



7437 - EDI und E-Business Standards

Praktikumsaufgabe 05 & 06
Bestellungen und Lieferavis mit
UN/EDIFACT und EANCOM,
Mapping per EDI-Bibliothek



Das Szenario



- Szenario
 - Die Praktikumssteilnehmer setzen ihre Handelspartner-Rolle (Lieferanten, Händlern) der Konsumgüterbranche aus Praktikum 01/02 fort.
 - Sie korrigieren ggf. die in Übung 04 entworfenen Beispieldaten für Bestellungen und Lieferavis und entwickeln ein Programm zur automatischen Erzeugung dieser Daten.
- Ziele der Übung
 - Praxis im Umgang mit einer Klassenbibliothek für EDI-Daten
 - Beherrschung der EDIFACT Service-Segmente
 - Mappingtechnik-Grundlagen



- Entwickeln mit Ruby
 - Erstellen Sie Dateien "*.rb"
 - Editoren: emacs (besitzt Ruby-Modus), SciTE (scite), ...
 - Zum Testen: irb (Interactive Ruby), eine sehr praktische Ruby-Shell
 - Ausführung auf der Kommandozeile, etwa:

```
$ ruby sample.rb
# Erste Zeile = #! /usr/bin/env ruby, dann: chmod +x sample.rb
$ ./sample.rb          # nun direkt ausführbar.
```
 - EDI-Klassenbibliothek einbinden
 - Diese Daten sind nicht regulär installiert, daher explizite Pfadangabe:

```
$ export RUBYLIB=/local0/werntges/kurse/edi/r-edi
```



- Dateikopf (Einbindung der Bibliothek):

```
#!/usr/bin/env ruby
require "r-edi/base"
```
- Erzeugung eines EDIFACT Interchange:

```
ic = EDI::Interchange_E.new({
  'charset' => 'UNOA',
  'version' => '2',
  'intref' => '123456',
  'output_mode' => 'formatted' # oder ''
})

cde = ic.header.cs002          # S002 aus UNB
cde.d0004 = "1234567890123"  # GLN zuweisen
cde.d0007 = "14"             # Qualifier setzen
# usw. für alle relevanten Header-Daten
```



- Einfügen einer Nachricht:

```
params = {
  'msg_type' => 'ORDERS',
  'version' => 'D',
  'release' => '96A',
  'resp_agency' => 'UN',
  'assigned_code' => 'EAN008'
}
msg = ic.new_message( params )
# Befüllen mit Segmenten, ..., schließlich:
ic.add msg      # Nachricht an Interchange anfügen
```



- Segmente erzeugen und befüllen:

```
seg = msg.new_segment( "BGM" )
seg.cC002.d1001 = '220'
seg.d1004 = '123456abc'      # Belegnummer
seg.d1225 = '9'             # Qualifier setzen
# usw., schließlich:
msg.add seg      # Segment an Nachricht anfügen
```

- Kopfdaten beeinflussen:

```
msg.header.d0062 = "543210" # UNH, DE 0062 setzen

# msg.trailer existiert zwar, sollte aber automatisch
# korrekt befüllt werden - nicht beschreiben!
```



- Erzeugte Daten ausgeben:

```
ic.validate    # Gegen Directories prüfen lassen
ic.write( stdout )

# Schreibt den Interchange nach STDOUT
# Wenn output_mode == 'formatted':
#     Eine Zeile pro Segment
# Sonst: Als character stream, wie im Standard.

# In Datei schreiben:
# Datei vor Aufruf öffnen, handle übergeben:
ic.output_mode = ''
File.new("my_output.edi", "w") {|hnd| ic.write hnd }
```



- Hinweise zum Ablauf
 - Erzeugen Sie zunächst das Interchange
 - Sender- und Empfänger-GLN können mit denen aus den NAD-Segmenten übereinstimmen.
 - Aktuelles Datum befüllen
 - Interchange Ref. mittels Nummerngenerator aus Übung 02 befüllen.
 - Legen Sie dann eine EDIFACT-Nachricht an
 - Header-Angaben fest vorgeben, eingebauten Nachrichtenzähler gewähren lassen (d.h. Sie greifen möglichst nicht in UNH und UNT ein).
 - Segmentweise befüllen, dabei auf korrekte Reihenfolge achten.
 - Ggf. Segmente in einem Array zwischenlagern.



- Ersatz-Aufgabe
 - Wer die Aufgaben bisher nicht (hinreichend) bearbeitet hat, kann dennoch hier "einsteigen"!
 - **Mappen Sie ggf. (statt Ihrer eigenen Daten) Beispiel-Bestelldaten im Format von Metro MGI's "MEC" WebEDI-Lösung nach EANCOM '02**
- Einzelheiten
 - Formatbeschreibung Quellformat:
 - kurse/edi/ss2003/proj/MEC-Schnittstelle-ORDERS.pdf
 - Beispiel-Quelldaten:
 - kurse/edi/ss2003/proj/bestell-khof1.mec
 - Daten bitte an Ihr Szenario anpassen (GLNs, GTINs!)



- Die Aufgabe
 - Mappen Sie Ihre Belege aus Übung 03 (Händler: eine Bestellung, Lieferanten: ein Lieferavis)
 - von Ihrer In-house Struktur aus Übung 03
 - nach EANCOM '02 gemäß des (korrigierten) Nachrichtenaufbaus aus Übung 04
 - Programm-Name:
`mapper05.rb`
 - Gewünschter Aufruf:
`mapper05.rb srcdata.msg > eancomdata.edi`



- Abzugeben
 - mapper05.rb # Ihr Programm-Code
 - srcdata.msg # Ihre verwendeten Quelldaten
 - eancomdata.edi # Ihre Ergebnisse(srcdata.msg und eancomdata.edi sollten von Ihrem Programm verwendet bzw. erzeugt worden sein. Sie können z.B. orders03.msg und orders04.edi entsprechen.)
- Abgabeordner
 - Wie üblich, unter `~wertges/kurse/edi/abgaben/a/<matrnr>`
- Annahmeschluss
 - 5 Minuten nach Beginn der **ü**bernächsten Übung (25. Mai).