



**IT- og Telestyrelsen**

Ministeriet for Videnskab  
Teknologi og Udvikling

# OIOUBL Guideline

UBL 2.0 UUID

OIOUBL UUID

G32

Version 1.1



This release is protected by Creative Commons License, Naming 2.5 

# Colophon

## Contact:

Danish National IT and Telecom Agency

E-mail: [oioubl@itst.dk](mailto:oioubl@itst.dk)

## OIOUBL Version 2.01

April 2007

Ministry of Science, Technology and Innovation  
National IT and Telecom Agency

Data Standardization Office  
Holsteinsgade 63  
DK-2100 Copenhagen Ø  
Phone +45 3545 0000  
Fax +45 3545 0010  
<http://www.itst.dk>  
[itst@itst.dk](mailto:itst@itst.dk)

Copyrights for this release in accordance with Creative Common, Naming 2.5: 

*Permission is granted to:*

- *produce processed works based on this document*
- *reproduce and make the document available to the public*
- *use the document for commercial purposes*  
*provided that the Danish National IT & Telecom Agency be clearly referenced as the source of this release.*

Further information about these rights is available at  
<http://creativecommons.org/licenses/by/2.5/deed.da>.

# Contents

1. Preface .....	4
1.1 Purpose of this document.....	4
1.2 Conclusions and recommendations.....	4
2. Relevant UBL Classes and Elements.....	5
2.1 DK field names and cardinality .....	5
3. Description.....	6
3.1 Example of a UUID.....	6
3.2 Algorithm.....	6
4. Examples.....	8
4.1 UUID.....	8
5. Terms and abbreviations.....	9

## 1. Preface

This guideline is one out of a series of documents describing the purpose and use of the business documents that comprise the Danish localization of UBL 2.0, known as OIOUBL.

For each business document, a guideline document has been prepared, as well as general guidelines describing the use of elements that apply across the documents.

### 1.1 Purpose of this document

This guideline is a work document that is the basis of the "cross-referential" description of *UUID* in OIOUBL.

### 1.2 Conclusions and recommendations

- *UUID* should be used whenever possible.
- When using *UUID* in a document, it is important that the *UUID* is generated every time the document is generated, i.e. this *UUID* identifies this instance specifically.
- When used at line level, the *UUID* can identify a line across all document instances. This may be used in, for instance, RDBMS where no double keys should be created in a line table, but instead only a primary key defined by *UUIDs*.
- The receivers of a document with *UUID* is obligated to detach the documents *UUID* at header level on all OIOUBL documents.
- The receivers of a document with *UUID* are not obligated to detach it at line level or deeper inside the document, unless it is used as a reference to another document.

## 2. Relevant UBL Classes and Elements

- *UUID*

### 2.1 DK field names and cardinality

Please note that classes/elements that are not used in a DK setting have been excluded from the below table.

UBL name	DK-name	Use
UUID	UniversaltUniktID	0..1

### 3. Description

*UUID* is a document instance ID generated by the source business system (such as ERP).

*UUID* and OIOUBL occur both at header and at line level. If a document or a line contains a *UUID*, other documents can refer directly to this exact instance and possible line without further explanations.

When sending copies the *UUID* must be generated again.

When a document is converted to, for example, other formats the *UUID* must be transferred into the new format or destination without any alterations.

#### 3.1 Example of a UUID

```
<cbc:UUID>6E09886B-DC6E-439F-82D1-7CCAC7F4E3B3</cbc:UUID>
```

Figure 1. Example of UUID

#### 3.2 Algorithm

Basically, a *UUID* is a 128 bit number represented in hexadecimals.

*UUID* is a standardized Unique ID, which, among other things, is described in the Internet Task Force RFC 4122. The formal specification of a UUID is (expressed in Backus-Naur Format):

```
UUID           = time-low "-" time-mid "-"
                  time-high-and-version "-"
                  clock-seq-and-reserved
                  clock-seq-low "-" node

time-low       = 4hexOctet
time-mid       = 2hexOctet
time-high-and-version = 2hexOctet
clock-seq-and-reserved = hexOctet
clock-seq-low  = hexOctet
node           = 6hexOctet
hexOctet       = hexDigit hexDigit
hexDigit       =
    "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9" /
    "a" / "b" / "c" / "d" / "e" / "f" /
    "A" / "B" / "C" / "D" / "E" / "F"
```

Figure 2. Algorithm for UUID

The "node" field is a number which the sender can make unique, i.e. something the sender has been assigned from an organization that guarantees unique IDs. The number must be 12 hexadecimal long, i.e. shorter than  $16^{12}$ . An example could be a Mac address on a network card, or an EAN location number (GLN) that is modified to match the format.

Further information about *UUID* is available at: "<http://www.ietf.org/rfc/rfc4122.txt>"

## 4. Examples

An example of a completed XML is shown below.

### 4.1 *UUID*

An example of a *UUID*.

```
<cbc:UUID>6E09886B-DC6E-439F-82D1-7CCAC7F4E3B3</cbc:UUID>
```

## 5. Terms and abbreviations

Term:	Explanation:
Header level	Fields on header level are all the fields that are found directly under the root element (the top element) in the XML structure. Fields on header level apply to the whole document.
Line level	Fields at line level, contrary to field at header level, only apply to the specific document line
Class	A class is a collection of fields. For example, the Price class contains fields such as PriceAmount, BaseQuantity, etc.
Fields	A field is an element in the XML structure. For example, the PriceAmount is the field containing the price in an invoice line.
Attributes	In an XML element, frequently it is possible to specify a property for the field in an attribute, e. g. the attribute unitCode in which the unit for a quantity may be specified, as in the example: <code>&lt;cbc:BaseQuantity unitCode="BO"&gt;1&lt;/cbc:BaseQuantity&gt;</code>