



Internationalisierung von Anwendungen – Anforderungen und Erfahrungen in der Umsetzung



Georg Kistenberger
IBM SWG zIM TS Senior IT-Specialist
kistenberger@de.ibm.com

Agenda

- Einführung / Motivation
- Bildung einer neuen GSE Arbeitsgruppe BUNID
„**Internationalisierung von legacy Anwendungen**“
- Allgemeine Betrachtungen zur Einführung von Unicode
- Themen / Inhalte der Guide Tagung



Etablierung einer neuen GSE Arbeitsgruppe

■ Motivation

- Diskussion während der GSE DB2 Guide Tagung in Düsseldorf, Anfang April 2011
- Forderung / Notwendigkeit für einen durchgängigen Unicode support

■ Zielsetzung

- Informationsaustausch zum Thema "Internationalisierung von legacy Anwendungen"
- Sensibilisierung
- Bewertung von gesetzlichen Vorschriften im Hinblick auf die technischen Anforderungen für legacy Anwendungen
- Wer benötigt für welche Applikationen Unicode support?
 - Form und Umfang?
 - Gesetzliche Notwendigkeiten?
- Wo stehen wir?
- Welche Erfahrungen gibt es mit Unicode Projekten?
- Probleme und Lösungsansätze
- Requirements



Gesetzliche Vorschriften und Richtlinien

- **Verwendung von lateinischen Sonderzeichen in Personennamen**
 - Persönlichkeitsrecht
 - Namensrecht
 - Diskriminierung aufgrund von Sprache
 - Bundesgesetze / BGB, EU-/ und völkerrechtliche Verträge
- **Handhabung ist nicht klar geregelt**
- **Zunehmende Verwendung von diakritischen Zeichen in den Medien und Marketing**
- **Festlegung von neuen IT-Standards im öffentlichen Bereich und internationalen Zahlungsverkehr**
 - SEPA EU Richtlinien / ISO 20022 messaging standard:
[http://www.europeanpaymentscouncil.eu/content.cfm?page=iso_20022_message_standards_\(sepa_data_format\)](http://www.europeanpaymentscouncil.eu/content.cfm?page=iso_20022_message_standards_(sepa_data_format))
- **Deutschland-Online**
 - Einheitliche e-Government Infrastruktur; verstärkte Internet Nutzung
 - <http://de.wikipedia.org/wiki/Deutschland-Online>
 - <http://www.studie-deutschland-online.de/>
 - SAGA – Standards und Architekturen e-Government Anwendungen
 - Basis ist Unicode (UTF-8)
 - http://www.cio.bund.de/DE/Standards/SAGA/saga_node.html

Gesetzliche Vorschriften und Richtlinien

▪ Datenaustauschstandards für

- **Ausländerbehörden – XAusländer**
- **Meldebehörden – XMeld**
- **Personenstandswesen – XPersonenstand**
- Ursprünglicher Termin: 1.11.2011; **Aktueller Termin: 1.11.2012**
- XML (string.latin) und UTF-8 zum Datenaustausch und Speicherung
 - Lateinische Schrift als Basis für diakritische Zeichen
 - ISO/IEC 10646:2003
- Spezifikation XAusländer 1.3.1
 - <http://www1.osci.de/sixcms/media.php/13/xauslaender131Spezifikation.pdf>
- Veröffentlichung vom Bundesministerium des Innern
 - https://www.ebundesanzeiger.de/ebanzwww/wexsservlet?page.navid=bookmark_official_sitetobookmark_official_siteprint&genericsearch_param.edition=eBAnz+AT10+2011
- BZSt – Bundeszentralamt für Steuern
- Bundesverwaltungsamt – Bundesstelle für Informationstechnik
 - Deutsches Verwaltungsdienstverzeichnis:
http://www.cio.bund.de/SharedDocs/Publikationen/DE/CeBIT/infoheft_dvdv_download.pdf?__blob=publicationFile

▪ Personalausweis und Reisepass

- Bundesdruckerei verwendet die lateinischen Untermenge des Unicode Zeichensatzes ISO 10646
- Elektronischer Personalausweis
 - Dienste und Anbieter:
 - http://www.personalausweisportal.de/DE/Neue-Moeglichkeiten/neue-moeglichkeiten_node.html
 - DATEV: Integration in das Identity- und Access Management System der DATEV zum Schutz des Zugriffs auf online bereitgestellte persönliche Daten, Informationen und Bescheinigungen.
 - BSI stellt AusweisApp zur Verfügung:
 - http://www.personalausweisportal.de/SharedDocs/Kurzmeldungen/DE/2010/AusweisApp_ab_3_Januar_fuer_Buerger.html?nn=830452

Konstituierende Sitzung bei der GAD!



GSE-Working Group BUNID Internationalisierung von Legacy Anwendungen

**Kick Off-Tagung
21.11.2011**

© GAD 2011

BUNID Tagungen

■ Wahl des Chairman Boards

- Chairman: Claus Weidenfeller – GAD
- Stellv. Chairman: Dr. Jörg Ortmann – TK
- IBM Liason: Georg Kistenberger

■ Verabschiedung der neuen GSE Working Group BUNID

- Abstract – Statuten

■ Projekterfahrungen

- Unicode Einführung bei einem Kunden im öffentlichen Bereich
 - CICS / COBOL / DB2 durch IBM/BP
 - COBOL Umsetzung nach Unicode
 - DB2 Datenbank bereits in Unicode vorhanden
- Unicode4TKeasy bei der Techniker Krankenkasse
 - Unicodeumsetzung anhand eines Business Cases

■ Produktbetrachtung / Unterstützung für Unicode

- Unicode Internals
- Standardisierung von Unicode Subsets
- Unicode support in Host Languages
- Unicode support in Assembler
- Unicode support beim AFP Druck

■ Erfahrungsaustausch ist „key“ !!!

Unicode Transformation Formats

- **Pre-Unicode:** , ASCII, ISO8859-x and code pages
- **Unicode consortium starting from X/OPEN in 1992 (Microsoft, Apple, IBM....)**
 - www.unicode.org/charts/
- **UCS-2: ISO10646 Universal Character Set**
- **UTF-16: Unicode Enhancement for UCS-2, variable coding**
- **UTF-8: Consecutive one-byte-encoding with variable length**
 - UTF: **U**nicode **T**ransformation **F**ormat
- **UCS-4, UTF-32: 4-byte code, fixed format**
 - practically not used at the moment

UCS-2 and UTF-16

■ ISO Standard UCS-2

- Fixed 2-byte presentation of character
- ASCII unchanged values X'00' – X'7F' -> X'00xx', e.g. X'0033' (Character '3')
- 65536 values with some restricted ranges

■ Unicode Standard UTF-16

- Extension of UCS-2
- Variable 2 or 4 bytes coding
- 2 bytes for range X'0000' – X'FFFF' (basic multilingual plane 'BMP')
- 4 bytes for range X'1 0000' – X'10 FFFF'
- 1 114 112 characters available
- Coding with surrogate range X'D800' – X'DFFF' (unused in UCS-2)
- BMP0 – Basic Multilingual Plane (63488 codepoints)
 - $65536 - 63488 = 2048$ surrogates

Unicode Array BMP plane 0

- 0000 -007F Basic Latinrelates to ASCII
- 0080 -00FF Latin-1 Supplement Corresponds to Latin1 Codepage including Basic latin
- 0100 -017F Latin Extended-A Extensions for european languages
- 0180 -024F Latin Extended-B Other latin special characters
- 0250 -02AF IPA Extensions.....
- 02B0 -02FF Spacing ModifierLetters.....
- 0300 -036F Combining DiacriticalMarks ..
- 0370 -03FF Greek and Coptic.....
- 0400 -04FF Cyrillic.....
- 0500 -052F CyrillicSupplement
- 0530 -058F Armenian.....
- 0590 -05FF Hebrew.....
- 0600 -06FF Arabic.....
- 0700 -074F Syriac.....
- 0750 -077F Arabic Supplement
- 0780 -07BF Thaana.....
- 07B0 -08FF N'Ko.....

<http://www.unicode.org/Public/UNIDATA/Blocks.txt>
 U+0000 –U+FFFF Basic Multilingual Plane (BMP, Plane 0)

Unicode Array BMP planes (UTF-16)

- **0000 - FFFF Basic Multilingual Plane**
- **10000 – 1FFFF Supplementary Multilingual Plane**
- **20000 – 2FFFF Supplementary Ideographic Plane**
- **30000 – 3FFFF Tertiary Ideographic Plane (currently unassigned)**
- **40000 – DFFFF unassigned**
- **E0000 – EFFFF Supplementary Special-purpose Plane**
- **F0000 – FFFFF Supplementary Private Use Area-A**
- **100000 – 10FFFF Supplementary Private Use Area-B**

Unicode newest Blocks

The newly-defined blocks in Version 6.1 (Feb 2012) are:

08A0.. 08FF	Arabic Extended
A1CC0.1CCF	Sundanese Supplement
AAE0.. AAFB	Meetei Mayek Extensions
10980..1099F	Meroitic Hieroglyphs
109A0..109FF	Meroitic Cursive
110D0..110FF	Sora Sompeng
11100..1114F	Chakma
11180..111DF	Sharada
11680..116CF	Takri
16F00..16F9F	Miao
1EE00..1EEFF	Arabic Mathematical Alphabetic Symb

UTF-8

- **ASCII Safe UNICODE (maps to 7-Bit ASCII)**
 - Bytes '00'x - '7F'x = 7-Bit ASCII
 - represented by single byte chars
- **Chars above '80'x are encoded by 2-4 byte chars**
 - Most characters take 2-3 bytes
 - Most CJK characters take 3 bytes
 - Most Extended Latin characters take 2 bytes

Codepoints	1st Byte	2nd Byte	3rd Byte	4th Byte
0000..007F	00..7F			
0080..07FF	C2..DF	80..BF		
0800..0FFF	E0	A0..BF	80..BF	
1000..FFFF	E1..EF	80..BF	80..BF	
10000..3FFFF	F0	90..BF	80..BF	80..BF
40000..FFFFFF	F1..F3	80..BF	80..BF	80..BF
100000..10FFFF	F4	80..8F	80..BF	80..BF

Unicode Format Samples

Character	ASCII	UTF-8	UTF-16	UTF-32
A	'41'x	'41'x	'0041'x	'00000041'x
a	'61'x	'61'x	'0061'x	'00000061'x
9	'39'x	'39'x	'0039'x	'00000039'x
Å (The character A with Ring accent)	'C5'x	'C385'x Note: 'C5'x becomes double byte in UTF-8	'00C5'x	'000000C5'x
顛 U+9860	'CDDDB'x (CCSID 939)	'E9A1A0'x	'9860'x	'00009860'x
𪛇 U+200D0	N/A	'F0A08390'x	'D840DCD0'x	'000200D0'x

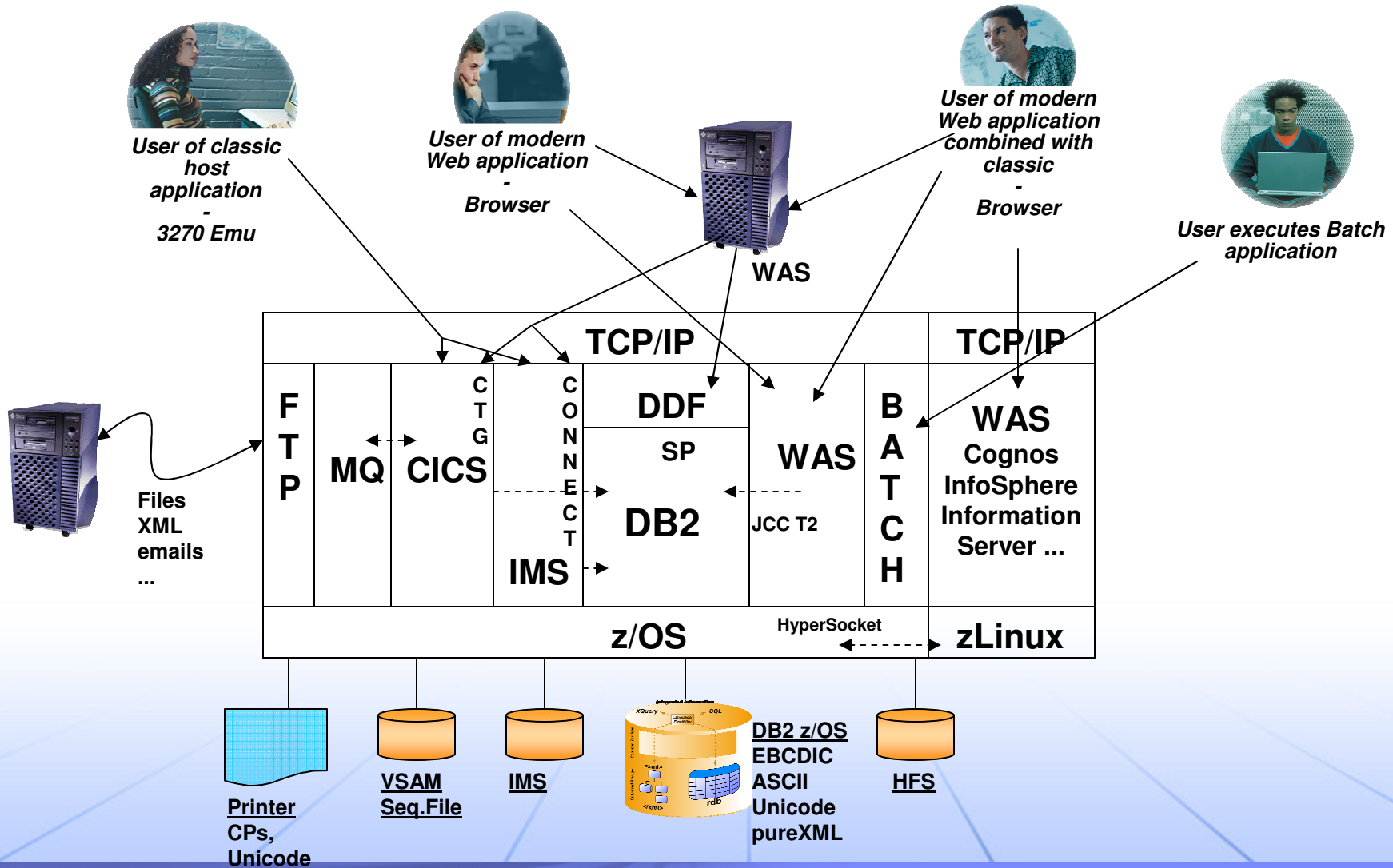
Note: UCS-2/UTF-16 and UCS-4/UTF-32 are using a technique called Zero Extension

Unicode Voraussetzungen

■ **Verlustfreie Unicode Verarbeitung**

- Alle Komponenten einer Anwendung benötigen Unicode support?!
 - Server
 - Programme
 - Datenbanken
 - Dateien
 - Druckertreiber und Drucker
- Datenübertragung
 - Alle Partner müssen unicode-fähig sein
- Unicode per default
 - HTTP, SOAP, HTML, WML, XML, Java, PDFs

Architektur zur Internationalisierung



Allgemeine Betrachtungen & Fragestellungen – 1 -

Die deutsche Codepage 1141 (DB2) gilt für Westeuropa:

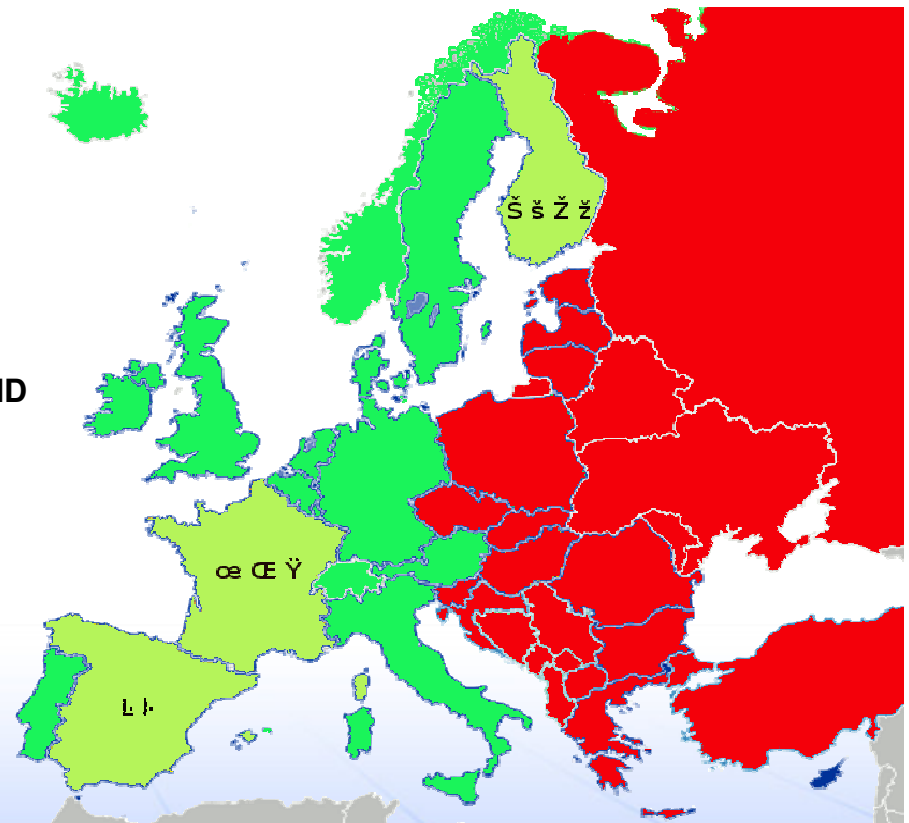
- Alle grün gekennzeichneten Länder können eingebunden werden
- Für spezielle Sonderzeichen gibt es keine Darstellung

Vorteile:

- Kein/minimaler Aufwand für DB2 Anwendungen
- Application Encoding Scheme auf Plan/Package-Ebene
- SQL DECLARE VARIABLE statement
- abgestimmt mit Precompile CCSID und Coprocessor SQLCCSID

Nachteile:

- Eine Erweiterung in den osteuropäischen Raum oder andere Länder ist nicht möglich



Codepage Deutschland - Frankreich

- Codepage Conversion stellt eine korrekte Zeichenumsetzung sicher

CP1141

HEX DIGITS 1ST → 2ND ↑	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	SP	&	-	ø	Ø	°	μ	¢	ä	ü	Ö	0
-1	(RSP)	é	/	É	a	j	ß	£	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	{	ë	[Ë	c	l	t	·	C	L	T	3
-4	à	è	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v	@	E	N	V	5
-6	ã	ï	Ã	Ï	f	o	w	¶	F	O	W	6
-7	ä	ī	Ä	Ï	g	p	x	¼	G	P	X	7
-8	ç	i	Ç	Ì	h	q	y	½	H	Q	Y	8
-9	ñ	~	Ñ	`	i	r	z	¾	I	R	Z	9
-A	Ä	Ü	ö	:	«	»	ı	¶	ö	ı	2	3
-B	.	\$,	#	»	»	ı	ı	ö	ı	Ö	Ü
-C	<	*	%	§	ö	æ	Đ	ı	ı	ı	ı	ı
-D	()	_	'	ý	ı	ı	ı	ı	ı	ı	ı
-E	+	;	>	=	þ	Æ	þ	'	ó	ú	Ó	Ú
-F	!	^	?	"	±	€	×	ö	ý	Ö	Û	Û

CP1147

HEX DIGITS 1ST → 2ND ↑	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	SP	&	-	ø	Ø	[`	¢	é	è	ç	0
-1	(RSP)	{	/	É	a	j	-	#	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4	@	}	À	È	d	m	u	©	D	M	U	4
-5	á	í	Á	Í	e	n	v]	E	N	V	5
-6	ã	ï	Ã	Ï	f	o	w	¶	F	O	W	6
-7	ä	ī	Ä	Ï	g	p	x	¼	G	P	X	7
-8	\	i	Ç	Ì	h	q	y	½	H	Q	Y	8
-9	ñ	ß	Ñ	μ	i	r	z	¾	I	R	Z	9
-A	°	§	ù	:	«	»	ı	¶	ö	ı	2	3
-B	.	\$,	£	»	»	ı	ı	ö	ı	Ö	Ü
-C	<	*	%	à	ö	æ	Đ	ı	ı	ı	ı	ı
-D	()	_	'	ý	ı	ı	ı	ı	ı	ı	ı
-E	+	;	>	=	þ	Æ	þ	'	ó	ú	Ó	Ú
-F	!	^	?	"	±	€	×	ö	ý	Ö	Û	Û

Allgemeine Betrachtungen & Fragestellungen – 2 -

■ In welchen Umfang soll Unicode genutzt werden?

- Beschränkung auf Unicode subsets
 - Lateinische Zeichen in allen Varianten?
 - Kyrrillische, griechische ... Zeichen?
 - Eingabe: Sonderzeichenmenues, customized Tastaturbelegungen, autohotkey Programm ...
 - Prüfroutinen
 - ISO6937 (327 Zeichen); WGL4 (652 Zeichen), XÖV (492 Zeichen), CEN/TC304 EES: Extended European Subset (ISO IEC 10646-1, 3109 Zeichen) ...
 - Vorgaben
- Beschränkung auf einzelne Use/Business Cases
 - Welche Anwendungsanforderung muss erfüllt werden?
 - Sicherstellen das die Post ankommt? Erfüllung einer Richtlinie? ...

■ Koexistenz von Daten mit ASCII/EBCDIC und Unicode Speicherung

- Ggf. Start mit der Datenbank DB2 ≥
 - Anwendung bedient zunächst Untermenge an Zeichen
 - Automatische Konvertierung zwischen DB2 und Anwendung
 - Application encoding scheme / Declare Variable SQL statement
 - Joins zwischen EBCDIC / ASCII und Unicode Tabellen möglich
 - Jedoch keine Unterstützung von RI-Definitionen mit string datatypes
 - Requirement MR012908405 / GGDB208012
 - Encoding scheme support auf Tablespace (Tabellen) Ebene
 - Requirement für Unicode Column-level support bzw. Unicode string datatype
 - MR122011716 / GGUNI11001 ...

Allgemeine Betrachtungen & Fragestellungen – 3 -

■ UTF-8 vs UTF-16

- Host languages, Java nutzen UTF-16
- DB2 unterstützt UTF-8 / UTF-16
 - UTF-8 bedeutet variable Länge (i.d.R. 1 – 3 bytes)
 - UTF-16 i.d.R. „feste“ Länge (2-bytes)
 - vereinfachtes Datenmodell
- Internet Technologie (WAS, HTML, XML ...) i.d.R. per default UTF-8 based
- Deutsche Textdokumente i.d.R. mit sehr geringen Speichermehrbedarf in UTF-8

Allgemeine Betrachtungen & Fragestellungen – 4 -

- **Suchen nach diakritischen Zeichen**
 - Mehrfachspeicherung
 - Transliteration & Original ≥
 - Volltextsuche
 - SQL BIF NORMALIZE_STRING
 - Unicode Normalization Services for z/OS ≥
- **Suchfunktionen im IMS DB über SSA (Segment Search Area) sind nur eingeschränkt möglich**
 - Equal ist o.k.; Range Abfragen jedoch nicht
- **Landesspezifische Sortierung**
 - Unicode Collation Services for z/OS
 - SQL BIF COLLATION_KEY
- **Umgebungen die keine codepages kennen (z.B. TSO nutzt PComm Codepage)**
- **3270-Emu / PComm ermöglicht keinen Unicode support**

Allgemeine Betrachtungen & Fragestellungen – 5 -

- **Werden eigene Codepages in Applikationen verwendet?**
 - Selbst-entwickelte Konvertierungsroutinen
 - Umsetzen von Zeichen innerhalb eines Programmes
- **Werden Bit-Daten in string Datentypen abgelegt?**
 - Sollten in CHARACTER FOR BIT DATA oder VAR/BINARY datatypes gespeichert sein
- **Welche codepages werden heute verwendet?**
- **Wie sieht die Anwendungsarchitektur aus?**
- **Welche Programmschnittstellen existieren?**
- **Für welche Komponenten wird Unicode support benötigt?**

GSE BUNID Guide Tagungen

- **Letzte Tagung vom 8. – 10. Oktober bei der TK!**
- **Auswirkungen von Unicode und Codepage-Einstellungen auf Cobol und DB2 - IBM GTS/SWG & TK**
- **z/OS Components and Unicode support - IBM z/OS Dev.**
- **nPA – Auswirkungen auf die IT - Bundesdruckerei**
- **Erfahrungen bei der Umstellung eines Teilprojektes auf Unicode - DBSystem**
- **Konvertierung und Komprimierung von Unicode Daten - Limes Datentechnik**
- **Angewandte z/OS Unicode Services - IBM SO**
- **Unicode – Auswirkung auf Sortierung - IBM BA Student**
- **Global View on Unicode support, Projects and Requirements - IBM SVL**
- **AFP - Unicode Unterstützung - Ricoh**
- **Application Development – zTools support for Unicode - IBM SWG**
- **Verwendung von diakritischen Zeichen mit DataStudio - IBM SWG**



- **Backup / Details**

Scenario 1: Unicode im Programm & EBCDIC in der DB

- Cobol Program Umwandlung in US (CCSID 1140 entspricht 037 plus „€“)

DB2 Installation in Deutschland

Anwendung soll auch für französische Anwender nutzbar sein

Schrittweise Einführung von Unicode im Programm

COBOL Program:

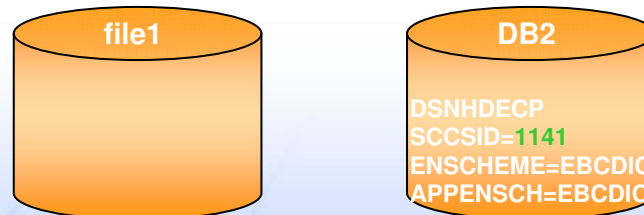
-> *HVin* *PIC X* → DB2 Hostvariable
 -> *HVout, HVproc* *PIC N* → DB2 Hostvariable (CCSID 1200)
 -> *Filevar, Outvar* *PIC X* → Program variable

-> **DECLARE VARIABLE** :HVproc, :HVout CCSID 1200
 -> **SELECT** Col3, Col4 INTO :HVout, :HVproc FROM TB1
 WHERE Col1 = :HVin
 AND Col2 = ,literalÜ' → SQL Parser 1140 -> 1208

-> **GET** Filevar FROM file1 → No conversion
 -> HVproc = ,WertÜ' → Conversion 1200 <-> 1140
 -> HVproc = Filevar → Conversion 1200 <-> 1140

-> **MOVE FUNCTION DISPLAY-OF** (HVout, 1147) TO Outvar
 -> Display (Outvar) → korrekte Darstellung / PComm CCSID 1147

1. COBOL Compiler with CODEPAGE 1140
→ CCSID für Source statements und Literale
2. DB2 Coprocessor NOSQLCCSID
DB2 Precompiler CCSID (1140)
3. BIND with ENCODING(1147)
→ CCSID für Daten in DB2 Hostvariable
4. DB2 Runtime
Col1 = :HVin → data conv 1141 <-> 1147
Col2 = ,literalÜ' → data conv 1141 <-> 1208
Col3/4 = :HVout/proc → data conv 1141 <-> 1200



-> DB2 Tabellen
 → EBCDIC encoding für tablespace
 → Daten in CCSID 1141 gespeichert

Fazit:

- **MOVE latin_data TO national** geht, aber nicht in umgekehrter Richtung
 → COBOL Intrinsic function erforderlich
- **DECLARE VARIABLE** für UTF-16 Definition erforderlich, da National Variable Definition in COBOL und SQL Package EBCDIC encoding definiert ist (ansonsten sqlcode -189).

Scenario 2: Unicode in der DB & EBCDIC im Programm

- Cobol Program Umwandlung in US (CCSID 1140 entspricht 037 plus „€“)
- Schrittweise Einführung von Unicode in der Datenbank
- Anwendung soll auch für französische Anwender nutzbar sein

COBOL Program:

-> HVin, HVout, HVproc PIC X → DB2 Hostvariable
-> Filevar, Outvar PIC X → Program variable

...
-> DECLARE VARIABLE :HVproc CCSID 1140
-> SELECT Col3, Col4 INTO :HVout, :HVproc FROM TB1
WHERE Col1 = :HVin
AND Col2 = ,literalÜ' → SQL Parser 1140 -> 1208

...
-> GET Filevar FROM file1 → No conversion

...
-> HVproc = ,WertÜ' → No conversion required
-> HVproc = Filevar → No conversion required

...
-> MOVE HVout TO Outvar → No conversion
-> Display (Outvar) → korrekte Darstellung / PComm CCSID 1147

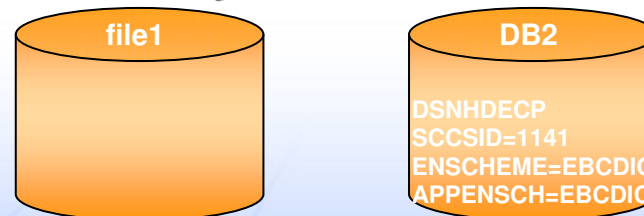
1. COBOL Compiler with CODEPAGE 1140
→ CCSID für Source statements und Literale

2. DB2 Coprocessor NOSQLCCSID
DB2 Precompiler CCSID (1140)

3. BIND with ENCODING(1147)
→ CCSID für Daten in DB2 Hostvariable

4. DB2 Runtime
Col1/3 = :HVin/out → data conv 1208 <-> 1147
Col2 = ,literalÜ' → data conv 1208 <-> 1208
Col4 = :HVproc → data conv 1208 <-> 1140

Change



-> DB2 Tabellen
→ Unicode encoding für tablespace
→ Daten in CCSID 1208 gespeichert

Change

- **Fazit:**
→ DB2 unterstützt UTF-8 und UTF-16 abhängig von verwendeten datatypes in der DB2 Tabelle
→ SQL Package pro Land; abgestimmt mit PComm Einstellung

Scenario 3: Unicode everywhere

- **Cobol Program Umwandlung & DB2 Installation in Deutschland**
- Durchgängige Unicode Implementierung zur Unterstützung eines erweiterten Zeichensatzes – diakritische Zeichen

Change COBOL Program:

-> Outvar PIC N → Program variable (CCSID 1200)
 -> HVin, HVout, HVproc PIC N → DB2 Hostvariable (CCSID 1200)
 -> Filevar PIC X → Program variable

...
 -> SELECT Col3, Col4 INTO :HVout, :HVproc FROM TB1
 WHERE Col1 = :HVin
 AND Col2 = ,literalÜ' → SQL Parser 1141 -> 1208

...
 -> GET Filevar FROM file1 → No conversion
 -> HVproc = ,WertÜ' → Conversion 1200 <-> 1141
 -> HVproc = Filevar → Conversion 1200 <-> 1141

Change MOVE HVout TO Outvar

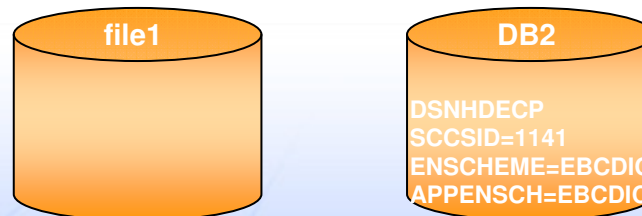
-> Display (Outvar) → Browser / GUI mit Unicode support

1. COBOL Compiler with CODEPAGE 1141
 → CCSID für Source statements und Literale

2. DB2 Coprocessor NOSQLCCSID
 DB2 Precompiler CCSID (1141)

3. BIND with ENCODING(1200)
 → CCSID für Daten in DB2 Hostvariable

4. DB2 Runtime
 Col1 = :HVin → data conv 1208 <-> 1200
 Col2 = ,literalÜ' → data conv 1208 <-> 1208
 Col4 = :HVout/proc → data conv 1208 <-> 1200



-> DB2 Tabellen
 → Unicode encoding für tablespace
 → Daten in CCSID 1208 gespeichert

▪ Fazit:

- DB2 und SQL Package encoding in Unicode
- COBOL National Variablen Deklarationen
- PComm/3270 Interface wird ersetzt durch Browser/GUI front-end

ICAO 9303 / ISO 7501

- <http://www.hasbrouck.org/documents/ICAO9303-pt2.pdf>

APPENDIX 3 to Section III

TRANSLITERATIONS RECOMMENDED FOR USE BY STATES

<i>Sequence number</i>	<i>National character</i>	<i>Description</i>	<i>Recommended transliteration</i>
A. Transliteration of multinational characters			
1	Á	A acute	A
2	À	A grave	A
3	Â	A circumflex	A
4	Ä	A diaeresis	AE
5	Ã	A tilde	A
6	Ă	A breve	A
7	Å	A ring	AA

Normalization

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

Source		NFD		NFC
š 1E69	:	s ◌ ◌ 0073 0323 0307		š 1E69
đ 1E0B 0323	:	d ◌ ◌ 0064 0323 0307		đ ◌ 1E0D 0307
q̇ 0071 0307 0323	:	q ◌ ◌ 0071 0323 0307		q ◌ ◌ 0071 0323 0307

Source		NFD		NFC		NFKD		NFKC
fi FB01	:	fi FB01		fi FB01		f i 0066 0069		f i 0066 0069
2 ⁵ 0C32 2075	:	2 ⁵ 0032 2075		2 ⁵ 0032 2075		2 5 0032 003E		2 5 0032 0035
ƒ̇ 1E9B 0323	:	f ◌ ◌ 017F 0323 0307		ƒ ◌ 1E9B 0323		s ◌ ◌ 0073 0323 0307		š 1E69