

Beschreibungsstandard für die Datenträger- überlassung

Erläuterungen zur Speicherung und Beschreibung
von Daten im Rahmen der GDPdU/GoBD

Version 1.6



Copyright
CaseWare Germany GmbH
Meitnerstr. 6
70563 Stuttgart
Deutschland

Bei der Abfassung dieses Leitfadens wurde mit größtmöglicher Sorgfalt und Gewissenhaftigkeit vorgegangen. Für Auslassungen oder Fehler können weder der Autor noch die CaseWare Germany GmbH eine Haftung oder juristische Verantwortung übernehmen.

Dieser Leitfaden ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen und Tabellen oder die Vervielfältigung auf anderen Wegen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind ohne die ausdrückliche schriftliche Genehmigung durch die CaseWare Germany GmbH nicht gestattet.

Aus Gründen der Lesbarkeit verwenden wir in unseren Texten die männliche Sprachform. Dies bedeutet jedoch keine Benachteiligung des weiblichen/dritten Geschlechts, sondern soll im Sinne der sprachlichen Vereinfachung als geschlechtsneutral zu verstehen sein.

IDEA ist ein eingetragenes Warenzeichen von CaseWare International Inc.

Beschreibungsstandard-Version 1.6 vom 1. März 2019

Änderung	Datum	Autor(en)
Beschreibungsstandard-Version 1.0	06.06.2002	Jon Chick
Beschreibungsstandard-Version 1.1	01.08.2002	Jon Chick
Beschreibungsstandard-Version 1.2	01.06.2003	Jon Chick
Beschreibungsstandard-Version 1.3	01.11.2003	Jon Chick
Beschreibungsstandard-Version 1.4	01.05.2004	Jon Chick
Beschreibungsstandard-Version 1.5	01.10.2004	Jon Chick
Beschreibungsstandard-Version 1.6	01.03.2019	Radu Furcea

*Tabelle 1: Änderungshistorie Beschreibungsstandard***Dokument-Version 1.6 vom 1. März 2019**

Änderung	Datum	Autor(en)
Dokument-Version 1.0	06.06.2002	Jon Chick, Tankred Giese, Martin Otto
Dokument-Version 1.1 Erläuterungen zu References Erweiterungen Version 1.1	01.08.2002	Martin Otto, Jon Chick, Tankred Giese, Michael Schleupen
Dokument-Version 1.2 Erläuterungen zu Alias Erweiterungen Version 1.2	01.06.2003	Martin Otto, Jon Chick, Tankred Giese, Michael Schleupen
Dokument-Version 1.3 Erläuterungen zu Extension Erweiterungen Version 1.3	01.11.2003	Martin Otto, Jon Chick, Tankred Giese, Michael Schleupen
Dokument-Version 1.4 Erläuterungen zu AcceptNoTables Erweiterungen Version 1.4	01.05.2004	Martin Otto, Jon Chick, Tankred Giese, Michael Schleupen
Dokument-Version 1.5 Erläuterungen zu mehreren Alias-Elementen Erweiterungen Version 1.5	01.10.2004	Martin Otto, Jon Chick, Tankred Giese, Michael Schleupen
Dokument-Version 1.6 Erläuterungen zum Datentyp Zeit Erweiterungen Version 1.6	01.03.2019	Radu Furcea, Tankred Giese, Michael Schleupen

Tabelle 2: Änderungshistorie Dokument

Inhalt

Adressaten dieses Papiers	5
Zusammenfassung/Ziel	5
Text von § 147 Abs. 6 AO	6
Datenträgerüberlassung	6
Die Datenorganisation bei der Datenträgerüberlassung	8
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	17
Hinweise zu XML	21
Beschreibung der Elemente.....	22
Beispiel 1 für eine Datei index.xml	40
Beispiel 2 für Verknüpfungen	43
Beispiel 3 für eine Datei index.xml (komprimierte Daten)	46
Beispiel 4 für eine Datei index.xml (Datentyp Zeit).....	47
Häufig gestellte Fragen	50
Synopse Beschreibungsstandard Version 1.0 und Version 1.1.....	53
Synopse Beschreibungsstandard Version 1.1 und Version 1.6.....	59
Abbildungsverzeichnis.....	69
Tabellenverzeichnis.....	70

Adressaten dieses Papiers

- Anwender von Software, die zur Bereitstellung eines Datenträgers/Datenpakets nach GDPdU/GoBD aufgefordert werden.
- Entwickler von Software, die steuerrelevante Daten für die Datenträgerüberlassung nach GDPdU/GoBD 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 von dem steuerpflichtigen Unternehmen (oder dem beauftragten Steuerberater, buchführenden [Sub-]Unternehmen etc.) in „maschinell auswertbarer Form“ auf geeigneten Datenträgern bereitgestellt werden.

Unter dem Begriff der „maschinellen 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 Software-Industrie beschreibt.

Bei Rückfragen wenden Sie sich bitte per E-Mail an sales.de@caseware.com, oder informieren Sie sich unter <https://www.caseware.net>.

Text von § 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 werden dem Unternehmen die Prüfung angekündigt und die Prüfungsschwerpunkte bekanntgegeben. Der Steuerprüfer bittet das Unternehmen, die steuerlich relevanten Daten auf einem maschinell auswertbaren Datenträger nach den Vorgaben der „Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen“ (GDPdU) gemäß BMF-Schreiben vom 16.07.2001 bzw. der „Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff“ (GoBD) gemäß BMF-Schreiben vom 28.11.2019 bereitzustellen.

Bei Prüfungsbeginn erhält der Steuerprüfer einen Datenträger oder ein Datenpaket.

Gemäß seinem Prüfungsschwerpunkt sucht der Prüfer die entsprechenden Tabellen heraus und liest diese ein. Auf dem Datenträger/in dem Datenpaket sind steuerrelevante Daten und beschreibende Daten enthalten, um die Daten ohne weitere Erklärungen einlesen zu können. Sobald die Daten in die Analysesoftware importiert sind, führt der Steuerprüfer eine oder mehrere Analysen (teilweise durch Makros automatisiert) aus.



Abbildung 1: Steuerprüfer

1. Fordert Datenträger/Datenpaket an
2. Liest Datenträger/Datenpaket 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. den 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 feststehen. Die Daten werden anschließend automatisch oder manuell auf einem beweglichen Datenträger/in einem Datenpaket gespeichert.



Abbildung 2: Datenträgerüberlassung

Die Datenorganisation bei der Datenträgerüberlassung

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

Nachdem die Finanzverwaltung des Bundes und der Länder die in der Wirtschaftsprüfung verbreitete Analysesoftware IDEA beschafft hat, sind die Steuerprüfer in der Lage, Daten maschinell zu lesen und (automatisiert oder manuell gesteuert) zu verarbeiten. Die Finanzverwaltung definiert nicht die Datenfelder und Inhalte, die bei der Datenträgerüberlassung auf den Datenträger/in das Datenpaket gebracht werden. Entsprechend ist es notwendig, dass Steuerpflichtige und/oder Hersteller von Software, die steuerlich relevante 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 zu den Dateninhalten und Formaten 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 bzw. 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 – wird auf einem gemeinsamen Datenträger/in einem gemeinsamen Datenpaket bereitgestellt.

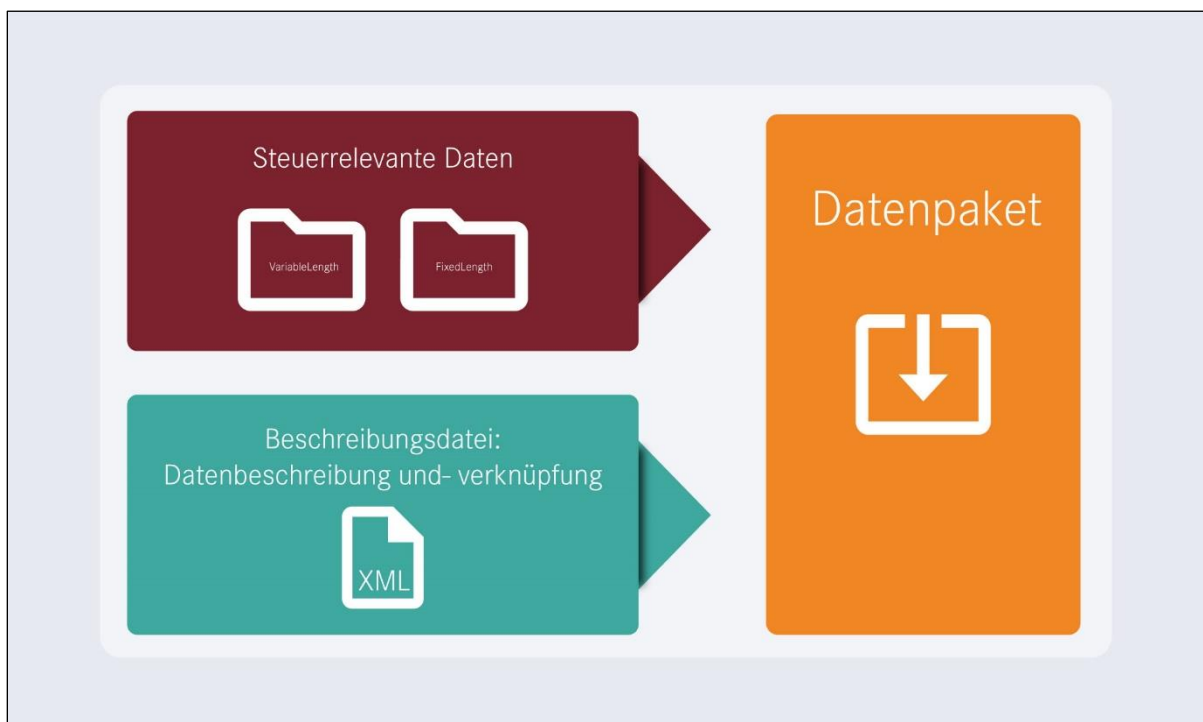


Abbildung 3: Datenorganisation

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 Beschreibungsstandards.

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 (validieren), ob ein XML-Dokument der vereinbarten Sprache entspricht. 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 Datei index.xml.

Weiterführende Informationen zu XML findet man auf <https://www.w3.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/in dem Datenpaket befinden wie die Datei index.xml.

Werden mehrere Datenträger (Medien) bereitgestellt, muss sich die Datei index.xml nur auf dem ersten Datenträger befinden.

Technischer Ablauf des Importprozesses

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

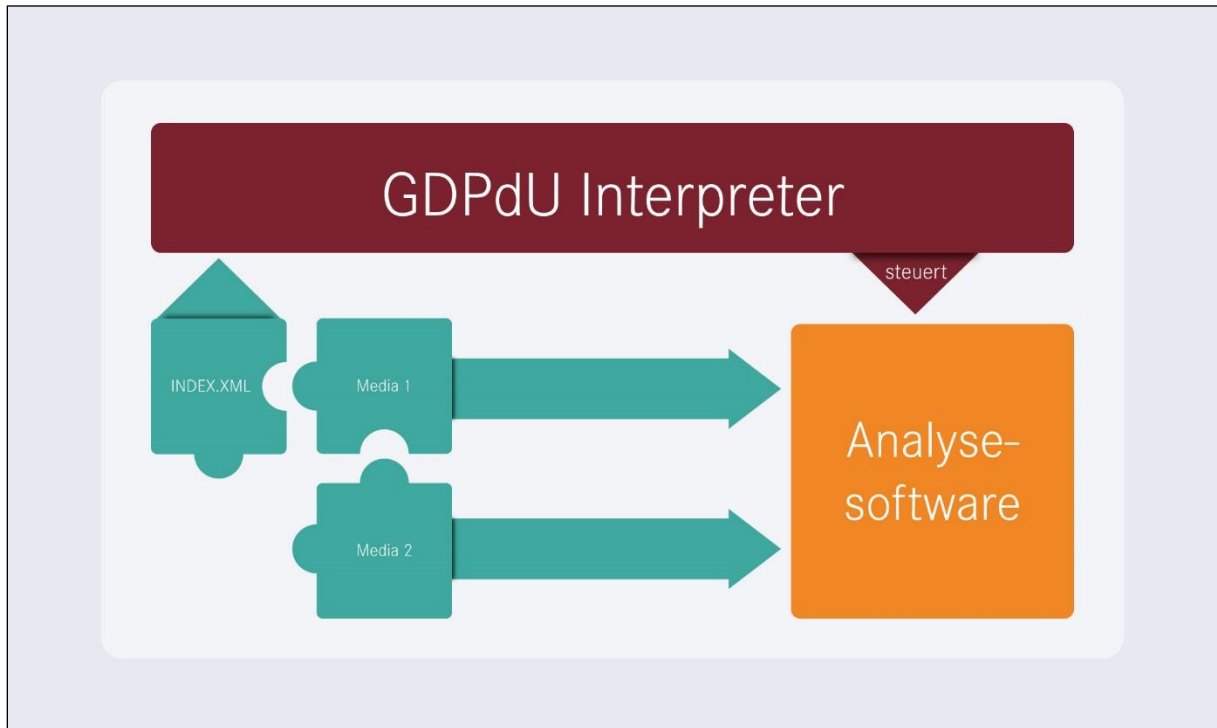


Abbildung 4: Importprozess

Die XML-DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Versions available:
1.1 (August-01-2002)

1.2 (June-01-2003)
    New Element Alias

1.3 (November-01-2003)
    New Element Extension

1.4 (May-01-2004)
    New Element AcceptNoTables
    <Table> is now optional

1.5 (September-01-2004)
    Updated the ForeignKey and Alias elements so they allow
    multiple Alias elements per ForeignKey. This allows one to use alias
    elements for a Composite key.

Example:

<ForeignKey>
  <Name>Order</Name>
  <Name>Customer</Name>
  <References>Orders</References>
  <Alias>
    <From>Order</From>
    <To>OrderId</To>
  </Alias>
  <Alias>
    <From>Customer</From>
    <To>CustomerId</To>
  </Alias>

1.6 (March-03-2019)
    Updated the Map and From/To elements descriptions so they allow
    importing an Alphanumeric column as Time column.
    This allows defining that an Alphanumeric column contains
    data of type Time and allows you to specify which is the time format used.

Example:

<Map>
  <From>HHMMSS</From>
  <To>HHMMSS</To>
</Map>
-->

<!-- 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)>

<!-- Sometimes, it is desirable to have a <Media> with
no tables.

Example:
An extension might provide metadata
that describes all tax-relevant data.

This option is turned off by default. -->
<!ELEMENT AcceptNoTables (#PCDATA)>

<!-- Use Alias to reference columns with different names in
a ForeignKey element. These Alias elements are optional.
```

The following rules apply to the Alias element:

- One Alias can be used per ForeignKey.
- Alias elements can appear in any order.

Example:

Table Orders has a primary key OrderId
Table Accounts has a foreign key Order.

You can use the Alias element to specify Order references OrderId.

```
<ForeignKey>
  <Name>Order</Name>
  <Name>Customer</Name>
  <References>Orders</References>
  <Alias>
    <From>Order</From>
    <To>OrderId</To>
  </Alias>
</ForeignKey> -->
```

<!ELEMENT Alias (From, To)>

<!-- 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 -->

<!-- Use Extension when you want to add application
    specific functionality to the existing standard.

    Name - the extension name or identifier.
    URL - the supplementary .xml file that corresponds to the
    extension.

    An application that extends the standard should scan the
    Dataset element for the presence of zero or more Extension
    elements. The application can use the Name element to identify
    the extension.

    When choosing a name for your extension, do not choose a common
    name. This will reduce undefined results for name conflicts.

    It is possible that future extensions will be ratified as
    mandatory to meet GDPdU guidelines. -->
<ELEMENT Extension (Name, URL)>

<!-- Self-explanatory -->
<ELEMENT Range (From, (To | Length)?)>
<ELEMENT FixedRange (From, (To | Length))>

<!-- The document element -->
<ELEMENT DataSet (Extension*, 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*, AcceptNoTables?)>

<!-- 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.

    see line 35 for information about Alias -->
<!ELEMENT ForeignKey (Name+, References, Alias*)>

<!-- 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

    In order to keep the backward compatibility and because the Time
    data type was not defined yet we allow specifying that an Alphanumeric
    column contains data of type Time like:

    <Map>
    <From>HHMMSS</From>
    <To>HHMMSS</To>
</Map>

    If the application which interprets the xml and the target application
    to import into are able to deal with time data type this column data could
    be imported as Time data type

    The value of the From element has to be equal with the value of the To
    element, in order to not be interpreted as an actual Alphanumeric data mapping
    in which the intention is to replace a certain text with a different text at

```

import time, and be a valid Time format.
The following formats are recognized as valid Time formats:
HH as symbols for hours
MM as symbols for minutes
SS as symbols for seconds
TT as symbols for AM/PM
SS and TT are optional when defining the format.
A time format separator can also be supplied if needed.

Examples of valid formats:
HHMM, HH:MM, HHMMSS, HHMMTT, HH:MMTT, HH:MM:SS, HHMSSTT,
HH:MM:SSTT, HHMM TT, HH:MM TT, HHMMSS TT, HH:MM:SS TT

-->

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

<!-- Documentation for table validity. -->

<!ELEMENT Validity (Range, Format?)>

<!-- End Complex Types -->

Abbildung 5: XML-DTD

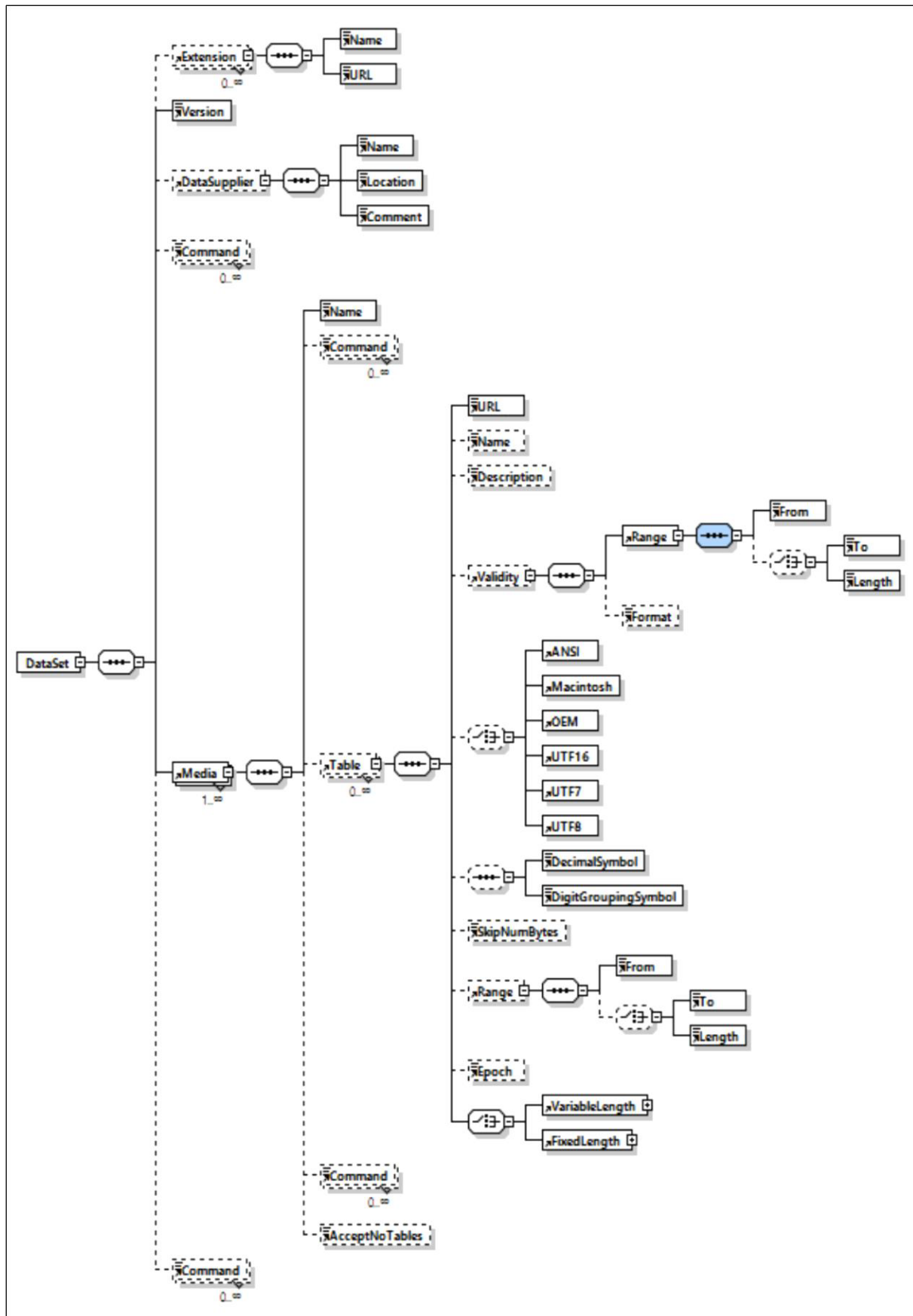


Abbildung 7: Beschreibungsstandard – DataSet

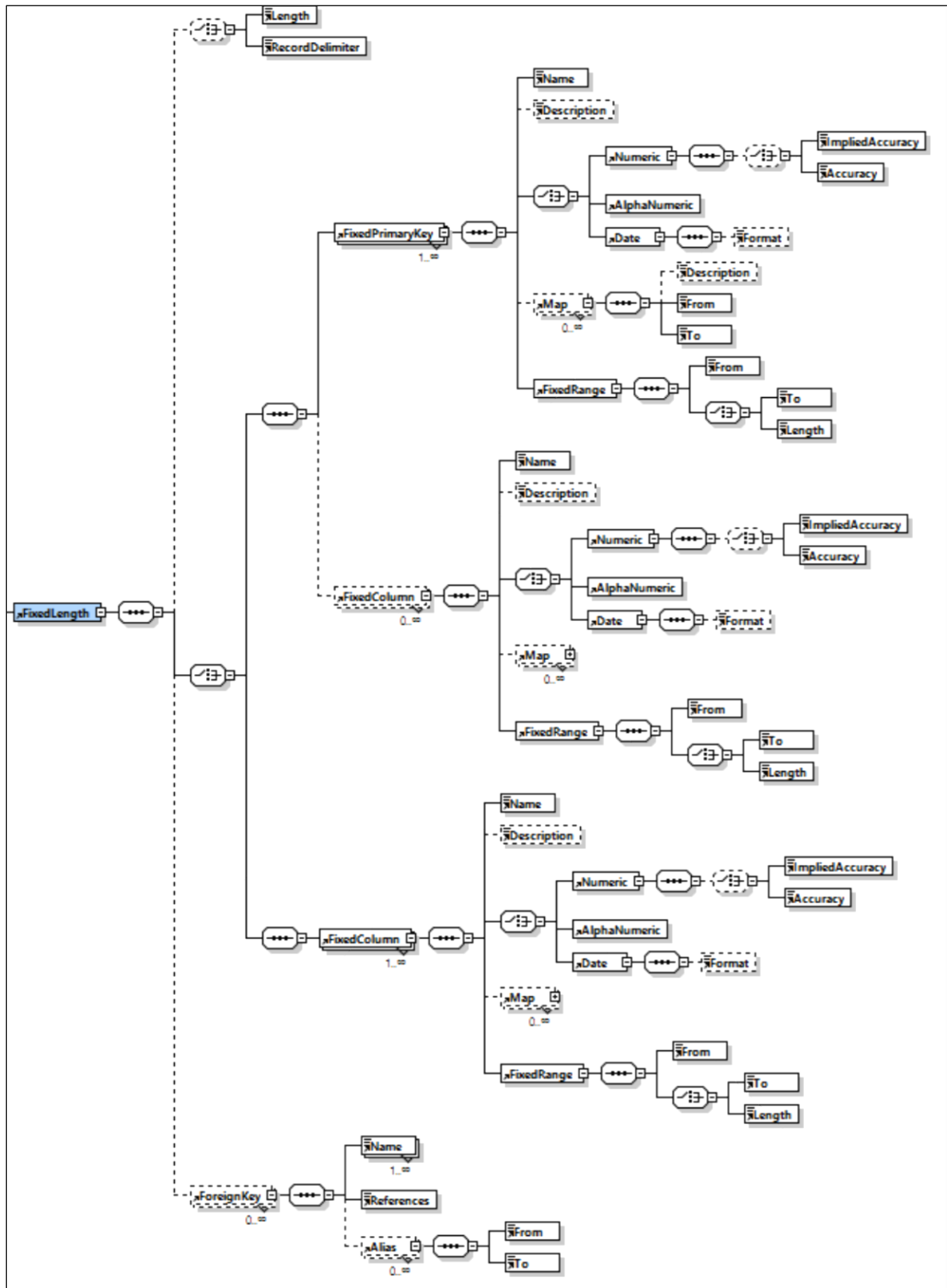


Abbildung 8: Beschreibungsstandard – FixedLength

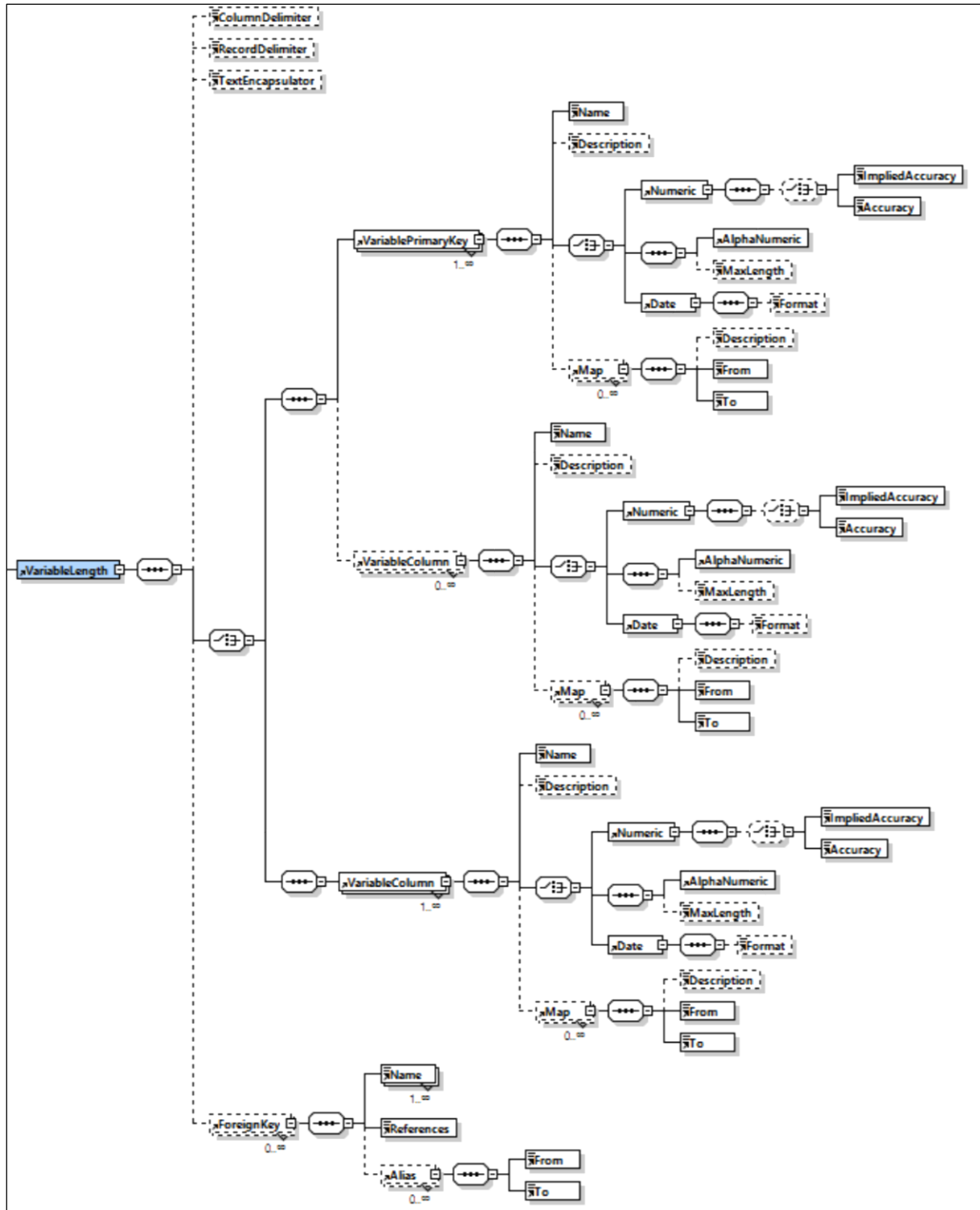


Abbildung 9: Beschreibungsstandard – VariableLength

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

Tabelle 3: Übersicht über die gängigen Operatoren

- #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 keines der folgenden Zeichen: " & < >

Verwenden Sie stattdessen den Namen oder die Nummer aus der untenstehenden Liste.

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

Tabelle 4: Liste der Namen und Nummern für die Zeichen " & < >

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	Kontextabhängige Beschreibung
DataSet	

Tabelle 5: ELEMENT Version

<!ELEMENT Location (#PCDATA)>

Beschreibt den Standort des Datenlieferanten.

Verwendet in Container	Kontextabhängige Beschreibung
DataSupplier	

Tabelle 6: ELEMENT Location

<!ELEMENT Comment (#PCDATA)>

Kommentarfeld für zusätzliche Informationen zum Datenlieferanten.

Verwendet in Container	Kontextabhängige Beschreibung
DataSupplier	

Tabelle 7: ELEMENT Comment

<!ELEMENT Length (#PCDATA)>

Enthält Informationen über die Länge von Attributen oder Datensätzen.

Verwendet in Container	Kontextabhängige Beschreibung
Range	

Tabelle 8: ELEMENT Length

<!ELEMENT References (#PCDATA)>

Enthält Informationen über Verknüpfungen.

Verwendet in Container	Kontextabhängige Beschreibung
ForeignKey	referenzierte Tabelle

Tabelle 9: ELEMENT References

Siehe hierzu auch das Beispiel auf Seite 43.

<!ELEMENT From (#PCDATA)>

Enthält Informationen über den Startwert eines Wertebereichs.

Verwendet in Container	Kontextabhängige Beschreibung
Range	Startwert in einem Wertebereich
Map	Redefinition von Daten
Alias	Name der Spalte der aktuellen Tabelle, die in einer Fremdschlüsseldefinition verwendet wird

Tabelle 10: ELEMENT From

<!ELEMENT To (#PCDATA)>

Enthält Informationen über den Endwert eines Wertebereichs.

Verwendet in Container	Kontextabhängige Beschreibung
Range	Endwert in einem Wertebereich
Map	Redefinition von Daten
Alias	Der entsprechende Spaltenname der referenzierten Tabelle für einen Fremdschlüssel

Tabelle 11: ELEMENT To

<!ELEMENT MaxLength (#PCDATA)>

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

Verwendet in Container	Kontextabhängige Beschreibung
VariableColumn	Maximale Länge des Feldes
VariablePrimaryKey	Maximale Länge des Feldes

Tabelle 12: ELEMENT MaxLength

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 werden.

<!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	Kontextabhängige Beschreibung
VariableLength	

Tabelle 13: ELEMENT TextEncapsulator

<!ELEMENT Accuracy (#PCDATA)>

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

Vorgabewert: 0

Verwendet in Container	Kontextabhängige Beschreibung
Numeric	

Tabelle 14: ELEMENT Accuracy

<!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	Kontextabhängige Beschreibung
Numeric	

Tabelle 15: ELEMENT ImpliedAccuracy

<ELEMENT Format (#PCDATA)>

Angabe über Formatdefinitionen.

Vorgabewert: DD.MM.YYYY

Verwendet in Container	Kontextabhängige Beschreibung
Date	Hier wird die Maskierung für die Darstellung von Datumsangaben hinterlegt. Folgende Angaben für Format sind gültig: • DD für Tag, • MM für Monat, • YY bzw. YYYY für Jahr. Beispiel (30. September 2001) • 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)

Tabelle 16: ELEMENT Format

<ELEMENT DecimalSymbol (#PCDATA)>

Angabe über das Dezimalstellen-Trennzeichen.

Vorgabewert: ,

Verwendet in Container	Kontextabhängige Beschreibung
Table	

Tabelle 17: ELEMENT DecimalSymbol

<ELEMENT DigitGroupingSymbol (#PCDATA)>

Definiert das Tausender-Trennzeichen.

Vorgabewert: .

Verwendet in Container	Kontextabhängige Beschreibung
Table	

Tabelle 18: ELEMENT DigitGroupingSymbol

<!ELEMENT Command (#PCDATA)>

Definiert ein Betriebssystemkommando. Commands 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	Kontextabhängige Beschreibung
DataSet	Command für den gesamten Vorgang
Media	Command für einen einzelnen Datenträger

Tabelle 19: ELEMENT Command

Siehe hierzu auch das Beispiel auf Seite 46.

<!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 Datei index.xml.

Beispiele für **ungültige** Einträge:

http://www.somewhere.com/data/Accounts.csv
ftp://ftp.somewhere.com/data/Accounts.csv

Beispiele für **gültige** Einträge:

Accounts.dat
data/Accounts.dat
data/january/Accounts.dat
../Accounts.dat

Verwendet in Container	Kontextabhängige Beschreibung
Table	

Tabelle 20: ELEMENT URL

<!ELEMENT Description (#PCDATA)>

Beschreibungsfeld

Verwendet in Container	Kontextabhängige 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

Tabelle 21: ELEMENT Description

<!ELEMENT Name (#PCDATA)>

Beschreibungsfeld

Vorgabewert: Angabe aus dem URL-Element

Verwendet in Container	Kontextabhängige 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 Tabellennamen
ForeignKey	Name des Fremdschlüssels

Tabelle 22: ELEMENT Name

<!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 oder 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	Kontextabhängige Beschreibung
Table	

Tabelle 23: ELEMENT Epoch

<!ELEMENT ColumnDelimiter(#PCDATA)>

Definiert Trennzeichen für Datenfelder.

Vorgabewert: ; (Semikolon „;“)

Verwendet in Container	Kontextabhängige Beschreibung
VariableLength	Datenfeld-Trennzeichen

Tabelle 24: ELEMENT ColumnDelimiter

Beispiel für übliche Trennzeichen:

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

Tabelle 25: ELEMENT ColumnDelimiter – Liste gängiger Trennzeichen

<ELEMENT RecordDelimiter (#PCDATA)>

Definiert Trennzeichen für Datensätze.

Vorgabewert: 
 (CRLF)

Verwendet in Container	Kontextabhängige Beschreibung
FixedLength	Datensatz-Trennzeichen
VariableLength	Datensatz-Trennzeichen

Tabelle 26: ELEMENT RecordDelimiter

Beispiel für übliche Trennzeichen:

Trennzeichen	XML-Notation
CR Carriage Return	
CRLF	

LF Linefeed	

Tabelle 27: ELEMENT RecordDelimiter – Liste gängiger Trennzeichen

<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	Kontextabhängige Beschreibung
Table	

Tabelle 28: ELEMENT SkipNumBytes

<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	Kontextabhängige Beschreibung
Table	Startdatensatz
Validity	Gültigkeitsbereich

Tabelle 29: ELEMENT Range

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

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

Verwendet in Container	Kontextabhängige Beschreibung
FixedColumn	Feldposition in FixedLength-Dateien
FixedPrimaryKey	Schlüsselposition in FixedLength-Dateien

Tabelle 30: ELEMENT FixedRange

<!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	Kontextabhängige 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

Tabelle 31: ELEMENT AlphaNumeric

<!ELEMENT Date (Format?)>

Definiert den Datentyp als Datum.

Verwendet in Container	Kontextabhängige 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

Tabelle 32: ELEMENT Date

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

Definiert den Datentyp als numerisch.

Verwendet in Container	Kontextabhängige 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

Tabelle 33: ELEMENT Numeric

<!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 Grafikzeichen. 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-)Grafikzeichen 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 Mac	Mac-spezifischer 8-Bit-Zeichensatz „Mac-ASCII“, der bis 127 ASCII entspricht und oberhalb 127 nationale Sonderzeichen enthält. Zeilentrenner ist CR.

Tabelle 34: ELEMENT UTF8

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	Kontextabhängige Beschreibung
Table	

Tabelle 35: ELEMENT FixedLength

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

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

Verwendet in Container	Kontextabhängige Beschreibung
FixedLength	

Tabelle 36: ELEMENT FixedColumn

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

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

Verwendet in Container	Kontextabhängige Beschreibung
FixedLength	

Tabelle 37: ELEMENT FixedPrimaryKey

Siehe hierzu auch das Beispiel auf Seite 43.

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

Definiert eine Datei vom Typ VariableLength.

Verwendet in Container	Kontextabhängige Beschreibung
Table	

Tabelle 38: ELEMENT VariableLength

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	Kontextabhängige Beschreibung
VariableLength	

Tabelle 39: ELEMENT VariableColumn

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	Kontextabhängige Beschreibung
VariableLength	

Tabelle 40: ELEMENT VariablePrimaryKey

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 43.

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

Enthält Angaben zum Datenlieferanten.

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

Tabelle 41: ELEMENT DataSupplier

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

Definiert den Inhalt eines Datenträgers.

Verwendet in Container	Kontextabhängige Beschreibung
DataSet	Datenträgerüberlassung

Tabelle 42: ELEMENT Media

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

Definiert eine Tabelle.

Verwendet in Container	Kontextabhängige Beschreibung
Media	Tabelle

Tabelle 43: ELEMENT Table

<!ELEMENT ForeignKey (Name+, References, Alias*)>

Definiert eine Spalte als Fremdschlüssel.

Die Spalte, auf die mit Namen Bezug genommen wird, muss in der gleichen Tabelle als FixedPrimaryKey/FixedColumn oder VariablePrimaryKey/VariableColumn definiert sein. References ist der Name der referenzierten Tabelle und Alias wird verwendet, wenn die referenzierte Spalte in der referenzierten Tabelle einen anderen Namen hat, siehe Definition des Elements Alias für weitere Informationen.

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

Tabelle 44: ELEMENT ForeignKey

Siehe hierzu auch das Beispiel auf Seite 43.

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

Mit dem Element Map können Redefinitionen auf Datenebene vorgenommen werden.

Beispiel:

From (Ausgangsdaten)	To (Zielfdaten)
1	wahr
0	falsch
Hr	Herr
Fr	Frau
01	verheiratet
02	ledig
03	Steuerklasse 3

Tabelle 45: ELEMENT Map – Redefinitionen

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

Tabelle 46: ELEMENT Map

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.

Mit Map können Sie auch angeben, dass eine alphanumerische Spalte Daten vom Typ Zeit enthält:

```
<Map>
  <From>HHMMSS</From>
  <To>HHMMSS</To>
</Map>
```

Wenn die Anwendung, die die XML-Datei interpretiert, und die Zielanwendung, in die importiert werden soll, mit dem Datentyp Zeit umgehen können, können die Daten dieser Spalte als Datentyp Zeit importiert werden.

Der Wert des From-Elements muss gleich dem Wert des To-Elements sein, damit er nicht als alphanumerisches Daten-Mapping interpretiert wird, bei dem ein bestimmter Text beim Import durch einen anderen Text ersetzt werden soll, und ein gültiges Time-Format ist.

Die folgenden Formate werden als gültige Zeitformate anerkannt:

HH als Symbole für Stunden

MM als Symbole für Minuten

SS als Symbole für Sekunden

TT als Symbole für AM/PM

SS und TT sind bei der Definition des Formats optional.

Bei Bedarf kann auch ein Zeitformattrennzeichen angegeben werden.

Beispiele für gültige Formate:

HHMM, HH:MM, HHMMSS, HHMMTT, HH:MMTT, HH:MM:SS, HHMMSSTT,

HH:MM:SSTT, HHMM TT, HH:MM TT, HHMMSS TT, HH:MM:SS TT

<!ELEMENT Validity (Range, Format?)>

Definiert den Gültigkeitszeitraum der Daten.

Verwendet in Container	Kontextabhängige Beschreibung
Table	Gültigkeitszeitraum

Tabelle 47: ELEMENT Validity

<!ELEMENT AcceptNoTables (#PCDATA)>

Legt fest, dass ein <Media> keine Tabellen hat.

Verwendet in Container	Kontextabhängige Beschreibung
Media	Media hat keine Tabellen

Tabelle 48: ELEMENT AcceptNoTables

Manchmal ist es wünschenswert, ein <Media> ohne Tabellen zu haben.

Beispiel:

Eine Erweiterung könnte Metadaten bereitstellen, die alle steuerrelevanten Daten beschreiben.

<!ELEMENT Alias (From, To)>

Definiert das entsprechende Spaltennamenpaar bei der Definition einer Fremdschlüsselreferenz.

Verwendet in Container	Kontextabhängige Beschreibung
ForeignKey	Für einen Spaltennamen den Namen der Spalte angeben, auf die er in der referenzierten Tabelle verweist

Tabelle 49: ELEMENT Alias

Verwenden Sie Alias, um auf Spalten mit unterschiedlichen Namen in einem ForeignKey-Element zu verweisen. Diese Alias-Elemente sind optional.
Die folgenden Regeln gelten für das Alias-Element:

- Ein oder mehrere Alias-Elemente können pro ForeignKey verwendet werden.
- Alias-Elemente können in beliebiger Reihenfolge erscheinen.

Beispiel:

Die Tabelle Orders hat einen Primärschlüssel OrderId
Die Tabelle Accounts hat einen Fremdschlüssel Order.

Sie können das Alias-Element verwenden, um anzugeben, dass Order auf OrderId verweist.

```
<ForeignKey>
  <Name>Order</Name>
  <Name>Customer</Name>
  <References>Orders</References>
  <Alias>
    <From>Order</From>
    <To>OrderId</To>
  </Alias>
</ForeignKey>
```

<!ELEMENT Extension (Name, URL)>

Ermöglicht es Ihnen, den bestehenden Standard um anwendungsspezifische Funktionen zu erweitern.

Verwendet in Container	Kontextabhängige Beschreibung
DataSet	

Tabelle 50: ELEMENT Extension

Verwenden Sie Extension, wenn Sie dem bestehenden Standard anwendungsspezifische Funktionen hinzufügen möchten.

Name – Name oder Kennung der Erweiterung.

URL – zusätzliche XML-Datei, die der Erweiterung entspricht.

Eine Anwendung, die den Standard erweitert, sollte das Element Dataset auf das Vorhandensein von null oder mehr Extension-Elementen überprüfen. Die Anwendung kann das Element Name verwenden, um die Erweiterung zu identifizieren.

Wenn Sie einen Namen für Ihre Erweiterung wählen, sollten Sie keinen allgemeinen Namen wählen. Dadurch werden undefinierte Ergebnisse für Namenskonflikte vermieden.

Es ist möglich, dass zukünftige Erweiterungen als obligatorisch eingestuft werden, um die GDPdU-/GoBD-Richtlinien zu erfüllen.

Beispiel 1 für eine Datei 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>
<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>

```

Abbildung 10: Beispiel 1 – Datei index.xml

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:

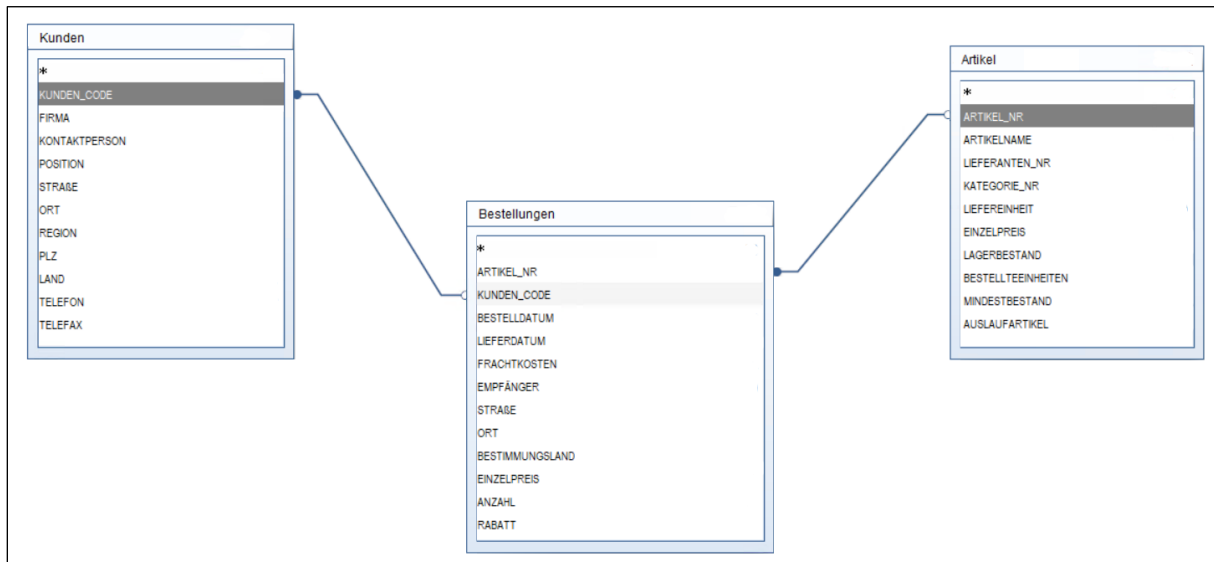


Abbildung 11: Beispiel 2 – Tabellen mit Bewegungsdaten und Stammdaten

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>
---	--

Abbildung 12: Beispiel 2 – Stammdaten

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>
```

Abbildung 13: Beispiel 2 – Bewegungsdaten

Bitte beachten Sie:

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.

```
<ForeignKey>
  <Name>FIRM</Name>
  <Name>DDKZ</Name>
  <Name>KKKZ</Name>
  <Name>UKTO</Name>
  <Name>BUJM</Name>
  <References>FPCPDPU</References>
  <Alias>
    <From>UKTO</From>
    <To>CKTO</To>
  </Alias>
  <Alias>
    <From>BUJM</From>
    <To>GAJM</To>
  </Alias>
</ForeignKey>
```

Abbildung 14: Beispiel 2 – Beispiel eines Fremdschlüssels mit mehreren Aliassen

Beispiel 3 für eine Datei 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>
```

Abbildung 15: Beispiel 3 – Datei index.xml

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
```

Abbildung 16: Beispiel 3 – Datei uncompress.bat

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

Die Datei gdpdu.zip enthält die gepackte Datenträgerbereitstellung (einschließlich der Datei index.xml und gdpdu-01-08-2002.dtd). Der Prüfer öffnet nach dem Dekomprimieren also die neue Datenbereitstellung in dem Verzeichnis %TEMP%\gdpdu.

Bitte beachten Sie:

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

Beispiel 4 für eine Datei index.xml (Datentyp Zeit)

index.xml

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE DataSet SYSTEM "gdpdu-01-03-2019.dtd">
<DataSet>
  <Version>1.0.00</Version>
  <DataSupplier>
    <Name />
    <Location />
    <Comment>SmartExporter</Comment>
  </DataSupplier>
  <Media>
    <Name>Disk1</Name>
    <Table>
      <URL>20181120175609__BKPF.csv</URL>
      <Name>BKPF</Name>
      <Description>
      </Description>
      <UTF8 />
      <DecimalSymbol>.</DecimalSymbol>
      <DigitGroupingSymbol>,</DigitGroupingSymbol>
      <VariableLength>
        <ColumnDelimiter> </ColumnDelimiter>
        <TextEncapsulator />
        <VariableColumn>
          <Name>BUKRS</Name>
          <Description>Company Code</Description>
          <AlphaNumeric />
          <MaxLength>4</MaxLength>
        </VariableColumn>
        <VariableColumn>
          <Name>BELNR</Name>
          <Description>Accounting Document Number</Description>
          <AlphaNumeric />
          <MaxLength>10</MaxLength>
        </VariableColumn>
        <VariableColumn>
          <Name>BLDAT</Name>
          <Description>Document Date in Document</Description>
          <Date>
            <Format>YYYYMMDD</Format>
          </Date>
        </VariableColumn>
        <VariableColumn>
          <Name>BUDAT</Name>
          <Description>Posting Date in the Document</Description>
          <Date>
            <Format>YYYYMMDD</Format>
          </Date>
        </VariableColumn>
        <VariableColumn>
          <Name>CPUTM</Name>
          <Description>Time of Entry</Description>
          <AlphaNumeric />
          <MaxLength>6</MaxLength>
        </VariableColumn>
      <Map>
```

```

        <From>HH:MM</From>
        <To>HH:MM</To>
    </Map>
</VariableColumn>
<VariableColumn>
    <Name>CPUTM1</Name>
    <Description>Time of Entry</Description>
    <AlphaNumeric />
    <MaxLength>7</MaxLength>
    <Map>
        <From>HH:MMTT</From>
        <To>HH:MMTT</To>
    </Map>
</VariableColumn>
<VariableColumn>
    <Name>CPUTM2</Name>
    <Description>Time of Entry</Description>
    <AlphaNumeric />
    <MaxLength>9</MaxLength>
    <Map>
        <From>HH:MM:SSTT</From>
        <To>HH:MM:SSTT</To>
    </Map>
</VariableColumn>
<VariableColumn>
    <Name>CPUTM3</Name>
    <Description>Time of Entry</Description>
    <AlphaNumeric />
    <MaxLength>6</MaxLength>
    <Map>
        <From>HHMM</From>
        <To>HHMM</To>
    </Map>
</VariableColumn>
<VariableColumn>
    <Name>CPUTM4</Name>
    <Description>Time of Entry</Description>
    <AlphaNumeric />
    <MaxLength>7</MaxLength>
    <Map>
        <From>HHMMTT</From>
        <To>HHMMTT</To>
    </Map>
</VariableColumn>
<VariableColumn>
    <Name>CPUTM5</Name>
    <Description>Time of Entry</Description>
    <AlphaNumeric />
    <MaxLength>9</MaxLength>
    <Map>
        <From>HHMMSS</From>
        <To>HHMMSS</To>
    </Map>
</VariableColumn>
<VariableColumn>
    <Name>CPUTM6</Name>
    <Description>Time of Entry</Description>
    <AlphaNumeric />
    <MaxLength>9</MaxLength>
    <Map>

```



```

        <From>HHMMSSTT</From>
        <To>HHMMSSTT</To>
    </Map>
</VariableColumn>
<VariableColumn>
    <Name>CPUTM7</Name>
    <Description>Time of Entry</Description>
    <AlphaNumeric />
    <MaxLength>9</MaxLength>
    <Map>
        <From>HH:MM:SS</From>
        <To>HH:MM:SS</To>
    </Map>
</VariableColumn>
<VariableColumn>
    <Name>CPUTM8</Name>
    <Description>Time of Entry</Description>
    <AlphaNumeric />
    <MaxLength>10</MaxLength>
    <Map>
        <From>HHMMSS</From>
        <To>HHMMSS</To>
    </Map>
</VariableColumn>
</VariableLength>
</Table>
</Media>
</DataSet>

```

Abbildung 17: Beispiel 4 – Datei index.xml mit Datentyp Zeit

Häufig gestellte Fragen

- **Müssen wir die Daten im XML-Format liefern?**
Nein. Nur die Beschreibung der Daten (Datei 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 (<https://xerces.apache.org/xerces-c/>).
- **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 Datei 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/Datenpakets von Ihrem System eingelesen werden können?**

Die sichere Variante der Prüfung ist der Import mit IDEA für GDPdU/GoBD (IDEA + SmartX). SmartV hilft Ihnen, die Beschreibungsstandard-Konformität der Datei index.xml und der zu importierenden Daten zu überprüfen.

Eine einfache Möglichkeit, die XML-Datei auf ihre syntaktische Korrektheit zu testen, ist, sie mit einem XML-fähigen Browser zu öffnen.

- **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 (in der Datei index.xml) sollte immer von der Stelle aus geschehen, die die Datenhoheit besitzt, also vom Softwarehersteller oder steuerpflichtigen Unternehmen.

Audicon unterstützt Sie bei der Überprüfung, ob die Datenträgerüberlassung mit dem Beschreibungsstandard übereinstimmt, und bei der Erstellung der Datei index.xml (siehe <https://www.caseware.net/consulting/dienstleistungspakete/>).

- **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 Datei gdpdu-01-08-2002.dtd mit auf dem Datenträger gespeichert werden?**

Ja. Ein Datenträger, der zur Datenträgerüberlassung dient, muss die Datei gdpdu-01-08-2002.dtd, die Datei 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>
```

Abbildung 18: Range-Angabe

- 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 1.0 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 --> <!ELEMENT TextEncapsulator (#PCDATA)> <!-- Specifies how many digits appear to the right of the decimal symbol. --> </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) -- > <!ELEMENT TextEncapsulator (#PCDATA)> <!-- Specifies how many digits appear to the right of the decimal symbol. </pre>

Version 1.0 vom 10. Mai 2002 (6.6.2002)	Version 1.1 vom 1. August 2002
<pre> <!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)> <!-- The first line in CSV URL contains column headers. 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. --> <!ELEMENT NamesFirstRow EMPTY> <!-- Enables you to change how gdpdu displays dates. The default is implementation dependant. --> <!ELEMENT Format (#PCDATA)> <!-- Specifies the symbol that indicates decimal values. The default is implementation dependant. Specified once per Table. --> <!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 </pre>	<pre> 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 </pre>

Version 1.0 vom 10. Mai 2002 (6.6.2002)	Version 1.1 vom 1. August 2002
<pre> * 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)> <!-- The logical names of specified element --> <!ELEMENT Name (#PCDATA)> <!-- Y2K Window Any year before Epoch is 2000+ Default value 50. --> </pre>	<pre> * 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. --> </pre>

Version 1.0 vom 10. Mai 2002 (6.6.2002)	Version 1.1 vom 1. August 2002
<pre> <!-- ELEMENT Epoch (#PCDATA)> <!-- Specifies the symbol that separates 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> <!-- 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> </pre>

Version 1.0 vom 10. Mai 2002 (6.6.2002)	Version 1.1 vom 1. August 2002
<pre> <!-- 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 ASCIIIFIXEDLENGTH & 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 ASCIIFixedLength)?)> <!-- ForeignKeys denote joins and relationships between tables. Not specifying ForeignKeys prevent joins from occurring. --> </pre>	<pre> <!-- 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. --> </pre>

Version 1.0 vom 10. Mai 2002 (6.6.2002)	Version 1.1 vom 1. August 2002
<pre> <!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 --> </pre>	<pre> <!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 --> </pre>

Tabelle 51: Gegenüberstellung Beschreibungsstandard Version 1.0 und 1.1

Synopse Beschreibungsstandard Version 1.1 und Version 1.6

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

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
<pre><?xml version="1.0" encoding="UTF-8"?> <!--Versions available: 1.1 (August-01-2002) --></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <!--Versions available: 1.1 (August-01-2002) 1.2 (June-01-2003) New Element Alias 1.3 (November-01-2003) New Element Extension 1.4 (May-01-2004) New Element AcceptNoTables <Table> is now optional 1.5 (September-01-2004) Updated the ForeignKey and Alias elements so they allow multiple Alias elements per ForeignKey. This allows one to use alias elements for a Composite key. Example: <ForeignKey> <Name>Order</Name> <Name>Customer</Name> <References>Orders</References> <Alias> <From>Order</From> <To>OrderId</To> </Alias> <Alias> <From>Customer</From></pre>

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
<pre> <!-- 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)> </pre>	<pre> <To>CustomerId</To> </Alias> 1.6 (March-03-2019) Updated the Map and From/To elements descriptions so they allow importing an Alphanumeric column as Time column. This allows defining that an Alphanumeric column contains data of type Time and allows you to specify which is the time format used. Example: <Map> <From>HHMMSS</From> <To>HHMMSS</To> </Map> --> <!-- 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)> <!-- Sometimes, it is desirable to have a <Media> with no tables. Example: An extension might provide metadata that describes all tax-relevant data. This option is turned off by default. --> <!ELEMENT AcceptNoTables (#PCDATA)> </pre>

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
<pre> <!-- 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 "</pre>	<pre> <!-- Use Alias to reference columns with different names in a ForeignKey element. These Alias elements are optional. The following rules apply to the Alias element: - One Alias can be used per ForeignKey. - Alias elements can appear in any order. Example: Table Orders has a primary key OrderId Table Accounts has a foreign key Order. You can use the Alias element to specify Order references OrderId. <ForeignKey> <Name>Order</Name> <Name>Customer</Name> <References>Orders</References> <Alias> <From>Order</From> <To>OrderId</To> </Alias> </ForeignKey> --> <!ELEMENT Alias (From, To)> <!-- 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 "</pre>

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
* 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.	* 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.

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
Dot is the default DigitGroupingSymbol or ThousandsSeperator. '.' Specified once per Table --> <pre><!ELEMENT DigitGroupingSymbol (#PCDATA)></pre> <!-- 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 --> <pre><!ELEMENT Command (#PCDATA)></pre> <!-- 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 --> <pre><!ELEMENT URL (#PCDATA)></pre> <!-- Textual description of specified element (Optional) -> <pre><!ELEMENT Description (#PCDATA)></pre> <!-- The logical name of specified element.	Dot is the default DigitGroupingSymbol or ThousandsSeperator. '.' Specified once per Table --> <pre><!ELEMENT DigitGroupingSymbol (#PCDATA)></pre> <!-- 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 --> <pre><!ELEMENT Command (#PCDATA)></pre> <!-- 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 --> <pre><!ELEMENT URL (#PCDATA)></pre> <!-- Textual description of specified element (Optional) -> <pre><!ELEMENT Description (#PCDATA)></pre> <!-- The logical name of specified element.

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
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 -->	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 --> <!-- Use Extension when you want to add application specific functionality to the existing standard. Name - the extension name or identifier. URL - the supplementary .xml file that corresponds to the extension. An application that extends the standard should scan the Dataset element for the presence of zero or more Extension elements. The application can use the Name element to identify the extension.

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
<pre> <!-- 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 --> </pre>	When choosing a name for your extension, do not choose a common name. This will reduce undefined results for name conflicts. It is possible that future extensions will be ratified as mandatory to meet GDPdU guidelines. --> <pre> <!-- Self-explanatory --> <!ELEMENT Range (From, (To Length)?)> <!ELEMENT FixedRange (From, (To Length))> <!-- The document element --> <!ELEMENT DataSet (Extension*, 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 --> </pre>

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
<pre> <!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))> </pre>	<pre> <!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*, AcceptNoTables?)> <!-- 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))> </pre>

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
<pre> <!-- 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 --> </pre>	<pre> <!-- 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. see line 35 for information about Alias --> <!ELEMENT ForeignKey (Name+, References, Alias*)> <!-- 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 In order to keep the backward compatibility and because the Time data type was not defined yet we allow specifying that an Alphanumeric column contains data of type Time like: <Map> <From>HHMMSS</From> <To>HHMMSS</To> </Map> </pre>

Version 1.1 vom 1. August 2002	Version 1.6 vom 1. März 2019
<pre> <!-- Documentation for table validity. --> <!-- End Complex Types --> </pre>	If the application which interprets the xml and the target application to import into are able to deal with time data type this column data could be imported as Time data type The value of the From element has to be equal with the value of the To element, in order to not be interpreted as an actual Alphanumeric data mapping in which the intention is to replace a certain text with a different text at import time, and be a valid Time format. The following formats are recognized as valid Time formats: - HH as symbols for hours - MM as symbols for minutes - SS as symbols for seconds - TT as symbols for AM/PM SS and TT are optional when defining the format. A time format separator can also be supplied if needed. Examples of valid formats: <pre> HHMM, HH:MM, HHMMSS, HHMMTT, HH:MMTT, HH:MM:SS, HHMSSSTT, HH:MM:SSTT, HHMM TT, HH:MM TT, HHMMSS TT, HH:MM:SS TT --> </pre> <pre> <!-- Documentation for table validity. --> <!-- End Complex Types --> </pre>

Tabelle 52: Gegenüberstellung Beschreibungsstandard Version 1.1 und 1.6

Abbildungsverzeichnis

Abbildung 1: Steuerprüfer.....	6
Abbildung 2: Datenträgerüberlassung	7
Abbildung 3: Datenorganisation.....	8
Abbildung 4: Importprozess	11
Abbildung 5: XML-DTD.....	16
Abbildung 6: Beschreibungsstandard – Organisation	17
Abbildung 7: Beschreibungsstandard – DataSet.....	18
Abbildung 8: Beschreibungsstandard – FixedLength.....	19
Abbildung 9: Beschreibungsstandard – VariableLength.....	20
Abbildung 10: Beispiel 1 – Datei index.xml.....	42
Abbildung 11: Beispiel 2 – Tabellen mit Bewegungsdaten und Stammdaten.....	43
Abbildung 12: Beispiel 2 – Stammdaten	43
Abbildung 13: Beispiel 2 – Bewegungsdaten.....	44
Abbildung 14: Beispiel 2 – Beispiel eines Fremdschlüssels mit mehreren Aliassen	45
Abbildung 15: Beispiel 3 – Datei index.xml.....	46
Abbildung 16: Beispiel 3 – Datei uncompress.bat.....	46
Abbildung 17: Beispiel 4 – Datei index.xml mit Datentyp Zeit	49
Abbildung 18: Range-Angabe.....	52

Tabellenverzeichnis

Tabelle 1: Änderungshistorie Beschreibungsstandard	3
Tabelle 2: Änderungshistorie Dokument	3
Tabelle 3: Übersicht über die gängigen Operatoren	21
Tabelle 4: Liste der Namen und Nummern für die Zeichen " & < >	21
Tabelle 5: ELEMENT Version	22
Tabelle 6: ELEMENT Location	22
Tabelle 7: ELEMENT Comment	22
Tabelle 8: ELEMENT Length	22
Tabelle 9: ELEMENT References	23
Tabelle 10: ELEMENT From	23
Tabelle 11: ELEMENT To	23
Tabelle 12: ELEMENT MaxLength	23
Tabelle 13: ELEMENT TextEncapsulator	24
Tabelle 14: ELEMENT Accuracy	24
Tabelle 15: ELEMENT ImpliedAccuracy	24
Tabelle 16: ELEMENT Format	25
Tabelle 17: ELEMENT DecimalSymbol	25
Tabelle 18: ELEMENT DigitGroupingSymbol	25
Tabelle 19: ELEMENT Command	26
Tabelle 20: ELEMENT URL	26
Tabelle 21: ELEMENT Description	27
Tabelle 22: ELEMENT Name	27
Tabelle 23: ELEMENT Epoch	28
Tabelle 24: ELEMENT ColumnDelimiter	28
Tabelle 25: ELEMENT ColumnDelimiter – Liste gängiger Trennzeichen	28
Tabelle 26: ELEMENT RecordDelimiter	29
Tabelle 27: ELEMENT RecordDelimiter – Liste gängiger Trennzeichen	29
Tabelle 28: ELEMENT SkipNumBytes	29
Tabelle 29: ELEMENT Range	29
Tabelle 30: ELEMENT FixedRange	30
Tabelle 31: ELEMENT AlphaNumeric	30
Tabelle 32: ELEMENT Date	30
Tabelle 33: ELEMENT Numeric	31
Tabelle 34: ELEMENT UTF8	32
Tabelle 35: ELEMENT FixedLength	33
Tabelle 36: ELEMENT FixedColumn	33
Tabelle 37: ELEMENT FixedPrimaryKey	33
Tabelle 38: ELEMENT VariableLength	33
Tabelle 39: ELEMENT VariableColumn	34
Tabelle 40: ELEMENT VariablePrimaryKey	34
Tabelle 41: ELEMENT DataSupplier	34
Tabelle 42: ELEMENT Media	34
Tabelle 43: ELEMENT Table	35
Tabelle 44: ELEMENT ForeignKey	35
Tabelle 45: ELEMENT Map – Redefinitionen	36
Tabelle 46: ELEMENT Map	36

Tabelle 47: ELEMENT Validity	37
Tabelle 48: ELEMENT AcceptNoTables	37
Tabelle 49: ELEMENT Alias	38
Tabelle 50: ELEMENT Extension	39
Tabelle 51: Gegenüberstellung Beschreibungsstandard Version 1.0 und 1.1	58
Tabelle 52: Gegenüberstellung Beschreibungsstandard Version 1.1 und 1.6	68