

---

# ATEDI - Verarbeitung von UN/EDIFACT Messages mit GNU Software

Franz Zahaurek

März 2005

## Inhaltsverzeichnis

ATEDI (EDI in AT) .....	2
Funktionalität .....	2
Struktur einer EDI-Message .....	2
Abschlusszeichen (terminator) vs. Trennzeichen (separator) .....	3
Der FLEX-Scanner .....	3
Der BISON-Parser .....	3
Basisfunktionen .....	4
Individuelle Erweiterungen .....	4
Codierung und Images: EBCDIC - ASCII .....	4
Copyright .....	5

Voraussetzung für den Austausch elektronischer Nachrichten (electronic data interchange - EDI) ist die Normierung der Nachrichtenformate. Sie erfolgt seit jeher unter der Schirmherrschaft der UNO durch die Organisation: UN/EDIFACT [<http://www.unece.org/trade/untdid>]. Für den Österreichischen Zahlungsverkehr wurde durch die Studiengesellschaft für Zusammenarbeit im Zahlungsverkehr GmbH - STUZZA [<http://www.stuzza.at>] eine Anpassung der internationalen Nachrichtenformate einerseits in Form von Vereinfachungen, andererseits aber auch Erweiterungen vorgenommen. Ganz im Gegensatz zur Offenheit der Spezifikationen, erfolgt die Verarbeitung der EDI-Nachrichten bei den Finanzinstituten sowie Handels- und Produktionsfirmen meist auf grossen Hosts, die unter proprietärer Software betrieben werden.

## ATEDI (EDI in AT)

Innerhalb des Projekts ATEDI soll für die öffentlichen Spezifikationen ebenfalls freie GNU-Software [<http://www.gnu.org>] zur Verarbeitung der EDI-Datenbestände bereitgestellt werden.

## Funktionalität

Die hier verfügbaren Programme heissen wie die EDI-Messages, die sie verarbeiten. (Das Programm atibxs verarbeitet also z.B. die EDI-Messages vom Typ ATIBxS) Die Programme lesen jeweils eine EDI-Message ein und stellen fest, ob sie strukturell korrekt ist. Dabei wird die gesamte Nachricht in ihre einzelne Segmente zerlegt und programmintern verfügbar gemacht. In EDI-Messages vom Typ ATIBxS enthaltene Images, eine besondere Erweiterung der STUZZA-Spezifikation, werden in separate Files ausgefiltert und/oder aus der EDI-Message entfernt. Über Optionen kann man das Ausgabeformat der EDI-Message steuern. Folgende Formate stehen zur Auswahl:

- einfacher Text, Segmente je Strukturierung eingerückt
- HTML-Format, mit Links auf die Images
- XML-Format

Abschliessend erfolgt eine Prüfung des im UNT-Segment enthaltenen Segmentzählers. Ein mögliches UNA-Segment zu Beginn einer Nachricht wird nicht beachtet. Wir gehen von folgender Grundeinstellung für die Trenn- und Sonderzeichen aus: UNA : + . ? ' '

- Subfeldtrenner: : (Doppelpunkt)
- Feldtrenner: + (Plus)
- Dezimalpunkt: . (Punkt)
- Escapezeichen: ? (Fragezeichen)
- Reservezeichen: Blank (Leerzeichen)
- Segmentabschlusszeichen: ' (einfaches Hochkomma)

## Struktur einer EDI-Message

Grundlage des Abgleiches einer konkreten EDI-Message mit ihrer Struktur ist die Formulierung der Strukturbeschreibung als sogenannte Grammatik. Der Kern dieser formalen Beschreibung kann dabei direkt aus den sogenannten Branch-Diagrammen bzw. den Übersichten der Segmentbeschreibungen aus den Spezifikationen abgeleitet werden.

# Abschlusszeichen (terminator) vs. Trennzeichen (separator)

Die EDI-Spezifikationen beschreiben eine EDI-Message als Folge von Segmenten. Jedes Segment besteht dabei aus:

- der 3-stelligen Segment-Kennung (tag)
- dem folgenden Feld-Trenner (+)
- danach einem oder mehreren Feldern, mit möglichen Subfeldern mit Subfeld-Trennern (:)
- dem abschliessenden Segmentendezeichen (')

Aus heutiger Sicht erweist sich das Konzept der Trennzeichen bei Feldern und Subfeldern als nachteilig. Das aktuell vorherrschende Konzept der Textauszeichnung (Markup) das bei XML-Dateien angewendet wird, klammert daher auch jedes Datenelement mit einem öffnenden und schliessenden Paar zweier Tags. Wegen des Fehlens niederwertiger Abschlusszeichen (':' nach dem letzten Subfeld, '+' nach dem letzten Feld) kann der Typ des ersten Datenelementes eines Feldes nur kontextbezogen als Feld oder Subfeld ermittelt werden. In ATEDI wird daher konzeptuell jedes Feld als Folge einzelner Subfelder betrachtet.

## Der FLEX-Scanner

Als Scanner für die EDI-Messages wird das GNU-Programm FLEX verwendet. Der Scanner qualifiziert jedes Datenelement der EDI-Message (Segment-Tags, Felder und Trenner) protokolliert diese Informationen und stellt sie dem Parser zur Verfügung. Alle Dateninhalte (Texte, Zahlen usw.) werden in einer Wertetabelle gesammelt und können vom Parser mit den Indexwerten Feldnummer und Sufeldfeldnummer referenziert werden.

## Der BISON-Parser

Als Parser für die EDI-Messages wird das GNU-Programm BISON verwendet. Er übernimmt die Informationen des Scanners und prüft die Übereinstimmung der Datenelemente (Token) mit der formalen Beschreibung. Segmente, die nicht der geforderten Struktur entsprechen werden protokolliert. Am Ende der Verarbeitung jedes Segmentes stehen alle seine Dateninhalte in der vom Scanner aufgebauten Wertetabelle bereit.

## Basisfunktionen

Folgende Basisfunktionen stehen zur Verfügung:

- Prüfung der Übereinstimmung mit der Gesamtstruktur
- bei Differenzen werden das unerwartete Element und die Liste möglicher Elemente ausgegeben
- Prüfung der maximalen Anzahl einzelner Segmente
- Prüfung der Segmentanzahl
- Typisierung jedes einzelnen Segmentes samt Inhalt
- Ausfiltern binärer Images in separate Files (ermöglicht Sichtprüfung der verbleibenden EDI-Message mit gängigen Editoren)
- Ausgabe der EDI-Message als Text-Datei
- Ausgabe der EDI-Message als HTML-Datei
- Ausgabe der EDI-Message als XML-Datei

## Individuelle Erweiterungen

Da alle Feldinhalte eines Segments gezielt angesprochen werden können, besteht eine gute Ausgangsbasis für individuelle Erweiterungen.

## Codierung und Images: EBCDIC - ASCII

Soll eine in EBCDIC kodierte Message auf einem LINUX- oder UNIX-System verarbeitet werden, muss sie zuerst nach ASCII konvertiert werden. Enthaltene Images werden dadurch unbrauchbar. Um sie betrachten zu können, nachdem sie durch das ATEDI-Tool atibxs aus der EDI-Message extrahiert wurden, ist daher ein Rückübersetzung nach EBCDIC nötig. Die Codeumsetzung erfolgt praktischerweise mit dem Programm `recode`. Der Pseudocode des Ablaufes:

```
recode ebcdic..ascii _atibxs_message_ (das ist die EBCDIC kodierte Message)
mkdir images
cat _atibxs_message_ | ./atibxs -H -f './images/atibxs_%d.tif' >_atibxs_.html
recode ascii..ebcdic images/atibxs_*.tif
```

`_atibxs_.html` kann mit einem Browser betrachtet werden und enthält Links, die auf die extrahierten Images weisen.

# Copyright

Die Programme des ATEDI Projekts stehen unter dem Copyright der GNU General Public Licence - GPL [<http://www.gnu.org/licenses/gpl.txt>].

## **Kontakt.**

- **Autor:** Franz Zahaurek
- **Website:** <http://www.fzk.at/atedi>.
- **eMail:** [atedi@fzk.at](mailto:atedi@fzk.at) .