



Schriften

451

4/99

ÖPNV-Datenmodell 5.0

"Schnittstellen-Initiative"

**Dateiformat für die Datenübertragung zwischen
ÖPNV - Anwendungen**

Version 1.0

**Gesamtbearbeitung:
Ausschuß für Informationsverarbeitung**

Inhaltsverzeichnis

1 VORWORT ÖPNV-DATENMODELL 5.0:	4
1.1 Das ÖPNV-Datenmodell als Orientierungsrahmen für die Datenmodellierung im Personenverkehr	4
1.2 Die VDV-Schnittstelleninitiative ÖPNV-Datenmodell	4
2 EINLEITUNG	6
2.1 Zweck	6
2.2 Abgrenzung	6
2.3 Textkonventionen	7
3 BEZEICHNUNG DER SCHNITTSTELLEN-DATEIEN	8
3.1 Dateinamen mit Tabellennummern	8
3.2 Dateiname entspricht Tabellennamen	9
4 AUFBAU DER SCHNITTSTELLEN-DATEIEN	10
4.1 Allgemeine Struktur der Dateien	10
4.2 Aufbau der Dateien	10
4.3 Befehlsübersicht	11
5 SEMANTIK DER EINZELNEN BEFEHLE	12
5.1 Headerbefehle	12
5.1.1 Datum-, Zeitformat und Darstellungsmodus	13
5.1.2 Erstellerinfo	13
5.1.3 Zeichensatz der Datensätze	14
5.1.4 Version	14
5.1.5 Interfaceversion	14
5.1.6 Datenversion	15
5.1.7 Dateiformat	15
5.2 Bodybefehle	15
5.2.1 Tabelle	15
5.2.2 Attributbezeichnungen	16
5.2.3 Attributformate	16
5.2.4 Datensatz	16
5.2.5 Tabellenende	17
5.2.6 Kommentar	17
5.3 Trailerbefehle	18
5.3.1 Dateiende	18
6 BEISPIELE	19

7 ZEICHENSÄTZE	21
7.1 ASCII Zeichensatz	21
7.2 ISO8859-1 Zeichensatz	22

1 Vorwort ÖPNV-Datenmodell 5.0:

1.1 Das ÖPNV-Datenmodell als Orientierungsrahmen für die Datenmodellierung im Personenverkehr

Acht Jahre nach der ersten Veröffentlichung der VDV-Schrift „ÖPNV-Datenmodell“ ist dieses, jenseits der deutschen Grenzen häufig unter dem Namen „VDV-Datenmodell“, zu einer nicht mehr wegzudenkenden Grundlage für die Datenmodellierung im ÖPNV geworden. Eine Dekade nach den ersten Überlegungen, allen Standardanwendungen im Verkehrsunternehmen ein allgemeingültiges gleiches Datenmodell zugrunde zu legen, haben sich tatsächlich viele Systemanbieter am ÖPNV-Datenmodell orientiert und ihr eigenes Produktdatenmodell auf seiner Basis entwickelt. Einer ganzen Reihe von ihnen wurde vom VDV für ihre Software das Prädikat „kompatibel mit dem ÖPNV-Datenmodell“ verliehen. In vielen Ausschreibungen der Verkehrsunternehmen ist die „Kompatibilität mit dem ÖPNV-Datenmodell“ eine selbstverständliche Forderung.

Vielleicht gerade wegen des großen Erfolgs des ÖPNV-Datenmodells wurden jedoch zunehmend Forderungen an den VDV herangetragen, über die Leistung des ÖPNV-Datenmodells hinaus, die als Basis im Sinne einer allgemein verständlichen Sprachregelung zu verstehen ist, auch praxisorientierte Lösungen zu forcieren. Die Vorstellungen gehen hin zu Standardschnittstellen, die im Sinne einer Steckerkompatibilität sofort einsetzbar sind und Standard-Softwaremodule ohne größeren Aufwand miteinander kommunizieren lassen.

1.2 Die VDV-Schnittstelleninitiative ÖPNV-Datenmodell

Die Forderung nach möglichst sofort einsetzbaren Schnittstellen im ÖPNV kommt sowohl aus den Verkehrsunternehmen als auch von Dritten. Gerade auf Gebieten wie der dynamischen Fahrgastinformation ist eine stetig steigende Nachfrage nach Informationen von Organisationen außerhalb der Verkehrsunternehmen festzustellen. Aber auch durch Kooperationen und die Bildung von Verkehrsverbänden steigt der Bedarf nach Kommunikation über die Unternehmensgrenzen hinweg. Aus diesem Grunde hat sich der VDV entschlossen, eine Initiative ins Leben zu rufen, die unter dem Titel „Schnittstelleninitiative ÖPNV“ die Realisierung von standardisierten Datenschnittstellen auf der Basis des ÖPNV-Datenmodells fördern soll.

Diese Schnittstellen stellen grundsätzlich einen Ausschnitt aus dem ÖPNV-Datenmodell dar. Es handelt sich also nicht um eine neue Festlegung, sondern vielmehr um eine konsequente Anwendung des auf der Basis des langjährigen Forschungsprojektes veröffentlichten ÖPNV-Datenmodells. Eine exaktere Beschreibung und die Ergänzung um technische Spezifikationen für die Datenübertragung sowie funktionelle Aspekte führt jedoch zu einer größeren Praxistauglichkeit, als dies mit dem reinen ÖPNV-Datenmodell möglich war.

Die VDV-Schrift 452 enthält die erste Schnittstellen-Definition aus dieser Initiative. Sie betrifft den Bereich „Liniennetz und Fahrplan“. Die Definition zeichnet sich im Vergleich zum ÖPNV-Datenmodell durch folgende Eigenschaften aus:

- Neben dem bereits in früheren Versionen des ÖPNV-Datenmodells geforderten SQL-Zugriff wird alternativ ein Dateiformat für den Offline-Datentransfer definiert.
- Der Mindestumfang des Datenmodells wird eindeutig festgelegt.
- Für die einzelnen Attribute der Nutzersicht werden Wertebereiche restriktiver definiert.
- Die Beschreibung der einzelnen Attribute wurde inhaltsreicher und damit präziser erfaßt.

2 Einleitung

Die vorliegende Beschreibung dokumentiert das Format der ÖPNV-Schnittstellendateien, das für den Datenaustausch eingesetzt wird.

Der Datenaustausch mit diesen Dateien ist bei folgenden Gegebenheiten notwendig:

- Daten werden von einem Fremdsystem importiert oder exportiert, evtl. werden die Daten auf einer anderen Hardware-Plattform weiterverwendet.
- Daten müssen mit marktüblicher Standard-Software nachbearbeitet, inspiziert oder ausgewertet werden:
 - z.B. soll mit einem Texteditor ein Datenbestand inspiziert oder modifiziert werden
 - z.B. soll mit einem Tabellenkalkulationsprogramm ein Datenbestand (evtl. mit Hilfe zusätzlicher Makros) ein- oder ausgelesen werden

2.1 Zweck

Um eine einheitliche Art der Datenübertragung zu erreichen, wird in diesem Dokument die Struktur und der Aufbau der Schnittstellen-Dateien beschrieben. Es soll eine einheitliche, klare und einfache Schnittstelle gebildet werden, welche ermöglicht, Probleme im Austausch von Daten zu lokalisieren und zu beheben. Der Aufbau der Schnittstellen-Dateien wurde so gewählt, daß eine Überprüfung der Daten möglich ist.

Die Dateien sollen von folgenden Produkten importiert und exportiert werden können:

- Datenbank (Ingres®, Oracle®, Sybase®) mit Hilfe eines kleinen Einleseprogrammes
- Texteditoren (Unix®, Windows®, DOS)
- MS-Excel® mit Makro
- MS-Access® mit Makro

Der Aufbau der Dateien ist primär für den Austausch von Tabellen aus relationalen Datenbanken geeignet und vorgesehen.

Das Dokument richtet sich an Hersteller von Planungsprogrammen im Bereich des öffentlichen Personennahverkehrs, die Daten mit anderen Herstellern über das ÖPNV-Datenmodell austauschen.

2.2 Abgrenzung

Das Dokument enthält keine Datenbankdefinitionen (wie Datentypen, Wertebereiche, Indizes, Vorbelegung für leere Felder etc.).

2.3 Textkonventionen

In dieser Beschreibung werden optische Standardtextformate verwendet:

<i>kursiv</i>	Im Text: Hervorhebung von Namen Beispiel: Das Konversionstool <i>unix2dos</i> .
<i>kursiv</i>	In Aufzählungen und Listen: Platzhalter für Daten Beispiel: chs; <i>Zeichensatz</i> Hier steht das Wort Zeichensatz für eine Zeichenkette, die einen wirklichen Zeichensatz bezeichnet, z.B. 'ASCII'.
<code>courier</code>	Befehle, die genau so eingegeben werden müssen, wie sie im Text erscheinen. Beispiel: <code>lp /pub/iso</code>
<code>courier</code>	Inhalt (oder Teile davon) einer Datei
fett	Informationen, die besonders wichtig sind und die leicht übersehen werden könnten.

3 Bezeichnung der Schnittstellen-Dateien

3.1 Dateinamen mit Tabellennummern

Die Bezeichnung von Dateinamen ist durch die 8+3 Zeichenanzahl von DOS-Dateinamen begrenzt. Lange Dateinamen, wie von Windows NT und UNIX verwendet, werden nicht unterstützt. In den Dateinamen dürfen nur **Kleinbuchstaben** verwendet werden.

Jeder Tabelle wird eine Nummer zugewiesen, die im Dateinamen verwendet wird:

Numerierung der Tabellen		
Kapitel	Tabelle	Nr.
Kalenderdaten	MENGE_BASIS_VERSIONEN	485
	BASIS_VER_GUELTIGKEIT	993
	FIRMENKALENDER	348
	MENGE_TAGESART	290
Ortsdaten	MENGE_ONR_TYP	998
	MENGE_ORT_TYP	997
	REC_HP	229
	REC_OM	295
	REC_ORT	253
Betriebsdaten	FAHRZEUG	443
	ZUL_VERKEHRSBETRIEB	992
	MENGE_BEREICH	333
	MENGE_FZG_TYP	293
	REC_ANR	996
	REC_ZNR	994
Netzdaten	REC_SEL	299
	REC_SEL_ZP	995
	MENGE_FGR	222
	ORT_HZTF	999
	SEL_FZT_FELD	282
	REC_UEB	225
	UEB_FZT	247
	MENGE_FAHRTART	332
Liniendaten	LID_VERLAUF	246
	REC_LID	226
Fahrplandaten	REC_FRT	715
	REC_FRT_HZT	308
	REC_UMLAUF	310

Die Dateien werden mit folgendem Namen versehen:

- Prefix (char[8]) 'i', (als Abkürzung für Interface-Datei)
Nummer der Tabelle, 3-stellig,
Tag des Jahres, 3-stellig,
(ist 001, 31.12 ist 365 bzw. 366 für Schaltjahre) '0' (Null)
- Suffix (char[4]) '.x10' (für: ASCII eXchange format version **1.0**)

Beispiel zur Erzeugung eines Dateinamens für die Tabelle 'MENGE_BASIS_VERSIONEN', Nr. 485, am 24.12.1995:
i4853580.x10

3.2 Dateiname entspricht Tabellennamen

Alternativ können als Dateinamen auch der Tabellenname der entsprechenden Tabelle (ausschließlich in Kleinbuchstaben) verwendet werden. Dieser Dateiname trägt als Extension ebenfalls die Ergänzung .x10. Beispiel: Der Dateiname für die Tabelle 'REC_FRT' heißt 'rec_frt.x10'.

4 Aufbau der Schnittstellen-Dateien

4.1 Allgemeine Struktur der Dateien

- Ein Datensatz besteht aus mehreren Datenfeldern, die durch Feldtrennzeichen voneinander getrennt sind. Nach dem letzten Datenfeld eines Datensatzes folgt kein Feldtrennzeichen, sondern ein Zeilenumbruch.
- Als Feldtrennzeichen wird der Strichpunkt (;) verwendet.
- Das erste Datenfeld pro Datensatz enthält ein Befehlskürzel (drei Kleinbuchstaben), das bestimmt, wie der Inhalt des Datensatzes zu interpretieren ist.
- Jeder Datensatz belegt genau eine Zeile.
- Ein Datenfeld kann ein Befehlskürzel, eine Formatdefinition, einen Namen, eine Zahl oder eine Zeichenkette (Textdaten) enthalten.
- Zeichenketten werden von Textbegrenzungszeichen eingerahmt.
- Als Textbegrenzungszeichen wird das Doublequote-Zeichen (") verwendet.
- Falls eine Zeichenkette ein Textbegrenzungszeichen enthalten soll, so ist dieses zu verdoppeln.
- Eine Zeichenkette darf auch Feldtrennzeichen enthalten. Ein solches wird innerhalb einer Zeichenkette als normales Zeichen interpretiert.
- Die Darstellung der Daten ist beim Export wählbar (Freiformat oder Kolonnenform).
- Die Darstellung in Kolonnenform ist für die visuelle Überprüfung des Dateiinhaltes vorgesehen.
- Beim Import spielt die Darstellungsart keine Rolle.
- Der für Tabellendatensätze zu verwendende Zeichensatz ist wählbar. Für alle anderen Angaben (Dateiheader, Tabellenbeschreibungen, Kommentare) ist der ASCII-Zeichensatz zu verwenden.

4.2 Aufbau der Dateien

Die Schnittstellen-Datei besteht grob aus folgenden Teilen:

- Header
- Body
 - Tabellenheader
 - Tabellenbody (bestehend aus beliebig vielen Datensätzen)
 - Tabellentrailer
- Trailer

4.3 Befehlsübersicht

Kürzel	Befehl	Bedeutung
mod	Mode	Datums- und Zeitformat, Darstellungsart
src	Source	Erstellendes Programm, Erstelldatum und -zeit
chs	Character Set	Zeichensatz der Daten in den Tabellen
ver	Version	Version des erstellenden Programmes
ifv	Interfaceversion	Version des Interfaces des schreibenden Programmes
dve	Datenversion	Version der Daten
fft	Fileformat	Typ der Daten nach dem Header

tbl	Begin of Table	Name der Tabelle
atr	Attribute	Namen der Attribute
frm	Format	Format der Attribute
rec	Record	Datensatz
end	End of Table	Ende der Tabelle

eof	End of File	Ende der Datei
-----	-------------	----------------

5 Semantik der einzelnen Befehle

In diesem Kapitel wird hauptsächlich die Semantik der einzelnen Befehle anhand von Beispielen erklärt.

5.1 Headerbefehle

Folgende Befehle müssen am Anfang der Datei stehen und geben die Darstellungsart wieder, damit das empfangende Programm die nachfolgenden Daten korrekt interpretieren kann.

1. Datums- und Zeitformat, Darstellungsmodus (Befehl *mod*, engl. *Mode*)
2. Interfaceversion des schreibenden Programmes (Befehl *ifv*)
3. Ersteller und Erstelldatum der Dateien (Befehl *src*, engl. *Source*)
4. Zeichensatz (Befehl *chs*, engl. *Character Set*)
5. Version (Befehl *ver*, engl. *Version*)

5.1.1 Datum-, Zeitformat und Darstellungsmodus

Mit dem Befehl Mode (*mod*) wird beschrieben, wie Datums- und Zeitangaben in der Datei dargestellt werden. Zusätzlich wird der Darstellungsmodus der Datensätze festgelegt.

<i>mod</i> ;	Datumsformat; Zeitformat; Mode
<i>Datumsformat</i>	Für Datumsfelder im Tabellenteil verwendetes Datumsformat
DD.MM.YYYY	
Datumstrennzeichen:	Punkt (.)
<i>Zeitformat</i>	Für Zeitfelder im Tabellenteil verwendetes Zeitformat
HH:MM:SS	
Zeittrennzeichen:	Doppelpunkt (:)
HH:	24-Stunden-Format
<i>Mode</i>	Darstellungsmodus der Datensätze
Normal: free	
Mögliche Darstellungsmodi:	
free	Frei formatierte Darstellung der Datensätze. Die Formatierung der Daten mit Leerschlägen ist nicht vorgegeben.
aligned	Datensätze werden mit Hilfe der Formatangaben mit Leerschlägen übersichtlich in Kolonnen dargestellt.

Beispiel zur Modusangabe:

```
mod; DD.MM.YYYY; HH:MM:SS; free
```

5.1.2 Erstellerinfo

Der Befehl Source (*src*) enthält die Quellenangabe der Schnittstellen-Datei.

<i>src</i> ;	Ersteller; Erstellungsdatum; Erstellungszeit
<i>Ersteller</i>	Beliebiger Text im ASCII-Zeichensatz, welcher den Ersteller (Programm, Modul) der Datei bezeichnet. Muß mit Textbegrenzungszeichen dargestellt werden.
<i>Erstellungsdatum</i>	Erstellungsdatum der Datei. Muß im Datumsformat gespeichert und mit Textbegrenzungszeichen dargestellt werden.
<i>Erstellungszeit</i>	Erstellungszeit der Datei. Muß im Zeitformat gespeichert und mit Textbegrenzungszeichen dargestellt werden.

Beispiel zur Erstellerinfo:

```
src; "LIO-Adapter VDV 4.0"; "01.01.1998"; "00:00:00"
```

5.1.3 Zeichensatz der Datensätze

Der Befehl Character Set (*chs*) legt fest, wie die Daten in den Tabellenfeldern zu interpretieren sind.

chs; Zeichensatz

Der mit dieser Angabe definierte Zeichensatz bezieht sich nur auf die Datensätze innerhalb der Schnittstellen-Datei. Der Zeichensatz ist für den Dateiheder, die Tabellenbeschreibungen und die Kommentare zu verwenden.

<i>Zeichensatz</i>	Bezeichnung des Zeichensatzes
Mögliche Zeichensätze:	
'ASCII'	Gewöhnlicher 7-bit-ASCII-Zeichensatz. Erlaubt sind die Zeichen 32 (sp) bis 126 (-).
'ISO8859-1'	ISO-Zeichensatz 'Latin-1'. Dieser Zeichensatz basiert auf den oben definierten erlaubten Zeichen des ASCII-Zeichensatzes und enthält die zusätzlichen Buchstaben der westeuropäischen Sprachen. Erlaubt sind die die im Anhang angegebenen Zeichen. Der von Windows verwendete 'ANSI-Zeichensatz' (WIN-CP1252) ist im Bereich der hier definierten erlaubten Zeichen mit ISO8859-1 kompatibel.

Beispiel für die Zeichensatzangabe:

chs; "ISO8859-1"

5.1.4 Version

Der Befehl Version (*ver*) sagt aus, mit welcher Version des Programmes, die in der Quellenangabe (Befehl *src*) angegeben wird, die Schnittstellen-Datei erzeugt worden ist.

ver; Version

Version Programmversion, beliebige Zeichenkette mit Textbegrenzungszeichen

Beispiel zur Versionsangabe:

ver; "1.0"

5.1.5 Interfaceversion

Der Befehl Interfaceversion (*ifv*) sagt aus, mit welcher Interfaceversion das schreibende Programm die Daten erzeugt hat. Das lesende Programm interpretiert diese Information und kann damit Versionskonflikte frühzeitig erkennen und melden.

ifv; Version

Version Interface-Version, beliebige Zeichenkette mit Textbegrenzungszeichen

Beispiel zur Interfaceversion:

ifv; "1.0"

5.2.2 Attributbezeichnungen

Mit dem Befehl Attribute (*atr*) werden die Attributnamen definiert.

atr; Bezeichnung₁; Bezeichnung₂; ...; Bezeichnung_n

Bezeichnung_i Attributbezeichnung aus dem ÖPNV-Datenmodell

Beispiel für eine Attributdefinitionen:

atr; FRT_START; FRT_FID

5.2.3 Attributformate

Der Befehl Format (*frm*) dient der Beschreibung der Formate der einzelnen Attribute. Diese Informationen werden einerseits für die Angabe des Typs, andererseits für die Ermittlung des Platzbedarfs bei der Darstellung in Kolonnenform benötigt.

frm; Format₁; Format₂; ...; Format_n

Format_i Datentyp und Platzbedarf bei der Darstellung in Kolonnenform

Gültige numerische Formate:

num[*n*.0] *n*-stellige Zahl ohne Nachkommastellen. Für die Darstellung in Kolonnenform muß noch ein Zeichen für das Vorzeichen dazugerechnet werden.

Gültige Textformate:

char[*n*] *n*-stellige Zeichenfolge (mit Textbegrenzungszeichen dargestellt)

Beispiel für Formatangaben:

frm; num[6.0]; char[8]; num[8.0]

frm; char[32]

5.2.4 Datensatz

Ein Datensatz wird mit dem Befehl Record (*rec*) eingeleitet und enthält die eigentlichen Daten. Die Attribute sind entsprechend den Attributbezeichnungen angeordnet. Je nach Darstellungsart werden die Daten frei formatiert oder in Kolonnenform dargestellt.

rec; Datenfeld₁; Datenfeld₂; ...; Datenfeld_n

Bei der **frei formatierten Darstellung** (*free*) werden die Datenfelder mit beliebigem Abstand aneinandergereiht, d.h. vor und nach den Feldtrennzeichen sind beliebig viele Leerzeichen erlaubt.

Bei der **Darstellung in Kolonnenform** (*aligned*) wird die Zahl der eingefügten Leerzeichen durch die Formatangabe (Befehl *frm*) zwingend vorgeschrieben. Als Länge des Datenfeldes verstehen wir in der nachstehenden Auflistung den Zwischenraum zwischen zwei Feldtrennzeichen:

Darstellung der Textformate in Kolonnenform:

char[*n*] Das Datenfeld hat wegen der Textbegrenzungszeichen die Länge *n*+2. Da Zeichenketten, die verdoppelte Textbegrenzungszeichen enthalten, die Ausnahme sind, kann der Einfachheit halber die Länge *n*+2 verwendet werden. Dies führt jedoch bei den erwähnten Zeichenketten zu Verschiebungen in der Darstellung.

Darstellung der numerischen Formate in Kolonnenform:

`num[n.0]` Das Datenfeld hat wegen des Vorzeichens die Länge n+1.

Bei der Implementation kann normalerweise die frei formatierte Darstellung angewendet werden. Die datenexportierende Applikation sollte jedoch eine Testfunktion enthalten, welche es erlaubt, die Daten in Kolonnenform zu exportieren. Beim Import der Daten werden Leerzeilen / Leerschläge zwischen den Datenfeldern entfernt.

Beispiel für die freiformatierte Darstellung:

```
mod; DD.MM.YY; HH:MM:SS; free
....
frm; num[3.0]; num[6.0]; char[8]
rec ;1; 2 ; "Bake 1"
```

Beispiel für die Darstellung in Kolonnenform:

```
mod; DD.MM.YY; HH:MM:SS; aligned
....
frm; num[3.0]; num[6.0]; char[8]
rec; 1; 2; "Bake 1"
```

Beispiel für die Darstellung in Kolonnenform (Textbegrenzungszeichen innerhalb einer Zeichenkette: Bake "1"):

```
mod; DD.MM.YY; HH:MM:SS; aligned
....
frm; num[3.0]; num[6.0]; char[8]
rec; 1; 2;"Bake "1"""
```

5.2.5 Tabellenende

Der Befehl End of Table (*end*) markiert das Ende der Tabelle und beinhaltet die Anzahl der Datensätze der Tabelle. Mit diesem Befehl kann einfach überprüft werden, ob alle Sätze gelesen wurden und das Ende der Tabelle erreicht worden ist.

`end;` Satzzahl

Satzzahl Anzahl der Datensätze in einer Tabelle

Beispiel für eine Tabellenendangabe:

```
end; 4711
```

5.2.6 Kommentar

Kommentare enthalten zusätzliche, dokumentierende Information. Diese Information wird vom Programm, das die Daten einliest, nicht verarbeitet. Kommentare (Befehl *com*, engl. *comment*) können in der Tabellenbeschreibung zwischen dem Tabellennamen und dem Tabellenende stehen. Kommentare dürfen nicht auf Folgezeilen fortgesetzt werden, d.h. es ist pro Kommentarzeile ein eigener Kommentarbefehl (*com*) zu verwenden.

`com;` *Kommentar*

Kommentar Beliebiger Text im ASCII-Zeichensatz mit Textbegrenzungszeichen.

Bsp. zum Kommentar:

```
com; Dies ist ein Kommentar
```


6 Beispiele

```

mod; YYYY/MM/DD; HH:MM:SS; aligned

src; "LIO-ADAPTER-VDV 4.0"; "1998/06/18"; "14:05:48"
chs; "ISO-8859-1"
ver; "Version_2.2"
ifv; "Version_1.7"
dve; "Nutzersicht_der_HPW_auf_das_OePNV-Datenmodell_4.0/4.1"
fft; "LIObus"

tbl; MENGE_ONR_TYP
atr; BASIS_VERSION; ONR_TYP_NR; STR_ONR_TYP; ONR_TYP_TEXT
frm; num[11.0]; num[11.0]; char[8]; char[32]
rec;          1;          1;          "HP";          "Haltepunkt"
rec;          1;          2;          "BHOF";        "Betriebshofpunkt"
rec;          1;          3;          "OM";          "Ortsmarke"
rec;          1;          4;          "LSA";         "LSA-Punkt"
end; 4

eof; 1

```

```

mod; DD.MM.YYYY; HH:MM:SS; aligned
src; "MFS-90"; "16.12.1997"; "14:33:07"
chs; "ISO8859-1"
ver; "V1.0"
ifv; "1.0"
dve; "1.0"
fft; "LIO"
tbl; MENGE_ONR_TYP
atr; BASIS_VERSION; ONR_TYP_NR; STR_ONR_TYP; ONR_TYP_TEXT
frm; num[10.0]; num[10.0]; char[8]; char[32]
rec; 1; 1; "HP "; "Haltepunkt"
rec; 1; 2; "BHOF"; "Betriebshofpunkt"
rec; 1; 3; "OM "; "Ortsmarke"
rec; 1; 4; "LSA "; "LSA-Punkt"
end; 0004
eof; 1

```

```

mod; DD.MM.YYYY; HH:MM:SS; aligned
src; " EPON -> LIO Schnittstelle "; "17.02.1998"; "11:05:43"
chs; "ISO8859-1"
ver; "V15.0-15"
ifv; "V4.0"
dve; "1.0"
fft; "LIO"
com;
tbl; MENGE_ONR_TYP
atr; BASIS_VERSION; ONR_TYP_NR; STR_ONR_TYP; ONR_TYP_TEXT
frm; num[9.0]; num[2.0]; char[8]; char[32]
rec;          1; 1; "HP"          ; "Haltepunkt"
rec;          1; 2; "BHOF"        ; "Betriebshof"
rec;          1; 3; "OM"          ; "Ortsmarke"
rec;          1; 4; "LSA"         ; "Lichtsignalanlage"
com;
end; 4
eof; 1

```

```

mod; DD.MM.YYYY; HH:MM:SS; free
src; "PROLIO, TransTeC-systeme, 1998"; "06.07.1998"; "10:03:34"
chs; "ISO8859-1"
ver; "1.0.3-RDB"
ifv; "1.0"
dve; "1.5"
fft; "LIO"
tbl; MENGE_ONR_TYP
atr; BASIS_VERSION; ONR_TYP_NR; STR_ONR_TYP; ONR_TYP_TEXT

```

```

frm; num[9.0]; num[2.0]; char[8]; char[32]
rec; 611; 1; "HP"; "Haltepunkt"
rec; 611; 2; "BHOF"; "Betriebshofpunkt"
rec; 611; 3; "OM"; "Ortsmarke"
rec; 611; 4; "LSA"; "LSA-Punkt"
end; 4
eof; 1

mod; YYYY/MM/DD; HH:MM:SS; aligned
src; "HOT II"; "1998/05/29"; "16:20:10"
chs; "ISO8859-1"
ver; "%I%a %G% / %I%a %G%"
ifv; "V1.0"
dve; "V0.1"
fft; "LIO"
tbl; MENGE_ONR_TYP
atr; BASIS_VERSION; ONR_TYP_NR; STR_ONR_TYP; ONR_TYP_TEXT
frm; num[9.0]; num[2.0]; char[8]; char[32]
rec; 10; 1; "HP"; "Haltepunkt"
rec; 10; 2; "BHOF"; "Betriebshofpunkt"
rec; 10; 3; "OM"; "Ortsmarke"
rec; 10; 4; "LSA"; "LSA-Punkt"
end; 4
eof; 1

```

7 Zeichensätze

7.1 ASCII Zeichensatz

Der ASCII-Zeichensatz definiert die Zeichen 0 bis 127, wobei der für die Schnittstellen-Dateien zulässige Bereich noch weiter eingeschränkt wird. Erlaubt sind nur Zeichen im Bereich 32 (sp) bis 126 (~) und die beiden Steuerzeichen Carriage Return (CR) und Line Feed (LF). Alle anderen Steuerzeichen (Form-Feed, Tabulator, etc) sind nicht erlaubt.

Dezimal	0	16	32	48	64	80	96	112
0				0	@	P	`	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			\$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8			(8	H	X	h	x
9)	9	I	Y	i	y
10	<i>LF</i>		*	:	J	Z	j	z
11			+	;	K	[k	{
12			,	<	L	\	l	
13	<i>CR</i>		-	=	M]	m	}
14			.	>	N	^	n	~
15			/	?	O	_	o	

7.2 ISO8859-1 Zeichensatz

Der Zeichensatz beinhaltet die 7-Bit-ASCII-Zeichen (siehe ASCII Zeichensatz). Erlaubt sind die folgenden Zeichen zwischen 161 (ı) und 255 (ÿ).

Dezimal	128	144	160	176	192	208	224	240
0					À		à	
1			ı	±	Á	Ñ	á	ñ
2			ç	²	Â	Ò	â	ò
3			£		Ã	Ó	ã	ó
4			¤		Ä	Ô	ä	ô
5			¥	µ	Å	Õ	å	õ
6					Æ	Ö	æ	ö
7					Ç		ç	÷
8				,	È	Ø	è	ø
9			©		É	Ù	é	ù
10			ª		Ê	Ú	ê	ú
11			«	»	Ë	Û	ë	û
12					Ì	Ü	ì	ü
13					Í		í	
14			®		Î		î	
15				ı	Ï	ß	ï	ÿ