:

€ , *f* „ ... † ‡ ^ ‰ Š ‹ Œ Ž
‘ ’ “ ” • — — ~ ™ š › œ ž Ÿ

€ erst seit 2000

Multi-Byte Character Sets MBCS

- Mehrere Bytes pro Zeichen
- Ostasiatische Sprachen (CJK)
- Stringlänge <> Länge der Zeichenkette
- Substring-Bildung schwieriger
- Werden von Delphi unterstützt

Unicode

Warum Unicode?

- Ein einziger Zeichensatz für alle Sprachen der Welt
- Keine Überschneidungen bei den Codes mehr
- Unabhängig von Hardware und Betriebssystem
- Zentrale Standardisierung ISO 10646

Unicode

- Erst 16 Bit pro Zeichen, später 32 Bit
- Darstellbare Zeichen: 1.114.112
- Vergeben ist derzeit nur ein Bruchteil davon
- Aktuelle Version: 4.0.1
- Definition der Zeichen, keine Glyphen
- Im Wesentlichen deckungsgleich zu ISO/IEC 10646

Zeichendefinition

- Unicode definiert für jedes Zeichen einen Zahlenwert (Skalar) und einen Identifier

0041	LATIN CAPITAL LETTER A
00E4	LATIN SMALL LETTER A WITH DIAERESIS
0391	GREEK CAPITAL LETTER ALPHA
05D0	HEBREW LETTER ALEF
0950	DEVANAGARI OM
1D56C	MATHEMATICAL BOLD FRAKTUR CAPITAL A

Unicode Code Points

- Codespace: 0..10FFFF
- Übliche Notation: hexadezimal mit vorangestelltem „U+“, nicht weniger als 4 Ziffern
- U+0020
- U+0041
- U+1D56C

Unicode Character Names

- Bestehen nur aus den Großbuchstaben A..Z, Ziffern 0..9, Bindestrich und Leerstellen
- BYZANTINE MUSICAL SYMBOL LEIMMA ENOS CHRONOU
- DESERET CAPITAL LETTER OW
- BRAILLE PATTERN DOTS-1245

Kodierung von Unicode

- Darstellung der Code Points im Speicher
- 32 Bits pro Zeichen wären i.d.R. unhandlich
- Daher haben sich verschiedene Kodierungen etabliert:
 - 8-Bit (UTF-8)
 - 16-Bit (UCS-2, UTF-16)
 - 32-Bit (UCS-4, UTF-32)

UCS-2

- 16 Bit pro Zeichen
- Darstellbarer Bereich: 0000..FFFF
= Basic Multilingual Plane (BMP)
- Seit Unicode 3.1 können nicht mehr alle Zeichen dargestellt werden
- Ersetzt durch UTF-16
- Der Begriff „Unicode“ wird oft synonym für UCS-2 verwendet

UTF-16

- 16 Bit pro Zeichen
- Einige Zeichen müssen als „Surrogat-Paar“ dargestellt werden und belegen dann 2 aufeinanderfolgende 16-Bit-Wörter
- Kompletter Codespace darstellbar
- Stringlängen-Ermittlung, Substrings erschwert

Endianness

- Problem bei UCS-2, UTF-16: Low/High-Byte-Reihenfolge
- Unterscheidung in UTF-16BE und UTF-16LE in den Metadaten
- Byte Order Mark (BOM) U+FEFF
- U+FEFF wird an den Anfang jedes Texts gesetzt
- U+FFFE ist und bleibt unbelegt

UCS-2 vs. UTF-16

- UTF-16 ist rückwärtskompatibel
- UCS-2 oft synonym mit „Unicode“
- WideString, wchar_t
- NT3, NT4: UCS-2
- Seit Windows 2000: UTF-16

UTF-8

- Kodierung als 8-Bit-Strings
- US-ASCII-Zeichen bleiben als solche erhalten, alle anderen belegen 2 bis 4 aufeinanderfolgende Bytes
- Kompletter Codespace darstellbar
- Vorteil: „Lateinische“ Texte recht kompakt
- Problem: Stringlängen, Substrings, usw.

UTF-8 Kodierung

- US-ASCII-Zeichen bleiben wie sie sind,
Bit 7 ist gelöscht 0xxxxxxx
- Alle andere sind Sequenzen von Bytes, bei
denen Bit 7 gesetzt ist 1xxxxxxx
- 1. Byte: Soviele gesetzte Bits am Anfang
wie Länge der Sequenz: 110xxxxx
- Folgebytes: Bit 7 gesetzt, Bit 6 gelöscht:
10xxxxxx

UTF-8 Kodierung

- Dadurch kann jedem Byte angesehen werden, ob es
 - ein komplettes Zeichen ist: 0xxxxxxx
 - Teil einer Sequenz ist: 1xxxxxxx
 - ein Sequenz-Start ist: 11xxxxxx
 - ein Sequenz-Folgebyte ist: 10xxxxxx

UTF-8 Kodierung

- Die Bits hinter der Kennung verbleiben für die Darstellung des Code Points

ä – LATIN SMALL LETTER A WITH DIAERESIS

$00E4_{16} = 11100100_2$

$110xxxxx \ 10xxxxxx$
 $ooo11 \ 100100$

11000011 10100100 bin

C3 A4 hex

195 164 dez

Ã æ Latin-1

00000 – 00007F:	0xxxxxxx
00080 – 0007FF:	110xxxxx 10xxxxxx
00800 – 00FFFF:	1110xxxx 10xxxxxx 10xxxxxx
10000 – 1FFFFF:	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

Es gibt nur ein' Rudi Völler



Tante Käthe

UCS-4, UTF-32

- 32 Bit pro Zeichen
- Direkte Darstellung jedes Codepoints als ein Speicherwort
- Sperrig aber praktisch (1 Wort = 1 Zeichen)
- Endianness-Problematik
- Kompletter Codespace darstellbar
- Nur akademische Unterschiede zwischen UCS-4 und UTF-32

Mörfelden

US-ASCII	M?rfelden
DE-ASCII	M rfelden
ISO 8859-1	Mörfelden
Windows 1252	Mörfelden
UTF-8	MÃ¶rfelden
UTF-16LE	M.ö.r.f.e.l.d.e.n
UTF-16BE	.M.ö.r.f.e.l.d.e.n
UTF-32	M...ö...r...f...e...l...d...e...n...

Was ist Plain Text?

- Zu jedem String, zu jedem Text (Datei, E-Mail, Attachment, Download, usw.) muss das Encoding bekannt sein
- Plain Text kann mit einer BOM beginnen

There Ain't No Such Thing As Plain Text.
-- Joel Spolsky

Encodings: Überblick

- ISO 8859-x
- Windows-12xx
- KOI8-R, KOI8-U
- Shift-JIS
- u. v. a. m.
- Unicode: UTF-8, (UCS-2), UTF-16, UTF-32
- Es gibt kein „Unicode“ als Encoding!

Unicode Besonderheiten

Zeichenbereiche

- General Scripts (Latin, Greek, Cyrillic, etc.)
- Symbols (Symbol, Dingbat, Punctuation, Math, etc.)
- CJK Phonetics and Symbols
- CJK Ideographs
- Yi Syllables, Hangul Syllables
- Surrogate Area, Private Use Area
- Compatibility and Specials

Steuerzeichen

- 0000..001F: Wie ASCII
- 007F: ASCII Delete/Rubout
- 0080..009F: Steuerzeichen

Kombinierte Zeichen

- Ein ä kann auf zwei Arten kodiert werden:
- 00E4: LATIN SMALL LETTER A WITH DIAERESIS
- 0061: LATIN SMALL LETTER A +
0308: COMBINING DIAERESIS
- Über solche „Combining Characters“ können verschiedene Zeichen zusammengesetzt werden

Normalisierung

- Problem: Text kann dann nicht mehr byteweise verglichen werden
- Normalisierung/Kanonisierung erforderlich
- Übersetzungstabellen
- Komposition, Dekomposition
- Technical Report #15: Unicode Normalization Forms

Bidirektionalität

- Unicode definiert die Ablage von Zeichen im Speicher
- Bei manchen Sprachen rechts- nach links
- Beispiel: Arabisch, Hebräisch
- Es werden verschiedene Regeln und Steuerzeichen für Rechts-Links und Links-Rechts-Text definiert

Transkodierung

- Umwandlung von speziellen Zeichensätzen in Unicode und zurück
- Windows-1252 → Unicode → ISO 8859-1
- Übersetzungs-Tabellen bei unicode.org
- Zeichen können verloren gehen
(ك wird zu ?)
- Zeichen können umgewandelt werden
(ç wird zu c)

Surrogat-Paare für UTF-16

- Zur Kodierung der Zeichen $> \text{FFFF}$
- Zeichenbereich D800..DFFF reserviert
- Es werden zwei 16-Bit-Worte belegt: High und Low Surrogate = Surrogate Pair
- High Surrogate: $\text{U}+\text{D800}..\text{U}+\text{DBFF}$
- Low Surrogate: $\text{U}+\text{DC00}..\text{U}+\text{DFFF}$

Surrogat-Paare

- Es kann einem 16-Bit-Wort also angesehen werden, ob es
 - ein nicht-Surrogat-Zeichen ist
 - zu einem Surrogat-Paar gehört
 - das High- oder das Low Surrogate ist

Sortierung

- Sortierungsregeln gelten auch bei Suche
- Es gibt kulturelle Unterschiede
 - ä wie a
 - ä wie ae
 - ä als eigenständiger Buchstabe nach z
- Unicode beschreibt Algorithmus und liefert Zeichen-Tabellen für Sortierung

Groß-/Kleinschreibung

- Gibt es nicht in jeder Sprache
- Nicht unbedingt reversibel
- Türkisch: $\text{ı} \rightarrow \text{I}$, $\text{i} \rightarrow \text{İ}$
- Nicht unbedingt 1:1 $\beta \rightarrow \text{SS}$
- Case Mappings sind sprachabhängig
- Unicode Technical Report #21
„Case Mappings“

Japanisch

- 私の名前はシュテファン ハイマンです。
- Watashi no namae wa shutefan haiman desu.
- Hiragana: Japanische Schriftzeichen
- Katakana: für fremdsprachliche Teile
- Romaji: Romanische Schriftzeichen, englische Sprechweise

Special Thanks to Yuzuri Kubota

Die Sonne geht im Osten auf

TAIYOU	WA	HIGASHI	KARA	NOBORU.
Die Sonne	(Partikel)	Osten	(von etwas)	aufgehen

太陽は 東から昇る。

TAIYOU WA HIGASHI KARA NOBORU.

Chinesisch, Koreanisch

- Chinesische vereinfachte Zeichen: 2 Komponenten: Bedeutungsgruppe + Laut
- mjen = Gesicht
- mjen + Essen = Nudeln
- Hangul: Koreanische Buchstaben, wie Alphabet

Unicode Standard

- Unicode Consortium www.unicode.org
- ISO 10646
- Buch: The Unicode Standard, Version 4.0
ISBN 0-321-18578-1
- Alle Kapitel und Files sowie komplette
Character Database bei www.unicode.org
- Technical Reports

Anwendung


Notepad

- Speichern unter ... > Codierung
 - ANSI (= Windows 1252)
 - Unicode (= UCS-2 bzw. UTF-16)
 - Unicode Big Endian
 - UTF-8

UltraEdit

- Unterstützung vorhanden
- Diverse Konvertierungen
- Insgesamt etwas schwach, keine klaren Begriffe

XML

- Jede XML-Entität ist definitionsgemäß Unicode. Default ist UTF-16 bzw. UTF-8.
- `<?xml version="1.0" encoding="utf-8" ?>`
- Jedes beliebige Unicode-Zeichen kann bei *jedem* Encoding als *Character Reference* eingefügt werden:
 - `€` €
 - `ॐ`  „DEVANAGARI OM“

HTML

- Unicode seit HTML 4.0
- Character References wie bei XML
- Im <head>:

```
<meta http-equiv="Content-Type"  
      content="text/html; charset=utf-8">
```

FrontPage

- Kann Seiten in verschiedenen Zeichensätzen speichern
- Auch UTF-8

Windows API

- Win32 API ist schon immer auf Unicode ausgelegt
- Alle Funktionen, die mit Strings arbeiten auch als „W“-Funktion verfügbar
- UCS-2 bis Windows NT 4.0
- UTF-16 ab Windows 2000

Delphi

- Die mitgelieferten VCL-Controls sind nicht Unicode-fähig, können aber mit dem lokalen Zeichensatz umgehen
- Troy Wolbrink: Tnt Delphi Unicode Controls
<http://tnt.ccci.org/>
- Mike Lischke (Virtual Treeview, etc.)
<http://www.delphi-gems.com/>
- JCL: JclUnicode.pas

Oracle

- Zeichensatz muss beim Anlegen einer Datenbank angegeben werden
- Client kann einen anderen Zeichensatz haben, wird dann automatisch umgewandelt
- Umgebungsvariable NLS_LANG
- WE8MSWIN1252, WE8ISO8859P1, UTF8, u.v. a. m.

InterBase/Firebird

- Verschiedene Zeichensätze (ISO8859_1, WIN1252)
- Unicode als Zeichensatz verwendbar
- Collations abhängig vom Zeichensatz
- `CREATE DATABASE 'employee.gdb'`
`DEFAULT CHARACTER SET ISO8859_1;`
- Für jeden CHAR/VARCHAR kann ein anderer Zeichensatz definiert werden

Danke!

Stefan Heymann

heymann@consic.de

EKON 8