

Beschreibungsstandard für die Datenträgerüberlassung Version 1.1 vom 1. August 2002

Erläuterungen zur Speicherung und Beschreibung von Daten im Rahmen der „Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (GDPdU)“

Beschreibungsstandard-Version 1.1 vom 1. August 2002

Änderung	Datum	Autor(en)
Beschreibungsstandard-Version 1.0	6.6.2002	Jon Chick
Beschreibungsstandard-Version 1.1	1.8.2002	Jon Chick

Dokument-Version 1.1 vom 1. August 2002

Änderung	Datum	Autor(en)
Dokument-Version 1.0	6.6.2002	Jon Chick, Tankred Giese, Martin Otto
Dokument-Version 1.1 Erläuterungen zu References Erweiterungen Version 1.1	1.8.2002	Martin Otto, Jon Chick, Tankred Giese, Michael Schleupen

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Adressaten dieses Papiers	3
Zusammenfassung/Ziel	3
Text des §147 Abs. 6 AO	4
Datenträgerüberlassung	4
Die Datenorganisation bei der Datenträgerüberlassung	7
Die Rolle des Softwareherstellers	9
XML-basierter Beschreibungsstandard	10
Überblick	10
Allgemeine Annahmen bzw. Voraussetzungen zur DTD	10
Technischer Ablauf des Importprozesses	11
Die XML - DTD	12
Organisation des Beschreibungsstandards	15
Hinweise zu XML	19
Beschreibung der Elemente	20
Beispiel 1 für eine INDEX.XML	35
Beispiel 2 für Verknüpfungen	39
Beispiel 3 für eine INDEX.XML (komprimierte Daten)	41
Häufig gestellte Fragen	42
Synopse Beschreibungsstandard Version 1 und Version 1.1	45

Adressaten dieses Papiers

- Anwender von Software, die zur Bereitstellung eines Datenträgers nach GDPdU aufgefordert werden.
- Entwickler von Software, die steuerrelevante Daten für die Datenträgerüberlassung nach GDPdU bereitstellen.

Zusammenfassung/Ziel

Nach §147 Abs.6 AO ist es der Finanzverwaltung möglich, die Daten von elektronischen Buchführungssystemen „digital“ zu prüfen, entweder durch Datenträgerüberlassung und/oder durch mittelbaren bzw. unmittelbaren Zugriff.

Für die Datenträgerüberlassung ist es notwendig, dass die Daten vom steuerpflichtigen Unternehmen (oder dem beauftragten Steuerberater, buchführenden (Sub-)Unternehmen, etc.) in „maschinell auswertbarer Form“ auf geeigneten Datenträgern bereitgestellt werden.

Unter dem Begriff „maschineller Auswertbarkeit“ versteht die Finanzverwaltung den wahlfreien Zugriff auf alle gespeicherten Daten einschließlich der Stammdaten und Verknüpfungen mit Sortier- und Filterfunktionen.

Um eine solche Auswertbarkeit oder Verwertbarkeit zu erreichen, ist es notwendig, dass die Dateiformate für die Datenträgerüberlassung definiert und standardisiert werden.

In diesem Dokument wird ein Verfahren vorgestellt, das die definierte und standardisierte Datenträgerüberlassung ermöglicht und einen Standard für die Softwareindustrie beschreibt.

Bei Rückfragen wenden Sie sich bitte an <mailto:GDPdU@audicon.net>, oder informieren Sie sich unter <http://www.audicon.net>.

Text des §147 Abs. 6 AO

[In der Fassung des Gesetzes zur Senkung der Steuersätze und zur Reform der Unternehmensbesteuerung (Steuersenkungsgesetz - StSenkG) vom 23.10.2000 BGBl. I 2000. S. 1433]

(6) Sind die Unterlagen nach Absatz 1 mit Hilfe eines Datenverarbeitungssystems erstellt worden, hat die Finanzbehörde im Rahmen einer Außenprüfung das Recht, Einsicht in die gespeicherten Daten zu nehmen und das Datenverarbeitungssystem zur Prüfung dieser Unterlagen zu nutzen. Sie kann im Rahmen einer Außenprüfung auch verlangen, dass die Daten nach ihren Vorgaben maschinell ausgewertet oder ihr die gespeicherten Unterlagen und Aufzeichnungen auf einem maschinell verwertbaren Datenträger zur Verfügung gestellt werden. Die Kosten trägt der Steuerpflichtige.

Datenträgerüberlassung

ein mögliches Szenario für eine Steuerprüfung mit Datenträgerüberlassung:

Ein Steuerprüfer bekommt den Auftrag, ein Unternehmen zu prüfen. Für bestimmte Prüfungsfelder und Prüfungszeiträume beschließt der Steuerprüfer, die Daten unter Zuhilfenahme von Prüfungssoftware zu analysieren. Entsprechend wird dem Unternehmen die Prüfung angekündigt und die Prüfungsschwerpunkte bekanntgegeben. Er bittet das Unternehmen, die steuerlich relevanten Daten auf einem maschinell auswertbarem Datenträger nach den Vorgaben der „Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen“ (BMF Schreiben vom 16.07.2001) bereitzustellen.

Bei Prüfungsbeginn erhält der Steuerprüfer eine CD-ROM als Datenträger.

Gemäß seinem Prüfungsschwerpunkt sucht der Prüfer die entsprechenden Tabellen heraus und liest diese ein. Auf der CD-ROM sind steuerrelevante Daten und beschreibende Daten enthalten, um die Daten ohne weitere Erklärungen einzulesen. Sobald die Daten in die Analysesoftware importiert sind, führt der Steuerprüfer eine oder mehrere Analysen (teilweise durch Makros automatisiert) aus.



Prüferin

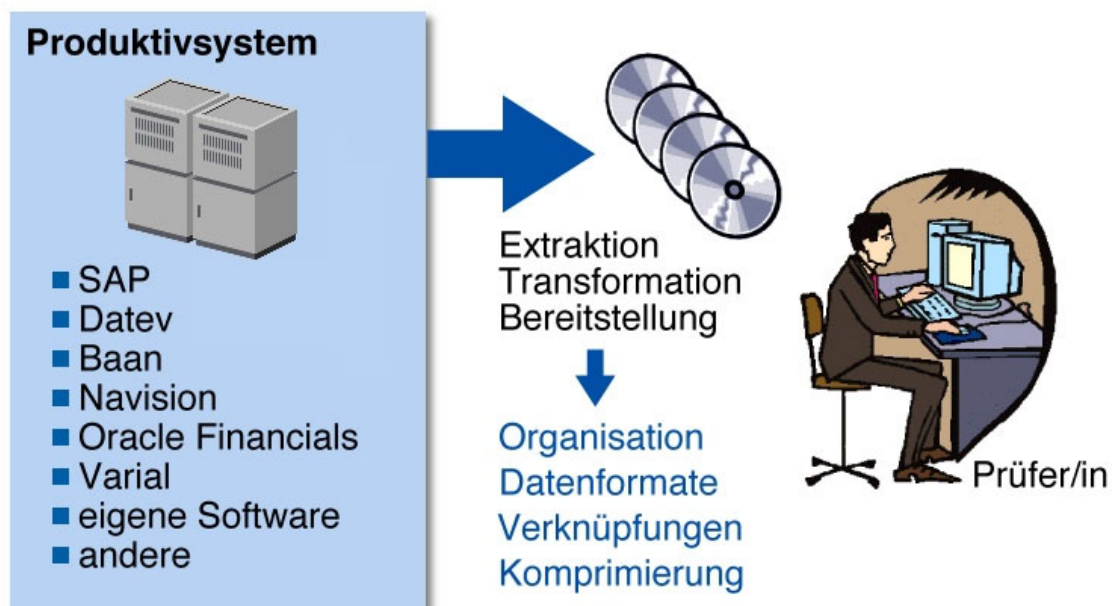
1. Fordert Daten-CD-ROM an
2. Liest Daten-CD-ROM ein
3. Führt Analyse aus bzw. lässt Analyse-Makros ablaufen

Aus Sicht des Unternehmens fallen bei dem oben beschriebenen Szenario folgende Arbeitsschritte an:

Ein Mitarbeiter initiiert die Datenextraktion bzw. Export aus der Produktionsdatenbank, wobei die Daten ggf. (automatisch) transformiert werden in dem Sinne, dass die Datenorganisation, die Struktur der Tabellen oder Objekte geändert wird.

Ebenso muss bei komprimierten oder verschlüsselten Daten die Komprimierung bzw. die Verschlüsselung aufgehoben werden. Zu den steuerrelevanten Daten muss nun noch eine Beschreibung in Form einer maschinenauswertbaren Datei erzeugt oder hinzukopiert werden. Eine solche Datei muss in dem Fall bei der Bereitstellung erzeugt werden, falls der Softwarehersteller nicht vorhersehen kann, welche Daten bei der Prüfung bereitgestellt werden sollen. Es kann eine statische Datei bereitgestellt werden, falls im Voraus der Umfang und die Struktur der Daten feststeht.

Die Daten werden anschließend automatisch oder manuell auf einem beweglichen Datenträger (CD-ROM, DVD,...) gespeichert.



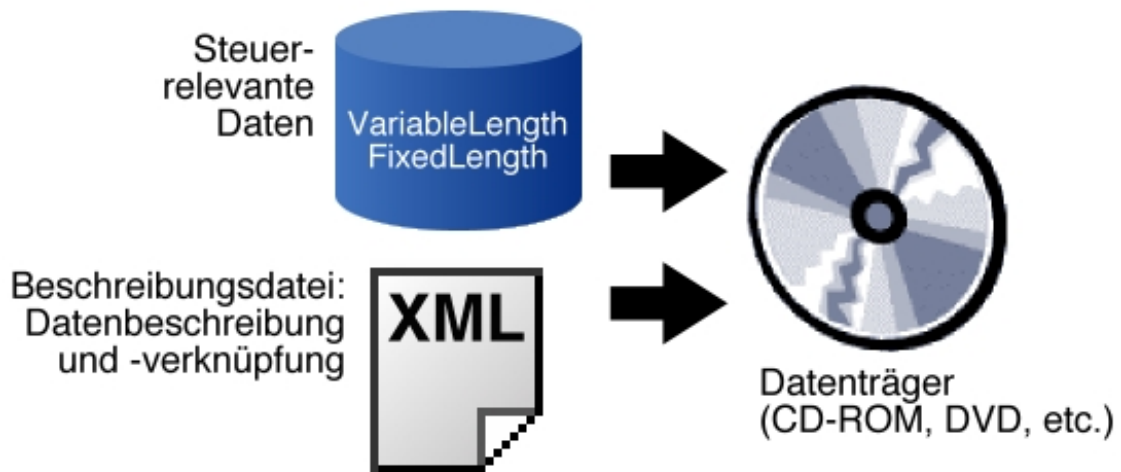
Die Datenorganisation bei der Datenträgerüberlassung

Das BMF-Schreiben vom 16.07.2001 gibt den geprüften Unternehmen bei der technischen Organisation des zu überlassenden Datenträgers grundsätzlich einen breiten Spielraum.

Nachdem die Finanzverwaltung des Bundes und der Länder eine in der Wirtschaftsprüfung verbreitete Analysesoftware (IDEA 2002) beschafft hat, sind die Steuerprüfer in der Lage, Daten maschinell zu lesen und (automatisiert oder manuell gesteuert) zu verarbeiten. Die Finanzverwaltung definiert die Datenfelder und Inhalte, die bei der Datenträgerüberlassung auf den Datenträger gebracht werden, nicht. Entsprechend ist es notwendig, dass Steuerpflichtige und/oder Hersteller von Software, die steuerlich relevanten Daten verarbeiten, selbst entscheiden, welche Daten bei der Datenträgerüberlassung bereitgestellt werden. Dabei können je nach EDV-System, Software-Strukturen, Unternehmensstruktur und -Größe unterschiedliche Dateninhalte sinnvoll und notwendig sein.

Um Rückfragen durch den Steuerprüfer über die Dateninhalte und Formate zu minimieren oder unnötig zu machen, wird hier ein Verfahren beschrieben, das die Daten, Datenstrukturen und Verknüpfungen beschreibt und kommentiert.

Um die unterschiedlichen Datenstrukturen verarbeiten zu können, stellt der Steuerpflichtige oder sein Berater die steuerrelevanten Daten weitgehend denormalisiert zur Verfügung. Zusätzlich liefert er eine maschinenauswertbare Beschreibung der Daten, Datenformate, und Verknüpfungen. Das Dateiformat für die steuerlich relevanten Daten wurde auf gängige Standardformate festgelegt. Beides, steuerrelevante Daten und Beschreibungsdaten, werden auf einem gemeinsamen Datenträger bereitgestellt.



Die unterstützten Dateiformate für die steuerlich relevanten Daten sind:

- VariableLength
- FixedLength

Damit ist es den Softwaresystemen möglich, bereits bestehende Exportfilter oder Speicherungsmechanismen zu nutzen.

Die Finanzverwaltung erlaubt keine Installation von Spezialsoftware zur Entschlüsselung oder zum Dekomprimieren von Daten auf ihren Systemen. Insbesondere beim Postversand von Datenträgern ist jedoch zumindest das Verschlüsseln von Daten aus Datenschutzgründen erforderlich. Im Beschreibungsstandard ist deshalb vorgesehen, z.B. Programme zur Entschlüsselung oder zum Dekomprimieren direkt vom Datenträger zu starten ohne dass sich diese auf dem System der Finanzverwaltung installieren.

Die Computer der Prüfer sind mit Zugriffsschutz-Software und ggf. Anti-Viren-Software ausgestattet. Die Funktionsfähigkeit dieser direkt aufgerufenen Programme sollte deshalb vom Daten-Bereitsteller im Vorfeld geklärt werden. Die Prüfung auf Lauffähigkeit dieser Routinen ist nicht Gegenstand des Beschreibungsstandard.

Die Rolle des Softwareherstellers

Im Folgenden wird als „Softwarehersteller“ - aus Sicht des steuerpflichtigen Unternehmens - ein externer Lieferant von Software oder einer Datenverarbeitungsanlage inklusive Programmen oder eine interne Abteilung, die entsprechende Programme bzw. Services bereitstellt, angenommen.

Gemäß den dargestellten Szenarien zur Betriebsprüfung muss bei der Betriebsprüfung u.a. eine maschinenauswertbare Beschreibung der Daten und der Verknüpfungen bereitgestellt werden.

Die Beschreibungsdatei wird von einem Programm gelesen, das gemäß dieser Beschreibung die Import-Funktion des Analyseprogramms des Steuerprüfers steuert. Es werden keine Operationen auf Datenbanken der Produktivsysteme ausgeführt.

Der Hersteller der Programme bzw. Prozeduren, mit denen das steuerpflichtige Unternehmen steuerrelevante Daten verarbeitet (Softwarehersteller), muss zusätzlich zu einer Extraktionsfunktionalität die Aufbereitung oder Bereitstellung der Beschreibungsfunktionalität sicherstellen.

XML-basierter Beschreibungsstandard

Überblick

XML ist die „eXtensible Markup Language“. XML gibt nur einen generischen Satz von Sprachelementen vor, so dass je nach Anwendungsfall eine eigene Sprache definiert werden muss. Die jeweils neu formulierte Sprache kann durch Strukturinformationen und Elementdefinitionen in einer DTD (Document Type Definition) maschinenlesbar hinterlegt werden. Damit kann ein XML-Parser überprüfen, ob ein XML-Dokument der vereinbarten Sprache entspricht (validieren). Man unterscheidet also zwischen Sprachbeschreibung in der DTD und den Inhalten in der XML-Instanz.

In der DTD selbst werden also keine Beschreibungsdaten für die Datenträgerüberlassung gespeichert, sondern in der XML Datei INDEX.XML. Die DTD beschreibt den Aufbau der INDEX.XML Datei.

Weiterführende Informationen zu XML findet man auf www.w3c.org.

Allgemeine Annahmen bzw. Voraussetzungen zur DTD

Es wird in der DTD davon ausgegangen, dass die steuerrelevanten Daten in Dateien organisiert sind. Pro Tabelle wird genau eine Datei angenommen, so dass die Organisation nach Datei und Tabelle physisch zusammenfallen.

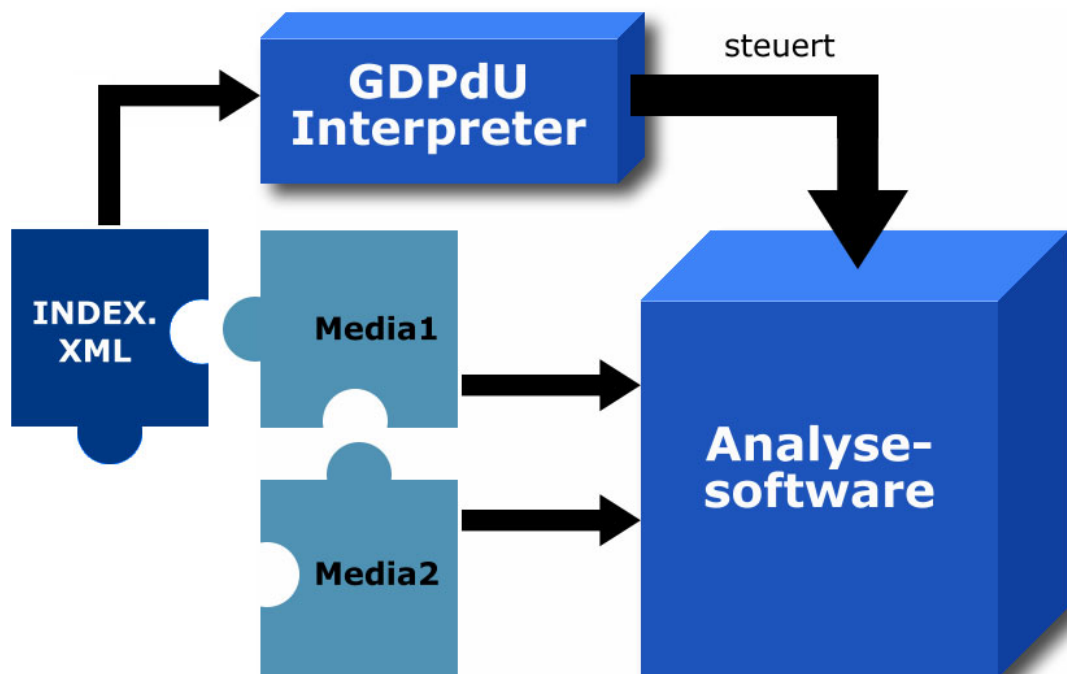
Die Beschreibungsdaten für die Datenträgerüberlassung werden in einer Datei INDEX.XML gespeichert.

Die DTD muss sich in demselben Verzeichnis auf dem Datenträger befinden wie die INDEX.XML Datei.

Werden mehrere Datenträger (Medien) bereitgestellt, muss sich die INDEX.XML nur auf dem ersten Datenträger befinden.

Technischer Ablauf des Importprozesses

Der Importprozess wird von einer Komponente gesteuert, die die INDEX.XML Datei interpretiert und die Exporttabellen von den einzelnen Datenträgern an die Analyse-Software übergibt. Diese Komponente überprüft die INDEX.XML Datei auf Korrektheit und dient dem Prüfer als Inhaltsverzeichnis für die gelieferten Daten.



Die XML - DTD

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Versions available:
1.1 (August-01-2002)
-->
<!-- Start Simple Types -->
<!-- Supplementary Vocabulary -->
<!ELEMENT Version (#PCDATA)>
<!ELEMENT Location (#PCDATA)>
<!ELEMENT Comment (#PCDATA)>
<!ELEMENT Length (#PCDATA)>
<!ELEMENT References (#PCDATA)>
<!ELEMENT From (#PCDATA)>
<!ELEMENT To (#PCDATA)>
<!-- Specifying a maximum length for a VariableLength column can
reduce a VariableLength tables' import time. If MaxLength
is not specified then we parse URL to determine the MaxLength
for each column.

* Only applies to VariableLength tables. -->
<!ELEMENT MaxLength (#PCDATA)>
<!-- Specifies which character (if any) encapsulates a
VariableLength AlphaNumeric column.

Doublequote is the default TextEncapsulator "

* Only applies to VariableLength tables. (Optional) -->
<!ELEMENT TextEncapsulator (#PCDATA)>
<!-- Specifies how many digits appear to the right of the decimal symbol.

CAUTION: Results are undefined when importing numeric data with
greater Accuracy than the Accuracy defined in index.xml

For example trying to import the value 1000,25 with an
accuracy of 0 might result in 1000 or an error. This
behavior is specific to the implementation.

Zero is the default Accuracy '0' (Optional)
-->
<!ELEMENT Accuracy (#PCDATA)>
<!-- The decimal place is not always stored with numbers. If each number
is supposed to have decimal places use ImpliedAccuracy -->
<!ELEMENT ImpliedAccuracy (#PCDATA)>
<!-- Enables you to change how GDPdU displays dates.
DD.MM.YYYY is the default Format -->
<!ELEMENT Format (#PCDATA)>
<!-- Specifies the symbol that indicates decimal values.
Comma is the default DecimalSymbol. ','
Specified once per Table. -->
<!ELEMENT DecimalSymbol (#PCDATA)>
<!-- Specifies the symbol that groups the digits in large numbers.
Dot is the default DigitGroupingSymbol or ThousandsSeperator. '.'
Specified once per Table -->
<!ELEMENT DigitGroupingSymbol (#PCDATA)>
<!-- Command(s) are executed in the following manner
* before the import process
* after the import process
* before a Media is imported
* after a Media is imported
-->
<!ELEMENT Command (#PCDATA)>

```

```

<!-- Only the file protocol is supported at this time.

* The standard uses relative URLs.

Absolute URLs are not allowed. The following are all invalid:
* http://www.somewhere.com/data/Accounts.dat
* ftp://ftp.somewhere.com/data/Accounts.dat
* file://localhost/Accounts.dat
* file:///Accounts.dat

The following are valid examples
* Accounts.dat
* data/Accounts.dat
* data/january/Accounts.dat
* ../Accounts.dat
-->
<!ELEMENT URL (#PCDATA)>
<!-- Textual description of specified element (Optional) -->
<!ELEMENT Description (#PCDATA)>
<!-- The logical name of specified element.
Sometimes referred to business name.

If missing, URL will be used in place of Name. -->
<!ELEMENT Name (#PCDATA)>
<!-- Y2K Window Any year before Epoch is 2000+
Default value 30. -->
<!ELEMENT Epoch (#PCDATA)>
<!-- Element(s) that separate columns or records.
Semicolon is the default ColumnDelimiter. ';'
CRLF or &#13;&#10; is the default RecordDelimiter. -->
<!ELEMENT ColumnDelimiter (#PCDATA)>
<!ELEMENT RecordDelimiter (#PCDATA)>
<!-- The number of bytes skipped before reading of URL commences.
Zero is the default when not specified. '0'
-->
<!ELEMENT SkipNumBytes (#PCDATA)>
<!-- End Simple Types -->
<!-- Start Complex Types -->
<!-- Self-explanatory -->
<!ELEMENT Range (From, (To | Length)?>
<!ELEMENT FixedRange (From, (To | Length)>
<!-- The document element -->
<!ELEMENT DataSet (Version, DataSupplier?, Command*, Media+, Command*)>
<!-- Supported datatypes (mandatory) -->
<!ELEMENT AlphaNumeric EMPTY>
<!ELEMENT Date (Format?)>
<!ELEMENT Numeric ((ImpliedAccuracy | Accuracy)?>
<!-- Supported codepages:
Be careful to explicitly define RecordDelimiter when using
a non-default codepage.

ANSI is the default codepage when not specified -->
<!ELEMENT ANSI EMPTY>
<!ELEMENT Macintosh EMPTY>
<!ELEMENT OEM EMPTY>
<!ELEMENT UTF16 EMPTY>
<!ELEMENT UTF7 EMPTY>
<!ELEMENT UTF8 EMPTY>
<!-- Supported file formats:
FixedLength
VariableLength -->
<!ELEMENT FixedLength ((Length | RecordDelimiter)?, ((FixedPrimaryKey+, FixedColumn*) | (FixedColumn+), ForeignKey*)>
<!ELEMENT FixedColumn (Name, Description?, (Numeric | AlphaNumeric | Date), Map*, FixedRange)>
<!ELEMENT FixedPrimaryKey (Name, Description?, (Numeric | AlphaNumeric | Date), Map*, FixedRange)>

```

```

<!ELEMENT VariableLength (ColumnDelimiter?, RecordDelimiter?, TextEncapsulator?, ((VariablePrimaryKey+,
VariableColumn*) | (VariableColumn+)), ForeignKey*)>
<!ELEMENT VariableColumn (Name, Description?, (Numeric | (AlphaNumeric, MaxLength?) | Date), Map*)>
<!ELEMENT VariablePrimaryKey (Name, Description?, (Numeric | (AlphaNumeric, MaxLength?) | Date), Map*)>
<!-- Description of the entity supplying the data. (Optional) -->
<!ELEMENT DataSupplier (Name, Location, Comment)>
<!-- The first Media will contain index.xml. Importing will process each media listed -->
<!ELEMENT Media (Name, Command*, Table+, Command*)>
<!-- Elements common to FixedLength & VariableLength are propagated to Table. -->
<!ELEMENT Table (URL, Name?, Description?, Validity?, (ANSI | Macintosh | OEM | UTF16 | UTF7 | UTF8)?,
(DecimalSymbol, DigitGroupingSymbol)?, SkipNumBytes?, Range?, Epoch?, (VariableLength | FixedLength))>
<!-- ForeignKeys denote joins or relationships between tables.
To successfully join two tables make sure both the PrimaryKey
and the referenced column (foreignkey) are of the same datatype.
Results are undefined when joining two tables with different
key datatypes. Most likely an error will occur. -->
<!ELEMENT ForeignKey (Name+, References)>
<!-- Maps AlphaNumeric columns from 'From' to 'To'
ie. From      To
=====
      True      1
      True     -1
      False     0

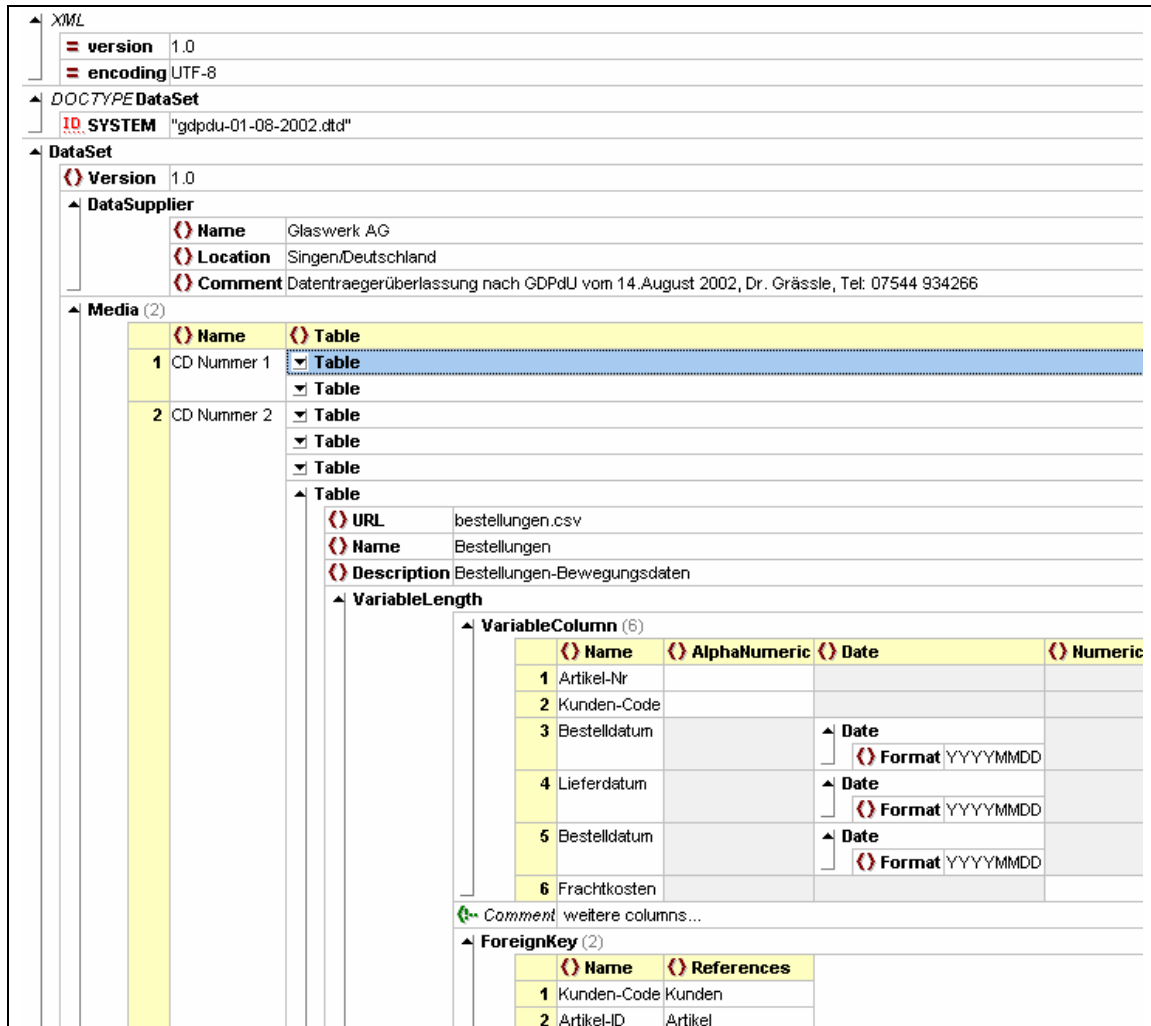
Basically, a map is an associative container.

The standard implementation only supports
AlphaNumeric datatypes. The following
conversions are NOT supported.

Numeric   -> AlphaNumeric
Date      -> AlphaNumeric
AlphaNumeric -> Date
AlphaNumeric -> Numeric
-->
<!ELEMENT Map (Description?, From, To)>
<!-- Documentation for table validity. -->
<!ELEMENT Validity (Range, Format?)>
<!-- End Complex Types -->

```

Organisation des Beschreibungsstandards

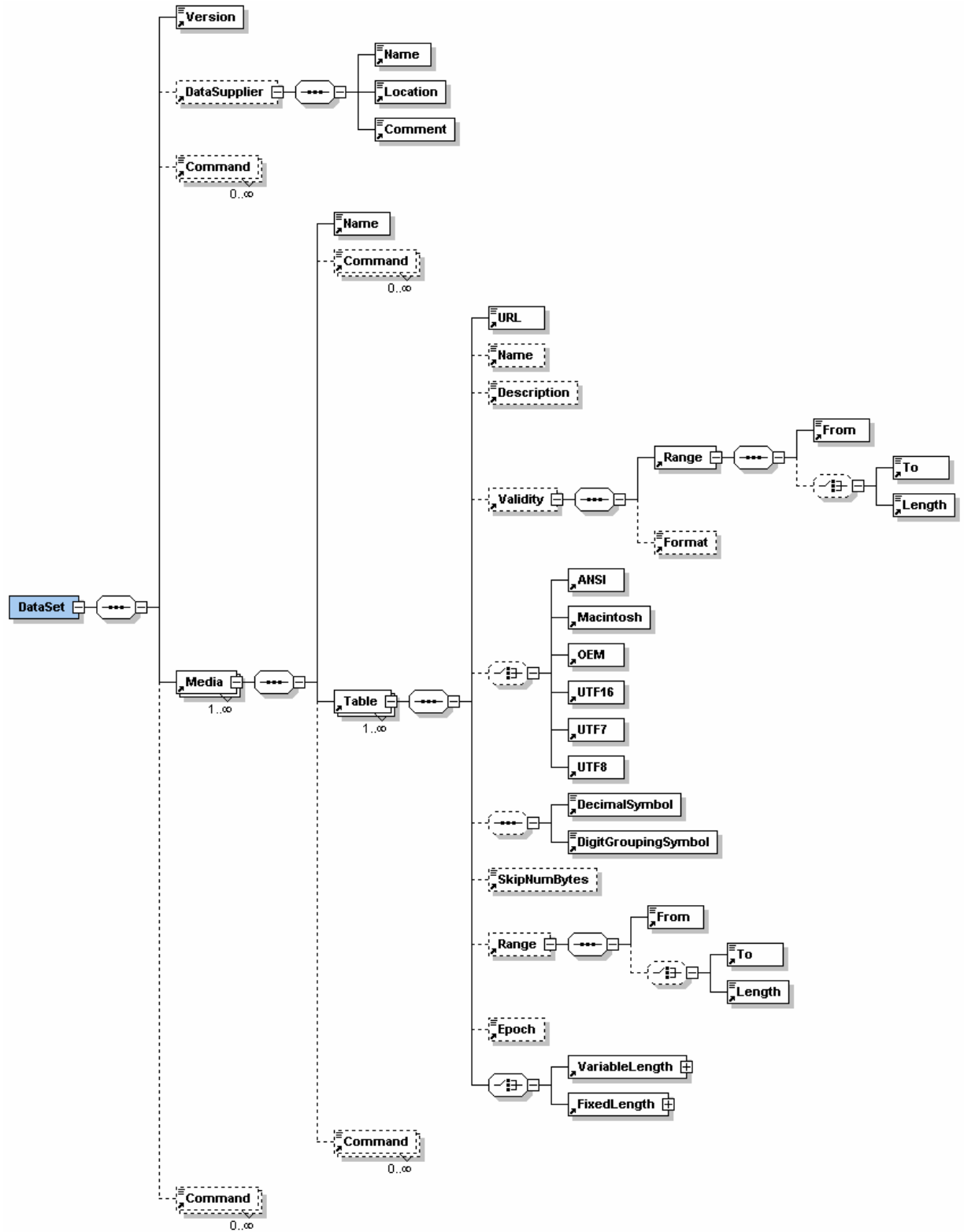


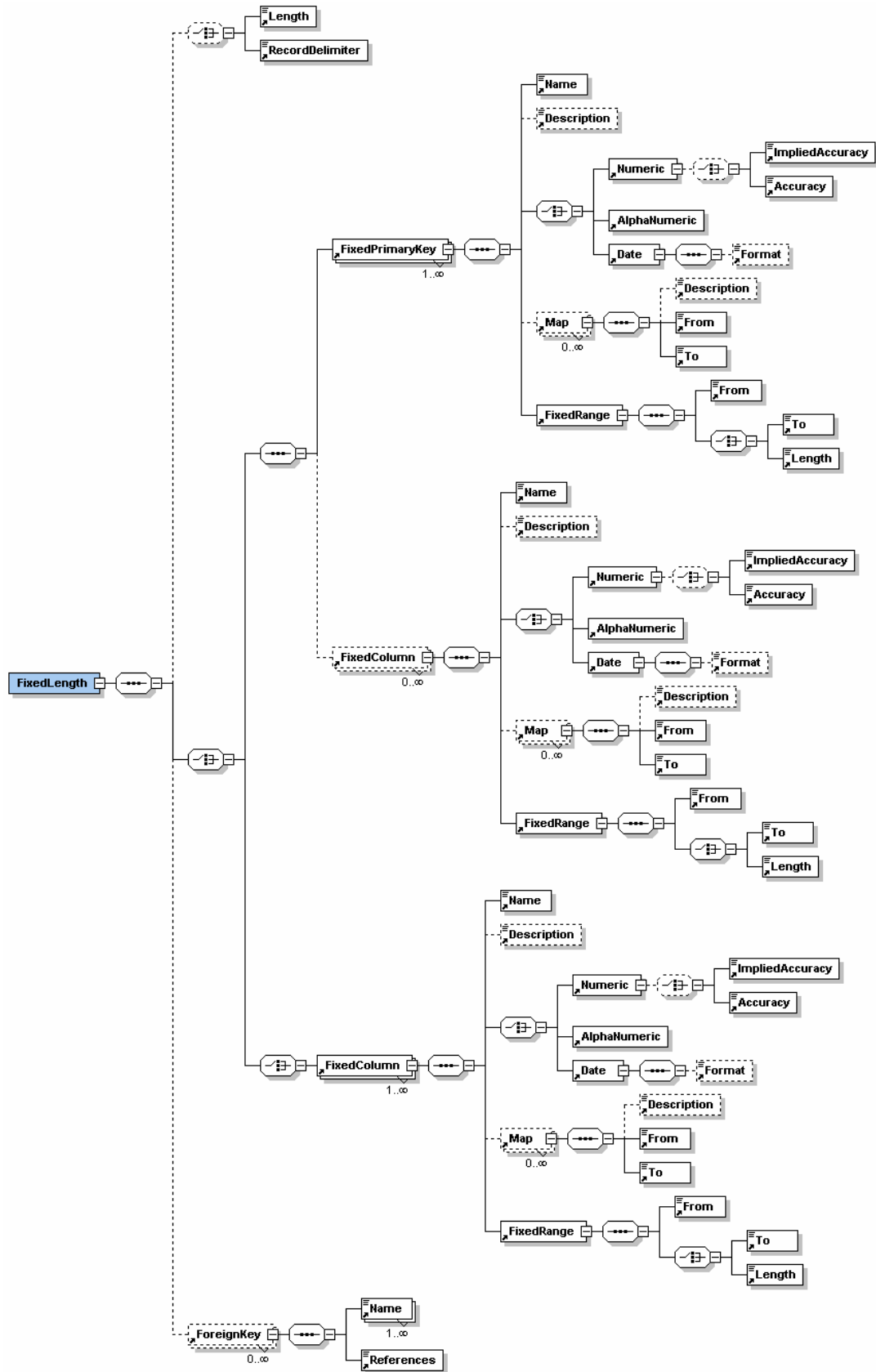
Name	AlphaNumeric	Date	Numeric
1 Artikel-Nr			
2 Kunden-Code			
3 Bestelldatum		▲ Date Format YYYYMMDD	
4 Lieferdatum		▲ Date Format YYYYMMDD	
5 Bestelldatum		▲ Date Format YYYYMMDD	
6 Frachtkosten			

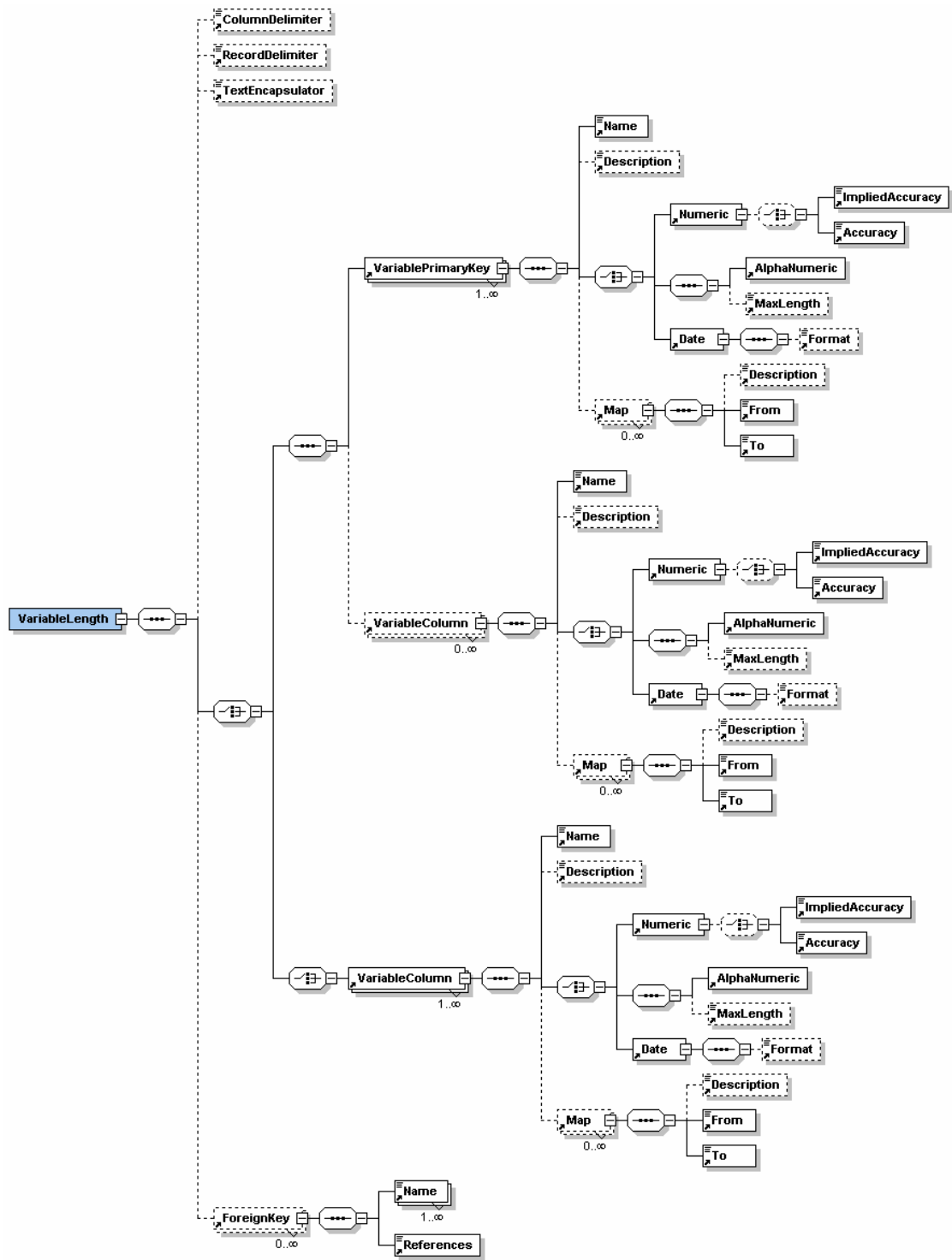
Name	References
1 Kunden-Code	Kunden
2 Artikel-ID	Artikel

Das Basiselement des Beschreibungsstandards ist das DataSet-Element. Hier enthalten sind die Elemente Version, DataSupplier, Command, und Media.

- Version enthält die Versionsnummer der Datenträgerbereitstellung.
- DataSupplier enthält Angaben über die Herkunft der Daten.
- Die Media-Container enthalten die Angaben über die bereitgestellten Datenträger und Tabellen, die darauf gespeichert sind. Hierbei wird für jedes bereitgestellte Datenträgermedium (CD-ROM) ein eigener Media-Eintrag vorgenommen. Somit ist es möglich, auch mehrere Datenträger (CD-ROMs) in einem DataSet zusammenzufassen.







Hinweise zu XML

- Hinweis zu den in XML üblichen Operatoren:

Operator	Elementtyp darf auftreten
?	0-mal oder 1-mal
*	0-mal oder beliebig oft
+	1-mal oder beliebig oft

- #PCDATA steht für parsed character Data und bedeutet, dass das Element für einen Texteintrag vorgesehen ist.
- Bei Aufzählungen werden die möglichen Werte durch senkrechte Striche („|“) getrennt.
- In XML muss auf Groß- und Kleinschreibung geachtet werden.
- Bitte verwenden Sie in den Beschreibungsfeldern und/oder den Tabellen- und Spaltennamen keine der Zeichen

" & < >

Verwenden Sie stattdessen Namen oder Nummer aus der untenstehenden Liste.

Zeichen	Name	Nummer
"	"	"
&	&	&
<	<	<
>	>	>

Beschreibung der Elemente

Es folgt eine Beschreibung der einzelnen ELEMENT-Deklarationen.

Im Allgemeinen gilt, dass die in den jeweiligen Attributen „description“ hinterlegten Texte dem Prüfer als Kommentar angezeigt werden. Die Länge des Textes in den description-Attributen sollte 255 Zeichen nicht überschreiten.

<!ELEMENT Version (#PCDATA)>

Enthält die Versionsnummer der Datenträgerbereitstellung. Dieses Element hat keine technische Auswirkung, sondern dient zur Beschreibung.

Verwendet in Container	
DataSet	

<!ELEMENT Location (#PCDATA)>

Beschreibt den Standort des Datenlieferanten.

Verwendet in Container	
DataSupplier	

<!ELEMENT Comment (#PCDATA)>

Kommentarfeld für zusätzliche Informationen zum Datenlieferanten.

Verwendet in Container	
DataSupplier	

<!ELEMENT Length (#PCDATA)>

enthält Informationen über die Länge von Attributen oder Datensätzen

Verwendet in Container	
Range	

<!ELEMENT References (#PCDATA)>

enthält Informationen über Verknüpfungen

Verwendet in Container	Kontext-abhängige Beschreibung
ForeignKey	referenzierte Tabelle

Siehe hierzu auch das Beispiel auf Seite 39.

<!ELEMENT From (#PCDATA)>

enthält Informationen über den Startwert eines Wertebereichs.

Verwendet in Container	Kontext-abhängige Beschreibung
Range	Startwert in einem Wertebereich
Map	Redefinition von Daten

<!ELEMENT To (#PCDATA)>

enthält Informationen über den Endwert eines Wertebereichs.

Verwendet in Container	Kontext-abhängige Beschreibung
Range	Endwert in einem Wertebereich
Map	Redefinition von Daten

<!ELEMENT MaxLength (#PCDATA)>

enthält Informationen über die maximale Länge eines alphanumerischen Feldes in einer VariableLength Tabelle.

Verwendet in Container	Kontext-abhängige Beschreibung
VariableColumn	Maximale Länge des Feldes
VariablePrimaryKey	Maximale Länge des Feldes

Die Angabe von MaxLength bei dem Dateityp VariableLength beschleunigt den Importprozess, da andernfalls die Datei vor dem Importprozess geprüft wird und die maximalen Längen der Felder ermittelt wird.

<!ELEMENT TextEncapsulator (#PCDATA)>

Bei VariableLength-Dateien kann man Textfelder durch ein „Encapsulator“-Zeichen einschließen, z.B. für den Fall, dass der Feldtrenner in den Daten vorkommt.

Vorgabewert:	"
--------------	---

Verwendet in Container	
VariableLength	

<!ELEMENT Accuracy (#PCDATA)>

Anzahl der Nachkommastellen. Vorsicht bei Daten, deren Genauigkeit grösser ist, als in Accuracy angegeben.

Vorgabewert:	0
--------------	---

Verwendet in Container	
Numeric	

<!ELEMENT ImpliedAccuracy (#PCDATA)>

Anzahl der Nachkommastellen, auch wenn in den Quelldaten keine Nachkommastellen angegeben sind.

Beispiel:

Quelldaten: „100; 200; 102; 5000;6587890“

ImpliedAccuracy 3: „0,100; 0,200; 0,102; 5,000; 6587,890“

Verwendet in Container	
Numeric	

<!ELEMENT Format (#PCDATA)>

Angabe über Formatdefinitionen.

Vorgabewert:	DD.MM.YYYY
--------------	------------

Verwendet in Container	Kontext-abhängige Beschreibung
Date	<p>Hier wird die Maskierung für die Darstellung von Datumsangaben hinterlegt.</p> <p>Folgende Angaben für Format sind gültig:</p> <ul style="list-style-type: none"> ○ DD für Tag, ○ MM für Monat, ○ YY bzw. YYYY für Jahr. <ul style="list-style-type: none"> • Beispiel (30. September 2001) <ul style="list-style-type: none"> ○ MM/DD/YY: 09/30/01 ○ DD.MM.YY: 30.09.01 ○ MM/DD/YYYY: 09/30/2001 ○ YYYY-MM-DD: 2001-09-30 ○ DDMMYY: 300901 ○ DDMMYYYY: 30092001
Validity	Maskierung für das verwendete Datumsformat (siehe Date)

<!ELEMENT DecimalSymbol (#PCDATA)>

Angabe über das Dezimalstellen-Trennzeichen

Vorgabewert:	,
--------------	---

Verwendet in Container	
Table	

<!ELEMENT DigitGroupingSymbol (#PCDATA)>

Definiert das Tausender-Trennzeichen.

Vorgabewert:	.
Verwendet in Container	
Table	

<!ELEMENT Command (#PCDATA)>

Command definiert ein Betriebssystemkommando.

Diese können an folgenden Stellen ausgeführt werden:

- vor dem gesamten Importprozess
- nach dem gesamten Importprozess
- bevor ein einzelner Datenträger importiert wird
- nachdem ein einzelner Datenträger importiert wurde

Verwendet in Container	Kontext-abhängige Beschreibung
DataSet	Command für den gesamten Vorgang
Media	Command für einen einzelnen Datenträger

Siehe hierzu auch das Beispiel auf Seite 41.

<!ELEMENT URL (#PCDATA)>

Dieses Element enthält den physischen Dateinamen der bereitgestellten Datei.

In dieser Version des Beschreibungsstandards werden ausschliesslich relative URLs unterstützt. Als Basisverzeichnis gilt der Pfad zur INDEX.XML.

Beispiele für ungültige Einträge:

<http://www.somewhere.com/data/Accounts.csv>

<ftp://ftp.somewhere.com/data/Accounts.csv>

Beispiel für gültige Einträge:

Accounts.dat

data/Accounts.dat

data/january/Accounts.dat

../Accounts.dat

Verwendet in Container	
Table	

<!ELEMENT Description (#PCDATA)>

Beschreibungsfeld..

Verwendet in Container	Kontext-abhängige Beschreibung
FixedColumn	Beschreibungsfeld für Spalten in FixedLength-Dateien
FixedPrimaryKey	Beschreibungsfeld für Schlüssel in FixedLength-Dateien
VariableColumn	Beschreibungsfeld für Spalten in VariableLength-Dateien
VariablePrimaryKey	Beschreibungsfeld für Schlüssel in VariableLength-Dateien
Table	Beschreibungsfeld für Tabellen
Map	Beschreibungsfeld für Redefinition von Daten

<!ELEMENT Name (#PCDATA)>

Beschreibungsfeld.

Vorgabewert:	Angabe aus dem URL Element
--------------	----------------------------

Verwendet in Container	Kontext-abhängige Beschreibung
FixedColumn	Spaltenname in FixedLength-Dateien
FixedPrimaryKey	Schlüsselname in FixedLength-Dateien
VariableColumn	Spaltenname in VariableLength-Dateien
VariablePrimaryKey	Schlüsselname in VariableLength-Dateien
DataSupplier	Name des Datenlieferanten (Firmenname,...)
Media	Datenträger-Name
URL	Physischer Tabellenname
ForeignKey	Name des Fremdschlüssels

<!ELEMENT Epoch (#PCDATA)>

Für Daten, die eine 2-ziffrige Jahresangabe haben, kann ein Standardplatzhalter definiert werden, der 19xx von 20xx trennt. Dieser Wert ist mit 30 vorbelegt.

In einigen Dateien könnte das Datum 25. November 55 entweder 1955 als auch 2055 bedeuten. Mit Epoch wird eingestellt, bis zu welchem zweistelligen Wert das Jahr als im nächsten Jahrhundert liegend interpretiert werden soll. Steht beispielsweise im Feld Jahrhundert-Trennwert der Wert 50, so werden alle Jahresangaben von 0 bis 49 als die Jahre 2000 bis 2049, die Jahresangaben 50 bis 99 als 1950 bis 1999 interpretiert.

Vorgabewert:	30
--------------	----

Verwendet in Container	
Table	

<!ELEMENT ColumnDelimiter(#PCDATA)>

Definiert Trennzeichen für Datenfelder.

Vorgabewert:	; (Semikolon „;“)
Verwendet in Container	Kontext-abhängige Beschreibung
VariableLength	Datenfeld-Trennzeichen

Beispiel für übliche Trennzeichen:

Trennzeichen	XML-Notation
Komma „,“	,
Semikolon „;“	;
Tab		

<!ELEMENT RecordDelimiter(#PCDATA)>

Definiert Trennzeichen für Datensätze.

Vorgabewert:	
 (CRLF)
Verwendet in Container	Kontext-abhängige Beschreibung
FixedLength	Datensatz-Trennzeichen
VariableLength	Datensatz-Trennzeichen

Beispiel für übliche Trennzeichen:

Trennzeichen	XML-Notation
CR Carriage Return	
CRLF	

LF Linefeed	

<!ELEMENT SkipNumBytes (#PCDATA)>

Falls eine Datei am Beginn unerwünschte Daten enthält wie z.B. Datei-Header, ist hier anzugeben, wie viele Bytes übersprungen und nicht gelesen werden sollen.

Vorgabewert:	0
--------------	---

Verwendet in Container	Kontext-abhängige Beschreibung
Table	

<!ELEMENT Range (From, (To | Length)?)>

Definiert einen Bereich oder einen Startwert.
Startwert bedeutet: wird kein *To* oder *Length* angegeben, definiert dieser Range nur einen Startwert (z.B. eine Datensatzposition).

Verwendet in Container	Kontext-abhängige Beschreibung
Table	Startdatensatz
Validity	Gültigkeitsbereich

<!ELEMENT FixedRange (From, (To | Length))>

Definiert einen Bereich mit Angabe von *From* und *To* oder *From* und *Length*.

Verwendet in Container	Kontext-abhängige Beschreibung
FixedColumn	Feldposition in FixedLength-Dateien
FixedPrimaryKey	Schlüsselposition in FixedLength-Dateien

<!ELEMENT DataSet (Version, DataSupplier?, Command*, Media+, Command*)>

Das Element *DataSet* ist das oberste Element (document-Element) in der Hierarchie des XML-Dokumentes. Das Element *DataSet* ist Träger der Version, Datenherkunft, Vorlauf- und Nachlauf-Prozesse und der Medien, die die Tabellen enthalten.

<!ELEMENT AlphaNumeric EMPTY>

Definiert den Datentyp als alphanumerisch.

Verwendet in Container	Kontext-abhängige Beschreibung
FixedColumn	Typ des Feldes in FixedLength-Datei
FixedPrimaryKey	Typ des Schlüssels in FixedLength-Datei
VariableColumn	Typ des Feldes in VariableLength-Dateien
VariablePrimaryKey	Typ des Schlüssels in VariableLength-Dateien

<!ELEMENT Date (Format?)>

Definiert den Datentyp als Datum.

Verwendet in Container	Kontext-abhängige Beschreibung
FixedColumn	Typ des Feldes in FixedLength-Datei
FixedPrimaryKey	Typ des Schlüssels in FixedLength-Datei
VariableColumn	Typ des Feldes in VariableLength-Dateien
VariablePrimaryKey	Typ des Schlüssels in VariableLength-Dateien

<!ELEMENT Numeric ((ImpliedAccuracy | Accuracy?)>

Definiert den Datentyp als numerisch.

Verwendet in Container	Kontext-abhängige Beschreibung
FixedColumn	Typ des Feldes in FixedLength-Datei
FixedPrimaryKey	Typ des Schlüssels in FixedLength-Datei
VariableColumn	Typ des Feldes in VariableLength-Dateien
VariablePrimaryKey	Typ des Schlüssels in VariableLength-Dateien

<!ELEMENT ANSI EMPTY>

Legt die Verwendung der Codepage ANSI fest.

<!ELEMENT Macintosh EMPTY>

Legt die Verwendung der Codepage Macintosh fest.

<!ELEMENT OEM EMPTY>

Legt die Verwendung der Codepage IBM-PC-ASCII fest.

<!ELEMENT UTF16 EMPTY>

Legt die Verwendung der Codepage UTF16 fest.

<!ELEMENT UTF7 EMPTY>

Legt die Verwendung der Codepage UTF7 fest.

<!ELEMENT UTF8 EMPTY>

Legt die Verwendung der Codepage UTF8 fest.

Vorgabewert:	ANSI
--------------	------

Plattform	Zeichensatz
Unix Rechner	7-Bit-ASCII-Zeichensatz (7-Bit-ISO Code, US-Variante), nur Zeichen mit ASCII-Code kleiner 128, also Buchstaben, Ziffern und Sonderzeichen (Punkt, Komma etc.) ohne nationale Sonderzeichen (Umlaute etc.) und ohne Graphikzeichen. Zeilentrenner ist LF (Line Feed), das Zeichen 10.
PC unter DOS	8-Bit-Zeichensatz "IBM-PC-ASCII", der in den unteren 127 Zeichen dem 7-Bit-ASCII entspricht und ab Zeichen 128 nationale Sonderzeichen und (Semi-) Graphikzeichen enthält. Zeilentrenner ist die Folge CR und LF, das sind die Zeichen 12 und 10 (Carriage Return, Line Feed).
PC unter Windows	8-Bit-ANSI-Zeichensatz, der bis 127 ASCII entspricht und oberhalb 127 nationale Sonderzeichen enthält, die nicht mit denen des IBM-PC-ASCII übereinstimmen. Zeilentrenner ist wie unter DOS die Folge CR, LF. (Spezielle Anwendungen können auch einen 16-Bit-Code, den Unicode benutzen.)
Apple Macintosh	Macintosh spezifischer 8-Bit-Zeichensatz "Mac-ASCII", der bis 127 ASCII entspricht und oberhalb 127 nationale Sonderzeichen enthält. Zeilentrenner ist CR.

Die Wahl der Codepage setzt keine Vorgabe für den RecordDelimiter. Sie müssen den RecordDelimiter immer explizit angeben, wenn der Vorgabewert (CRLF) nicht geeignet ist.

<!ELEMENT FixedLength ((Length | RecordDelimiter)?, ((FixedPrimaryKey+, FixedColumn*) | (FixedColumn+)), ForeignKey*)>

Definiert eine Datei vom Typ FixedLength.

Verwendet in Container	
Table	

<!ELEMENT FixedColumn (Name, Description?, (Numeric | AlphaNumeric | Date), Map*, FixedRange)>

Definiert eine Spalte (=Column) in einer Datei vom Typ FixedLength.

Verwendet in Container	
FixedLength	

<!ELEMENT FixedPrimaryKey (Name, Description?, (Numeric | AlphaNumeric | Date), Map*, Range)>

Definiert ein Schlüsselfeld (=Key) in einer Datei vom Typ FixedLength.

Verwendet in Container	
FixedLength	

Siehe hierzu auch das Beispiel auf Seite 39.

<!ELEMENT VariableLength (ColumnDelimiter?, RecordDelimiter?, TextEncapsulator?, ((VariablePrimaryKey+, VariableColumn*) | (VariableColumn+))?, ForeignKey*)>

Definiert eine Datei vom Typ VariableLength.

Verwendet in Container	
Table	

Die Felder (VariableColumn, VariablePrimaryKey) müssen in der Reihenfolge definiert werden, in der sie in der Datei geliefert werden.

<!ELEMENT VariableColumn (Name, Description?, (Numeric | AlphaNumeric, MaxLength ? | Date), Map*)>

Definiert eine Spalte (=Column) in einer Datei vom Typ VariableLength.

Verwendet in Container	
VariableLength	

Die Felder (VariableColumn, VariablePrimaryKey) müssen in der Reihenfolge definiert werden, in der sie in der Datei geliefert werden.

<!ELEMENT VariablePrimaryKey (Name, Description?, (Numeric | AlphaNumeric, MaxLength ? | Date), Map*)>

Definiert ein Schlüsselfeld (=Key) in einer Datei vom Typ VariableLength.

Verwendet in Container	
VariableLength	

Die Felder (VariableColumn, VariablePrimaryKey) müssen in der Reihenfolge definiert werden, in der sie in der Datei geliefert werden.

Zu Verknüpfungen siehe auch das Beispiel auf Seite 39.

<!ELEMENT DataSupplier (Name, Location, Comment)>

Enthält Angaben zum Datenlieferanten.

Verwendet in Container	
DataSet	Lieferant (Firma,...)

<!ELEMENT Media (Name, Command*, Table+, Command*)>

Definiert den Inhalt eines Datenträgers.

Verwendet in Container	
DataSet	Datenträgerüberlassung

<!ELEMENT Table (URL, Name?, Description?, Validity?, (ANSI | Macintosh | OEM | UTF16 | UTF7 | UTF8)?, (DecimalSymbol, DigitGroupingSymbol)?, SkipNumBytes?, Range?, (VariableLength | FixedLength)?)>

Definiert eine Tabelle.

Verwendet in Container	
Media	Tabelle

<!ELEMENT ForeignKey (Name+, References)>

Definiert eine Spalte als Fremdschlüssel.

Die Spalte, auf die mit Name Bezug genommen wird, muss in der gleichen Table als FixedColumn oder VariableColumn definiert sein.

Verwendet in Container	Kontext-abhängige Beschreibung
FixedLength	Feld in FixedLength-Datei, referenzierte Tabelle
VariableLength	Feld in FixedLength-Datei, referenzierte Tabelle

Siehe hierzu auch das Beispiel auf Seite 39.

<!ELEMENT Map (Description?, From, To)>

Mit dem Element MAP können Redefinitionen auf Datenebene vorgenommen werden.

Beispiel:

From (Ausgangsdaten)	To (Zieldaten)
1	wahr
0	falsch
Hr	Herr
Fr	Frau
01	Verheiratet
02	Ledig
03	Steuerklasse 3

Verwendet in Container	Kontext-abhängige Beschreibung
FixedColumn	Feld in FixedLength-Datei
FixedPrimaryKey	Schlüssel in FixedLength-Datei
VariableColumn	Feld in VariableLength-Dateien
VariablePrimaryKey	Schlüssel in VariableLength-Dateien

Map darf nur für alphanumerische Felder benutzt werden.

Sie sollten vermeiden, in der Datenträgerüberlassung kodierte Felder zu liefern, sondern alle Codes in die textuelle Beschreibung umwandeln. Wenn diese Umwandlung nicht bereits von dem Exportprogramm vorgenommen wird, ist Map eine Möglichkeit, Codes über die Datenbeschreibung umzuwandeln.

<!ELEMENT Validity (Range, Format?)>

Definiert den Gültigkeitszeitraum der Daten.

Verwendet in Container	
Table	Gültigkeitszeitraum

Beispiel 1 für eine INDEX.XML

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DataSet SYSTEM "gdpdu-01-08-2002.dtd">
<DataSet>
  <Version>1.0</Version>
  <DataSupplier>
    <Name>Glaswerk AG</Name>
    <Location>Singen/Deutschland</Location>
    <Comment>Datentraegerüberlassung nach GDPdU vom 14.August 2002, Dr. Grässle, Tel: 07544
934266</Comment>
  </DataSupplier>
  <Media>
    <Name>CD Nummer 1</Name>
    <Table>
      <URL>Account.csv</URL>
      <Name>Account</Name>
      <Description>Accounts Receivables</Description>
      <!-- Diese Tabelle gilt von 1.Januar 2000- 31.Dezember 2002 -->
      <Validity>
        <Range>
          <From>20000101</From>
          <To>20021231</To>
        </Range>
        <Format>YYYYMMDD</Format>
      </Validity>
      <!-- Spezifiziert den Dezimaltrenner -->
      <DecimalSymbol>.</DecimalSymbol>
      <!-- Spezifiziert den Tausender-Separator -->
      <DigitGroupingSymbol>,</DigitGroupingSymbol>
      <VariableLength>
        <!-- ein zusammengesetzter Primaerschlüssel -->
        <VariablePrimaryKey>
          <Name>RegionId</Name>
          <AlphaNumeric/>
        </VariablePrimaryKey>
        <VariablePrimaryKey>
          <Name>Id</Name>
          <AlphaNumeric/>
        </VariablePrimaryKey>
        <VariableColumn>
          <Name>Account Description</Name>
          <AlphaNumeric/>
        </VariableColumn>
        <VariableColumn>
          <Name>Balance</Name>
          <Numeric/>
        </VariableColumn>
        <VariableColumn>
          <Name>Old Balance</Name>
          <Numeric/>
        </VariableColumn>
      </VariableLength>
    </Table>
    <Table>
      <URL>Region.csv</URL>
      <VariableLength>
        <VariablePrimaryKey>
          <Name>RegionId</Name>
          <AlphaNumeric/>
        </VariablePrimaryKey>

```

```

        <VariableColumn>
          <Name>Description</Name>
          <AlphaNumeric/>
        </VariableColumn>
      </VariableLength>
    </Table>
  </Media>
  <Media>
    <Name>CD Nummer 2</Name>
    <Table>
      <URL>Sales.csv</URL>
      <Range>
        <From>56</From>
      </Range>
      <FixedLength>
        <FixedPrimaryKey>
          <Name>SalesId</Name>
          <AlphaNumeric/>
          <FixedRange>
            <!-- Position des Feldes: -->
            <From>1</From>
            <To>10</To>
          </FixedRange>
        </FixedPrimaryKey>
        <FixedColumn>
          <Name>RegionId</Name>
          <AlphaNumeric/>
          <FixedRange>
            <From>11</From>
            <Length>10</Length>
          </FixedRange>
        </FixedColumn>
        <FixedColumn>
          <Name>Id</Name>
          <AlphaNumeric/>
          <FixedRange>
            <From>21</From>
            <To>30</To>
          </FixedRange>
        </FixedColumn>
        <FixedColumn>
          <Name>SalesComplete</Name>
          <AlphaNumeric/>
          <!-- Redefinition des Wertes "1" nach "True" und "0" nach "False" -->
          <Map>
            <From>1</From>
            <To>True</To>
          </Map>
          <Map>
            <From>0</From>
            <To>False</To>
          </Map>
          <FixedRange>
            <From>31</From>
            <To>32</To>
          </FixedRange>
        </FixedColumn>
        <!-- ein zusammengesetzter Fremdschlüssel der den Primaerschlüssel in Tabelle Account
referenziert -->
        <ForeignKey>
          <Name>RegionId</Name>
          <Name>Id</Name>
          <References>Account</References>
        </ForeignKey>
      </FixedLength>
    </Table>
  </Media>

```

```

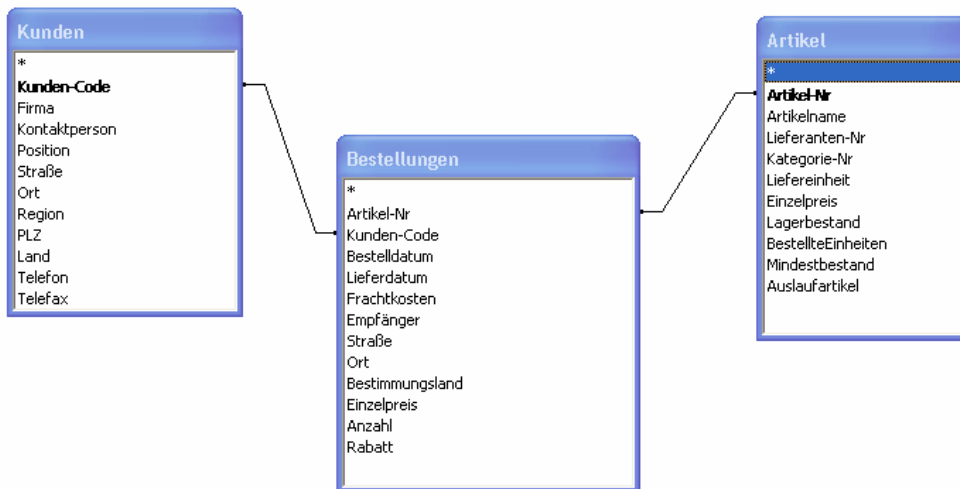
<Table>
  <URL>kunden.csv</URL>
  <Name>Kunden</Name>
  <Description>Kunden-Stammdaten</Description>
  <VariableLength>
    <VariablePrimaryKey>
      <Name>Kunden-Code</Name>
      <AlphaNumeric/>
    </VariablePrimaryKey>
    <VariableColumn>
      <Name>Firma</Name>
      <AlphaNumeric/>
    </VariableColumn>
    <VariableColumn>
      <Name>Kontaktperson</Name>
      <AlphaNumeric/>
    </VariableColumn>
    <!-- weitere columns... -->
  </VariableLength>
</Table>
<Table>
  <URL>artikel.csv</URL>
  <Name>Artikel</Name>
  <Description>Artikel-Stammdaten</Description>
  <VariableLength>
    <VariablePrimaryKey>
      <Name>Artikel-Nr</Name>
      <AlphaNumeric/>
    </VariablePrimaryKey>
    <VariableColumn>
      <Name>Artikelname</Name>
      <AlphaNumeric/>
    </VariableColumn>
    <VariableColumn>
      <Name>Lieferanten-Nr</Name>
      <AlphaNumeric/>
    </VariableColumn>
    <!-- weitere columns... -->
  </VariableLength>
</Table>
<Table>
  <URL>bestellungen.csv</URL>
  <Name>Bestellungen</Name>
  <Description>Bestellungen-Bewegungsdaten</Description>
  <VariableLength>
    <VariableColumn>
      <Name>Artikel-Nr</Name>
      <AlphaNumeric/>
    </VariableColumn>
    <VariableColumn>
      <Name>Kunden-Code</Name>
      <AlphaNumeric/>
    </VariableColumn>
    <VariableColumn>
      <Name>Bestelldatum</Name>
      <Date>
        <Format>YYYYMMDD</Format>
      </Date>
    </VariableColumn>
    <VariableColumn>
      <Name>Lieferdatum</Name>
      <Date>
        <Format>YYYYMMDD</Format>
      </Date>
    </VariableColumn>
    <VariableColumn>
  
```

```
<Name>Bestelldatum</Name>
<Date>
  <Format>YYYYMMDD</Format>
</Date>
</VariableColumn>
<VariableColumn>
  <Name>Frachtkosten</Name>
  <Numeric/>
</VariableColumn>
<!-- weitere columns... -->
<ForeignKey>
  <Name>Kunden-Code</Name>
  <References>Kunden</References>
</ForeignKey>
<ForeignKey>
  <Name>Artikel-ID</Name>
  <References>Artikel</References>
</ForeignKey>
</VariableLength>
</Table>
</Media>
</DataSet>
```

Beispiel 2 für Verknüpfungen

In der Datenträgerüberlassung kann es sinnvoll sein, Bewegungsdaten und Stammdaten nicht in einer breiten Tabelle zu liefern, sondern in getrennten Tabellen abzuspeichern:

Beispiel:



In den Stammdaten werden die Primärschlüssel beschrieben:

<pre> <Table> <URL>kunden.csv</URL> <Name>Kunden</Name> <Description>Kunden-Stammdaten</Description> <VariableLength> <VariablePrimaryKey> <Name>Kunden-Code</Name> <AlphaNumeric/> </VariablePrimaryKey> <VariableColumn> <Name>Firma</Name> <AlphaNumeric/> </VariableColumn> <VariableColumn> <Name>Kontaktperson</Name> <AlphaNumeric/> </VariableColumn> <!-- weitere columns... --> </VariableLength> </Table> </pre>	<pre> <Table> <URL>artikel.csv</URL> <Name>Artikel</Name> <Description>Artikel-Stammdaten</Description> <VariableLength> <VariablePrimaryKey> <Name>Artikel-Nr</Name> <AlphaNumeric/> </VariablePrimaryKey> <VariableColumn> <Name>Artikelname</Name> <AlphaNumeric/> </VariableColumn> <VariableColumn> <Name>Lieferanten-Nr</Name> <AlphaNumeric/> </VariableColumn> <!-- weitere columns... --> </VariableLength> </Table> </pre>
---	--

Die Bewegungsdaten beschreiben die Verknüpfungen zu den Stammdaten über die Angabe der Fremdschlüsselbeziehung:

```

<Table>
  <URL>bestellungen.csv</URL>
  <Name>Bestellungen</Name>
  <Description>Bestellungen-Bewegungsdaten</Description>
  <VariableLength>
    <VariableColumn>
      <Name>Artikel-Nr</Name>
      <AlphaNumeric/>
    </VariableColumn>
    <VariableColumn>
      <Name>Kunden-Code</Name>
      <AlphaNumeric/>
    </VariableColumn>
    <VariableColumn>
      <Name>Bestelldatum</Name>
      <Date>
        <Format>YYYYMMDD</Format>
      </Date>
    </VariableColumn>
    <VariableColumn>
      <Name>Lieferdatum</Name>
      <Date>
        <Format>YYYYMMDD</Format>
      </Date>
    </VariableColumn>
    <VariableColumn>
      <Name>Bestelldatum</Name>
      <Date>
        <Format>YYYYMMDD</Format>
      </Date>
    </VariableColumn>
    <VariableColumn>
      <Name>Frachtkosten</Name>
      <Numeric/>
    </VariableColumn>
    <!-- weitere columns... -->
    <ForeignKey>
      <Name>Kunden-Code</Name>
      <References>Kunden</References>
    </ForeignKey>
    <ForeignKey>
      <Name>Artikel-Nr</Name>
      <References>Artikel</References>
    </ForeignKey>
  </VariableLength>
</Table>

```

Hinweis: Während die Beschreibung eines VariablePrimaryKey die Spalte vollständig beschreibt, definiert ForeignKey keine neue Spalte. Die im ForeignKey benutzte Spalte muss vorher bereits als VariableColumn oder FixedColumn definiert worden sein.

Beispiel 3 für eine INDEX.XML (komprimierte Daten)

Index.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DataSet SYSTEM "gdpdu-01-08-2002.dtd">
<DataSet>
  <Version>1.2</Version>
  <DataSupplier>
    <Name>Audicon GmbH</Name>
    <Location>Germany</Location>
    <Comment>A compressed sample DataSet</Comment>
  </DataSupplier>
  <Command>uncompress.bat</Command>
  <Media>
    <Name>Disk 1</Name>
    <Table>
      <URL>dummy.csv</URL>
      <VariableLength>
        <VariablePrimaryKey>
          <Name>dummy</Name>
          <Numeric/>
        </VariablePrimaryKey>
      </VariableLength>
    </Table>
  </Media>
</DataSet>
```

uncompress.bat

```
rem -----
rem uncompress.bat
rem expandiert gdpdu.zip in das windows-temp-verzeichnis, unterverzeichnis gdpdu
rem -----
mkdir %temp%\gdpdu
pkunzip gdpdu.zip %temp%\gdpdu
```

%temp% ist eine Betriebssystem-Umgebungsvariable, die den Pfad zum Temporär-Verzeichnis repräsentiert.

Die Datei gdpdu.zip enthält die gepackte Datenträgerbereitstellung (einschließlich INDEX.XML und GDPDU-01-08-2002.DTD). Der Prüfer öffnet nach dem Dekomprimieren also die neue Datenbereitstellung in dem Verzeichnis %TEMP%\gdpdu.

Hinweis: Die <Version>-Angabe bezieht sich nicht auf die Version des Beschreibungsstandards sondern auf die Version der Datenträgerüberlassung.

Häufig gestellte Fragen

- *müssen wir die Daten im XML-Format liefern?*
Nein! Nur die Beschreibung der Daten (INDEX.XML) ist im XML-Format. Die Nutzdaten sind im FixedLength oder VariableLength Format.
- *kann Ihr System mit Daten in DOS/Unix/Mac-Formaten (Zeilentrenner) umgehen?*
Ja, denn die Zeilentrenner können explizit angegeben werden.
- *Wie sind negative Zahlen darzustellen (vor/hinter der Zahl, Leerzeichen zwischen Vorzeichen und Zahl erlaubt)?*
Negativen Zahlen ist ein Minuszeichen ohne Leerstelle vor- oder nachzustellen
(z.B. „-1782,90“ oder „1782,90-“).
- *Wenn eine überlassene Datei zu groß für einen Datenträger ist, kann sie dann auf mehrere Datenträger gesplittet werden?*
Der Beschreibungsstandard sieht ein Splitten einer Datei nicht vor. In diesem Fall können Sie aber die Daten komprimieren oder ein Speichermedium mit größerer Kapazität (z.B. DVD) wählen.
- *Welchen XML-Parser verwenden Sie?*
Wir verwenden Xerces (<http://xml.apache.org/xerces-c/index.html>).
- *Ist es ein Fehler, wenn ForeignKey eine Tabelle referenziert, die nicht im DataSet enthalten ist, oder würde der ForeignKey dann ignoriert werden?*
Die Beschreibung der Verlinkungen muss mit den gelieferten Daten übereinstimmen, andernfalls ist Ihre Datenträgerüberlassung nicht konsistent.
Der Import-Prozess weist den Prüfer auf die Inkonsistenz hin.
- *Was passiert, wenn in der index.xml eine Tabelle beschrieben wird, die aber anschließend nicht vorhanden ist?*
Die Beschreibung der Dateien muss mit den gelieferten Dateien übereinstimmen, andernfalls ist Ihre Datenträgerüberlassung nicht konsistent.
Der Import-Prozess weist den Prüfer auf die Inkonsistenz hin.

- *Sind Commands zwingend erforderlich? Stellen Sie für Commands Umgebungsvariablen zur Verfügung?*
Commands sind nicht zwingend erforderlich. Wenn eine .BAT oder .CMD Datei aufgerufen wird, stehen die üblichen Betriebssystem-Umgebungsvariablen zur Verfügung. Der Import-Prozess setzt selbst keine Umgebungsvariablen.

- *Wie kann abschließend geprüft werden, ob die Daten des Datenträgers von Ihrem System eingelesen werden kann?*
Die sichere Variante der Prüfung ist der Import mit IDEA für GDPdU (IDEA 2002 + SmartX).
Audicon plant ein SDK (Software Development Kit), das u.a. eine Verifizierungs-Möglichkeiten zur Verfügung stellt.

Eine einfache Möglichkeit, die XML-Datei auf ihre syntaktische Korrektheit zu testen, ist, sie mit einem XML-fähigen Browser zu öffnen (z.B. dem Microsoft Internet Explorer ab Version 5.5).

- *Da ich mit XML-Standards nicht vertraut bin, ist mir nicht klar, wie ich die Dateien: INDEX.XML und GDPDU.DTD konkret aufzubauen habe. Ich habe Ihnen eine VariableLength-Datei zugeschickt, können Sie mir die korrespondierende XML- bzw. DTD-Datei erstellen?*
Sie sollten sich Grundkenntnisse in XML aneignen, die DTD-Datei müssen (dürfen!) Sie nicht modifizieren.
Die Beschreibung der Daten (also die INDEX.XML) sollte immer von der Stelle aus geschehen, die die Datenhoheit besitzt, also vom Softwarehersteller oder dem steuerpflichtigen Unternehmen.
Eine Informationsquelle sind die GDPdU-Seminare von Audicon, die auch auf die technischen Details eingehen (siehe www.GDPdU.com).

- *In einigen Pressemitteilungen werden außer VariableLength und FixedLength noch zahlreiche weitere Dateiformate genannt. Welche Formate sind für die Datenträgerüberlassung relevant?*
Die in einigen Artikeln genannten Dateiformate beziehen sich auf die originären Import-Möglichkeiten der Software IDEA.
Der Beschreibungsstandard ist aber produktunabhängig und unterstützt in der aktuellen Version nur die beiden allgemeingültigen Formate VariableLength und FixedLength.

- *Muss die GDPDU-01-08-2002.DTD mit auf die CD-ROM gebrannt werden?*
Ja. Eine CD-ROM, die zur Datenträgerüberlassung dient, muss die GDPDU-01-08-0220.DTD, die INDEX.XML und alle Nutzdaten (z.B. T1.CSV, T2.CSV, T3.CSV, T4.ASC) enthalten.
- *In einer VariableLength-Datei sind im ersten Datensatz die Feldnamen gespeichert. Wie kann man den ersten Satz überlesen?*
Verwenden Sie die Range-Angabe im Table-Element:

```
<Table>  
  <Range>  
    <From>2</From>  
  </Range>  
  <URL>Region.csv</URL>  
  <VariableLength>  
    <VariablePrimaryKey>  
      <Name>RegionId</Name>  
      <AlphaNumeric/>  
    </VariablePrimaryKey>  
    <VariableColumn>  
      <Name>Description</Name>  
      <AlphaNumeric/>  
    </VariableColumn>  
  </VariableLength>  
</Table>
```

- *Wird durch die Angabe eines Zeichensatzes, z.B. UTF7 oder Macintosh, die Angabe des Zeilenbegrenzers automatisch übersteuert?*
Nein! Nur die explizite Angabe des Zeilenbegrenzers (RecordDelimiter) übersteuert den Vorgabewert (CRLF).

Synopse Beschreibungsstandard Version 1 und Version 1.1

Umbenennungen sind **blau** gekennzeichnet, neue oder gelöschte Elemente **rot**.

Version 1.0 vom 10.Mai 2002 (6.6.2002)	Version 1.1 vom 1.August 2002
<pre> <?xml version="1.0" encoding="UTF-8"?> <!--Versions available: 1.0 (May-10-2002) --> <!-- Start Simple Types --> <!-- Supplementary Vocabulary --> <!ELEMENT Version (#PCDATA)> <!ELEMENT Location (#PCDATA)> <!ELEMENT Comment (#PCDATA)> <!ELEMENT Length (#PCDATA)> <!ELEMENT References (#PCDATA)> <!ELEMENT From (#PCDATA)> <!ELEMENT To (#PCDATA)> <!-- Specifies which character (if any) encapsulates a CSV text column --> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <!--Versions available: 1.1 (August-01-2002) --> <!-- Start Simple Types --> <!-- Supplementary Vocabulary --> <!ELEMENT Version (#PCDATA)> <!ELEMENT Location (#PCDATA)> <!ELEMENT Comment (#PCDATA)> <!ELEMENT Length (#PCDATA)> <!ELEMENT References (#PCDATA)> <!ELEMENT From (#PCDATA)> <!ELEMENT To (#PCDATA)> <!-- Specifying a maximum length for a VariableLength column can reduce a VariableLength tables' import time. If MaxLength is not specified then we parse URL to determine the MaxLength for each column. * Only applies to VariableLength tables. --> <!ELEMENT MaxLength (#PCDATA)> <!-- Specifies which character (if any) encapsulates a VariableLength AlphaNumeric column. Doublequote is the default TextEncapsulator " * Only applies to VariableLength tables. (Optional) -- </pre>

<p><!ELEMENT TextEncapsulator (#PCDATA)></p> <p><!-- Specifies how many digits appear to the right of the decimal symbol. --></p> <p><!ELEMENT Accuracy (#PCDATA)></p> <p><!-- The decimal place is not always stored with numbers. If each number is supposed to have decimal places use ImpliedAccuracy --></p> <p><!ELEMENT ImpliedAccuracy (#PCDATA)></p> <p><!-- The first line in CSV URL contains column headers.</p> <p>A value of False is interpreted if the element is missing. Any columns specified in the XML file take precedence over the file. This lets you override some columns in the file and leave the others as they are.</p> <p>--></p> <p><!ELEMENT NamesFirstRow EMPTY></p> <p><!-- Enables you to change how gdpdu displays dates. The default is implementation dependant. --></p> <p><!ELEMENT Format (#PCDATA)></p> <p><!-- Specifies the symbol that indicates decimal values. The default is implementation dependant. Specified once per Table. --></p>	<p>></p> <p><!ELEMENT TextEncapsulator (#PCDATA)></p> <p><!-- Specifies how many digits appear to the right of the decimal symbol.</p> <p>CAUTION: Results are undefined when importing numeric data with greater Accuracy than the Accuracy defined in index.xml</p> <p>For example trying to import the value 1000,25 with an accuracy of 0 might result in 1000 or an error. This behavior is specific to the implementation.</p> <p>Zero is the default Accuracy '0' (Optional) --></p> <p><!ELEMENT Accuracy (#PCDATA)></p> <p><!-- The decimal place is not always stored with numbers. If each number is supposed to have decimal places use ImpliedAccuracy --></p> <p><!ELEMENT ImpliedAccuracy (#PCDATA)></p> <p><!-- Enables you to change how GDPdU displays dates. DD.MM.YYYY is the default Format --></p> <p><!ELEMENT Format (#PCDATA)></p> <p><!-- Specifies the symbol that indicates decimal values. Comma is the default DecimalSymbol. ',' Specified once per Table. --></p>
---	---

<pre> <!ELEMENT DecimalSymbol (#PCDATA)> <!-- Specifies the symbol that groups the digits in large numbers. The default is implementation dependant. Specified once per Table --> <!ELEMENT DigitGroupingSymbol (#PCDATA)> <!-- Command(s) are executed in the following manner * before the import process * after the import process * before a Media is imported * after a Media is imported --> <!ELEMENT Command (#PCDATA)> <!-- Only the file protocol is supported at this time. The root path is the path of Media The following are all invalid * http://www.somewhere.com/data/Accounts.csv * ftp://ftp.somewhere.com/data/Accounts.csv * ... The following is valid. * file://Accounts.csv --> <!ELEMENT URL (#PCDATA)> <!-- Textual description of specified element --> <!ELEMENT Description (#PCDATA)> </pre>	<pre> <!ELEMENT DecimalSymbol (#PCDATA)> <!-- Specifies the symbol that groups the digits in large numbers. Dot is the default DigitGroupingSymbol or ThousandsSeparator. '.' Specified once per Table --> <!ELEMENT DigitGroupingSymbol (#PCDATA)> <!-- Command(s) are executed in the following manner * before the import process * after the import process * before a Media is imported * after a Media is imported --> <!ELEMENT Command (#PCDATA)> <!-- Only the file protocol is supported at this time. * The standard uses relative URLs. Absolute URLs are not allowed. The following are all invalid: * http://www.somewhere.com/data/Accounts.dat * ftp://ftp.somewhere.com/data/Accounts.dat * file://localhost/Accounts.dat * file:///Accounts.dat The following are valid examples * Accounts.dat * data/Accounts.dat * data/january/Accounts.dat * ../Accounts.dat --> <!ELEMENT URL (#PCDATA)> <!-- Textual description of specified element (Optional) --> <!ELEMENT Description (#PCDATA)> </pre>
---	---

<pre> <!-- The logical names of specified element --> <!ELEMENT Name (#PCDATA)> <!-- Y2K Window Any year before Epoch is 2000+ Default value 50. --> <!ELEMENT Epoch (#PCDATA)> <!-- Specifies the symbol that seperates CSV columns or ASCIIFixedLength records. --> <!ELEMENT Delimiter (#PCDATA)> <!-- The number of bytes skipped before reading of URL commences. --> <!ELEMENT SkipNumBytes (#PCDATA)> <!-- End Simple Types --> <!-- Start Complex Types --> <!-- Self-explanatory --> <!ELEMENT Range (From, (To Length)?)> <!-- The document element --> <!ELEMENT DataSet (Version, DataSupplier?, Command*, Media+, Command*)> <!-- Supported datatypes AlphaNumeric is the default datatype when a datatype is not defined. --> <!ELEMENT AlphaNumeric EMPTY> <!ELEMENT Date (Format?, Epoch?)> <!ELEMENT Numeric ((ImpliedAccuracy Accuracy)?)> </pre>	<pre> <!-- The logical name of specified element. Sometimes referred to business name. If missing, URL will be used in place of Name. --> <!ELEMENT Name (#PCDATA)> <!-- Y2K Window Any year before Epoch is 2000+ Default value 30. --> <!ELEMENT Epoch (#PCDATA)> <!-- Element(s) that separate columns or records. Semicolon is the default ColumnDelimiter. ';' CRLF or &#13;&#10; is the default RecordDelimiter. --> <!ELEMENT ColumnDelimiter (#PCDATA)> <!ELEMENT RecordDelimiter (#PCDATA)> <!-- The number of bytes skipped before reading of URL commences. Zero is the default when not specified. '0' --> <!ELEMENT SkipNumBytes (#PCDATA)> <!-- End Simple Types --> <!-- Start Complex Types --> <!-- Self-explanatory --> <!ELEMENT Range (From, (To Length)?)> <!ELEMENT FixedRange (From, (To Length))> <!-- The document element --> <!ELEMENT DataSet (Version, DataSupplier?, Command*, Media+, Command*)> <!-- Supported datatypes (mandatory) --> <!ELEMENT AlphaNumeric EMPTY> <!ELEMENT Date (Format?)> <!ELEMENT Numeric ((ImpliedAccuracy Accuracy)?)> </pre>
---	---


```

<!-- Supported codepages:
Be careful to explicitly define RecordDelimiter when using a
non-default codepage.
ANSI is the default codepage when not specified -->
<!ELEMENT ANSI EMPTY>
<!ELEMENT Macintosh EMPTY>
<!ELEMENT OEM EMPTY>
<!ELEMENT UTF16 EMPTY>
<!ELEMENT UTF7 EMPTY>
<!ELEMENT UTF8 EMPTY>

<!-- Supported file formats: FixedLength, VariableLength -->
<!ELEMENT FixedLength ((Length | RecordDelimiter)?,
((FixedPrimaryKey+, FixedColumn*) | (FixedColumn+)),
ForeignKey*)>
<!ELEMENT FixedColumn (Name, Description?, (Numeric |
AlphaNumeric | Date), Map*, FixedRange)>
<!ELEMENT FixedPrimaryKey (Name, Description?, (Numeric |
AlphaNumeric | Date), Map*, FixedRange)>
<!ELEMENT VariableLength (ColumnDelimiter?, RecordDelimiter?,
TextEncapsulator?, ((VariablePrimaryKey+, VariableColumn*) |
(VariableColumn+)), ForeignKey*)>
<!ELEMENT VariableColumn (Name, Description?, (Numeric |
AlphaNumeric, MaxLength?) | Date), Map*)>
<!ELEMENT VariablePrimaryKey (Name, Description?, (Numeric |
AlphaNumeric, MaxLength?) | Date), Map*)>

<!-- Description of the entity supplying the data. (Optional)
-->
<!ELEMENT DataSupplier (Name, Location, Comment)>

<!-- The first Media will contain index.xml. Importing will
process each media listed -->
<!ELEMENT Media (Name, Command*, Table+, Command*)>

<!-- Elements common to FixedLength & VariableLength are
propagated to Table. -->

```

```

<!-- Supported file formats -->
<!ELEMENT ASCIIFixedLength ((Length | Delimiter)?,
((FixedPrimaryKey+, FixedColumn*) | (FixedPrimaryKey*,
FixedColumn+)), ForeignKey*)>
<!ELEMENT FixedColumn (Name, Description?, (Numeric |
AlphaNumeric | Date)?, Map*, Range)>
<!ELEMENT FixedPrimaryKey (Name, Description?, (Numeric
| AlphaNumeric | Date)?, Map*, Range)>
<!ELEMENT CSV (NamesFirstRow?, Delimiter?,
TextEncapsulator?, ((PrimaryKey+, Column*) |
(PrimaryKey*, Column+)), ForeignKey*)>
<!ELEMENT Column (Name, Description?, (Numeric |
AlphaNumeric | Date)?, Map*)>
<!ELEMENT PrimaryKey (Name, Description?, (Numeric |
AlphaNumeric | Date)?, Map*)>

<!-- Description of the entity supplying the data.
(Optional) -->
<!ELEMENT DataSupplier (Name, Location, Comment)>

<!-- The first Media will contain index.xml. Importing
will process each media listed -->
<!ELEMENT Media (Name, Command*, Table+, Command*)>

<!-- Elements common to ASCIIFIXEDLENGTH & CSV are
propagated to Table.
CSV is the default format if none is specified. -->

```

```

<!ELEMENT Table (URL, Name?, Description?, Validity?,
(DecimalSymbol, DigitGroupingSymbol)?, SkipNumBytes?,
Range?, (CSV | ASCIIIFixedLength)?)>

<!-- ForeignKeys denote joins and relationships between
tables.
      Not specifying ForeignKeys prevent joins from
      occurring. -->

<!ELEMENT ForeignKey (Name+, References)>

<!-- Maps data from 'From' to 'To'
      ie. From      To
      =====
      True         1
      True         -1
      False        0

      A map is basically an associative container.
      Take caution that 'From' is of the same type
      as the column datatype.

-->

<!ELEMENT Map (Description?, From, To)>

<!-- Documentation for table validity. -->
<!ELEMENT Validity (Range, Format?)>

<!-- End Complex Types -->

```

```

<!ELEMENT Table (URL, Name?, Description?, Validity?, (ANSI |
Macintosh | OEM | UTF16 | UTF7 | UTF8)?, (DecimalSymbol,
DigitGroupingSymbol)?, SkipNumBytes?, Range?, Epoch?,
(VariableLength | FixedLength))>

<!-- ForeignKeys denote joins or relationships between
tables. To successfully join two tables make sure both the
PrimaryKey and the referenced column (foreignkey) are of the
same datatype.

Results are undefined when joining two tables with different
key datatypes. Most likely an error will occur. -->

<!ELEMENT ForeignKey (Name+, References)>

<!-- Maps AlphaNumeric columns from 'From' to 'To'
      ie. From      To
      =====
      True         1
      True         -1
      False        0

      Basically, a map is an associative container.
      The standard implementation only supports
      AlphaNumeric datatypes. The following
      conversions are NOT supported.

      Numeric      -> AlphaNumeric
      Date         -> AlphaNumeric
      AlphaNumeric -> Date
      AlphaNumeric -> Numeric

-->

<!ELEMENT Map (Description?, From, To)>

<!-- Documentation for table validity. -->
<!ELEMENT Validity (Range, Format?)>

<!-- End Complex Types -->

```