

# **OIOXML NDR**

## **Navngivnings- og Design Regler**

**Version 3.0**

**Dato: 2004-12-15**

**Publikation: OIO 6 [3:2004]**

**IT- og Telestyrelsen**

---

# Indholdsfortegnelse

Forord .....	v
1. OIOXML grundlaget .....	v
2. Denne publikation .....	v
3. Version .....	v
4. Målgruppe .....	v
5. Forudsætninger .....	vi
5.1. Anbefalet læsning .....	vi
5.2. Påkrævet læsning .....	vi
5.3. Afgrænsning .....	vi
6. Kommentarer .....	vi
1. Introduktion .....	1
1.1. Formål .....	1
1.2. Status .....	1
1.3. Terminologi og notationer .....	1
1.3.1. Klassifikation af regler .....	1
1.3.2. Regelprefix værdier .....	1
1.3.3. Brugen af eksempler .....	2
1.3.3.1. Foruddefinerede skemaer i Kerneklassen .....	2
1.3.4. Specielle bemærkninger .....	3
2. OIOXML regler .....	4
2.1. Genbrug af eksisterende OIOXML elementer og typer [OIO-1] [OIO-2] .....	4
2.2. Genbrug af de indbyggede simple typer [OIO-3] .....	5
2.3. Genbrug af element fremfor type [OIO-4] .....	6
2.4. Genbrug af seneste skemaversion [OIO-5] .....	8
2.5. Et OIOXML skema's indhold [OIO-6] [OIO-7] .....	8
2.6. Skemareferencer [OIO-8] .....	10
2.7. OIOXML skemaer i Infostrukturbasen [OIO-9] .....	10
2.8. Systemuafhængige OIOXML skemaer [OIO-10] [OIO-11] .....	10
2.9. Klare og entydige skemaer [OIO-12] .....	11
3. Generelle XML Skema regler .....	12
3.1. Valg af skemasprog [GXS-1] .....	12
3.2. Version af XML [GXS-2] .....	12
3.3. Valg af encoding scheme [GXS-3] .....	12
3.4. Tilknytning til namespace [GXS-4] .....	12
3.5. Skemareferencer [GXS-5] [GXS-6] .....	13
3.6. Anvendelse af notation [GXS-7] .....	13
3.7. Anvendelse af schemaLocation [GXS-8] .....	13
4. Navngivning .....	15
4.1. Generelle navngivningsregler .....	15
4.1.1. Unik navngivning [GNR-1] .....	15
4.1.2. Navngivningsmodellen for elementer, attributter og typer [GNR-2] ..	15
4.1.2.1. Termen Objekt [GNR-2a] [GNR-2b] .....	16
4.1.2.2. Termen Egenskab [GNR-2c] .....	16
4.1.2.3. Termen Repræsentation [GNR-2d] .....	16
4.1.2.4. Identiske eller enslydende fraser [GNR-2e] .....	17
4.1.2.5. Entalsform af navne [GNR-2f] .....	18
4.1.2.6. Forkortelser og akronymer [GNR-2g] .....	18

---

4.1.2.7. Bindeord med mere [GNR-2h] .....	18
4.1.2.8. Anvendelse af tegn i navne [GNR-2i] .....	18
4.2. Engelsk/dansk sprogvvalg i navngivning .....	18
4.2.1. Sprogvvalg for et OIOXML skema .....	19
4.2.1.1. Engelsk og dansk OIOXML skema [LNR-1] .....	19
4.2.1.2. Sprogangivelse med xml:lang [LNR-2] .....	19
4.2.1.3. Ord- og fagbøger for et sprog [LNR-3] [LNR-4] .....	19
4.2.1.4. Anvendelse af Æ, Ø og Å [LNR-5] .....	20
4.2.2. Sprogvvalg i relation til OIOXML klasserne .....	20
4.2.2.1. Sprogvvalg i Kerneklassen [LNR-6] .....	20
4.2.2.2. Sprogvvalg i Domæne- og NDR-klassen [LNR-7] [LNR-8] [LNR-9] .....	20
4.2.3. Kontrolleret sprogsift i et OIOXML skema .....	21
4.2.3.1. Reference fra engelsk til dansk OIOXML skema [LNR-10] ....	21
4.2.3.2. Genbrug i engelske elementer og attributer [LNR-11] .....	21
4.2.3.3. Genbrug i engelske typer [LNR-12] [LNR-13] .....	22
4.2.3.4. Vilkårligt sprogsift måske mulig via ny konstruktionen .....	22
4.3. Navngivning af typer .....	23
4.3.1. Type suffix [TPN-1] .....	23
4.3.2. Navne for komplekse typer [TPN-2] .....	23
4.3.3. Anvendelse af UpperCamelCase [TPN-3] .....	23
4.3.4. Navne for støttetyper [TPN-4] .....	23
4.4. Navngivning af elementer .....	24
4.4.1. Sammenhæng mellem element- og typenavn [ELN-1] .....	24
4.4.2. Anvendelse af UpperCamelCase [ELN-2] .....	24
4.5. Navngivning af attributter .....	24
4.5.1. Anvendelse af lowerCamelCase [ATN-1] .....	24
4.6. Navngivning af OIOXML skema- og metadatafiler .....	25
4.6.1. Navngivning af OIOXML skema fil [FNR-1] .....	25
4.6.2. Navngivning af metadata fil [FNR-2] .....	25
5. Regler for typedefinitioner .....	26
5.1. Generelle regler for typedefinitioner .....	26
5.1.1. Stærke datatyper [GTD-1] .....	26
5.1.2. Globale typedefinitioner [GTD-2] [GTD-3] .....	27
5.1.3. Ny definition af en eksisterende type [GTD-4] .....	27
5.1.4. anyType og anySimpleType [GTD-5] .....	27
5.1.5. Håndtering af binært indhold [GTD-6] .....	27
5.1.6. Abstrakte typer [GTD-7] .....	28
5.1.7. Kontrol af typeafledninger [GTD-8] .....	28
5.1.8. Anvendelse af støttetyper [GTD-9] [GTD-10] .....	28
5.2. Regler for simple typedefinitioner .....	29
5.2.1. Anvendelse af list [STD-1] .....	29
5.2.2. Anvendelse af union [STD-2] .....	29
5.2.3. Længden af string [STD-3] .....	30
5.2.4. Repræsentation af kodelister [STD-4] .....	30
5.2.5. Værdier i enumerationer [STD-5] [STD-6] .....	31
5.2.6. Anvendelse af whitespace facetten og dertil koblede typer [STD-7] ..	31
5.3. Regler for komplekse typedefinitioner .....	31
5.3.1. Opbygning af komplekse typer [CTD-1] [CTD-2] .....	31
5.3.2. Definition af komplekse typer ved afledning [CTD-3] [CTD-4] .....	32
5.3.3. Anvendelse af blandet indholdsmodel [CTD-5] .....	32
5.3.4. Anvendelse af tom indholdsmodel [CTD-6] .....	32

---

5.3.5. Anvendelse af any [CTD-7] [CTD-8] [CTD-9] .....	33
5.3.6. Anvendelse af anyAttribute [CTD-10] .....	34
5.3.7. Komplekse typer i Kerneschemaer [CTD-11] .....	34
6. Regler for element- og attributklæringer .....	35
6.1. Regler for elementklæringer .....	35
6.1.1. Globale elementklæringer [ELD-1] [ELD-2] .....	35
6.1.2. Namespace for elementer [ELD-3] .....	35
6.1.3. Anvendelse af substitution groups [ELD-4] .....	35
6.1.4. Anvendelse af nillable [ELD-5] .....	35
6.1.5. Default- og konstantværdier [ELD-8] [ELD-9] .....	37
6.2. Regler for attributklæringer .....	37
6.2.1. Anvendelsen af attributter [ATD-1] .....	37
6.2.2. Attributter, genbrug og namespaces [ATD-2] [ATD-3] .....	37
6.2.3. Default- og konstantværdier [ATD-4] [ATD-5] .....	38
7. Versionerings- og namespace-regler .....	39
7.1. Namespaceversionering [VER-1] .....	39
7.2. Låsning af et OIOXML skema [VER-2] .....	39
7.3. Bagudkompatible udvidelser [VER-3] .....	39
7.4. Anvendelse af attributten version [VER-4] .....	39
7.5. Namespace repræsentation [NMS-1] .....	39
7.6. Navngivning af prefix i namespaceklæringer [NMS-2] .....	40
8. Dokumentation og metadata .....	41
8.1. Dokumentation .....	41
8.1.1. Angivelse af dokumentation i metadata [DOC-1] .....	41
8.1.1.1. Title [DOC-2] .....	41
8.1.1.2. DescriptionEnglish og DescriptionDanish [DOC-3] [DOC-4] .....	42
8.1.2. Dokumentation for OIOXML klasserne .....	42
8.1.2.1. Dokumentation for Kerneklassen [DOC-5] .....	42
8.1.2.2. Dokumentation for Domæne- og NDR-klassen [DOC-6] .....	42
8.1.3. Anvendelse af Labels metadata .....	43
8.1.3.1. Angivelse af brugervenlig term [DOC-7] .....	43
8.1.3.2. LabelEnglish og LabelDanish [DOC-8] [DOC-9] .....	43
8.1.4. Anvendelse af appinfo [DOC-10] .....	44
8.1.5. Dokumentation af kodeliste [DOC-11] .....	44
8.1.6. Afleveringsdokument [DOC-12] .....	44
8.2. Metadata .....	44
8.2.1. Metadata for OIOXML skemaer [MTA-1] .....	44
A. OIOXML NDR Quick Reference .....	46
Stikordsregister .....	57

---

# Forord

## 1. OIOXML grundlaget

*OIOXML grundlaget* består af et antal publikationer, som hver afdækker vigtige emneområder af OIOXML paradigmet. Disse OIOXML publikationer spiller en informerende og vejledende rolle for alle, som har berøring med OIOXML paradigmet. De enkelte OIOXML publikationer fungerer bl.a. som referencegrundlag med både normative og informative definitioner og retningslinier for bl.a.:

- Informationsarkitekter til datamodellering
- OIOXML skemaudviklere til modellering og udvikling af OIOXML skemaer
- OIOXML sagsbehandlere til sagsbehandling og godkendelse af OIOXML skemaer i regi af XML Komitéen og tilknyttede instanser som f.eks. domænekomitéer

## 2. Denne publikation

Denne publikation *OIOXML Navngivnings- og Design Regler* (engelsk: *OIOXML Naming and Design Rules*), også kort betegnet *OIOXML NDR* eller bare *NDR*, definerer de regler, som et XML skema skal overholde for at kunne blive godkendt som et OIOXML skema til anvendelse indenfor OIOXML regi. Godkendte OIOXML skemaer er altid grundlæggende baseret på XML Schema anbefalingen.

*Denne publikation* tilhører en serie, der vedrører *OIOXML skemaudvikling og -vedligeholdelse* og som samlet til formål at dokumentere alle aspekter af OIOXML paradigmet, som berører udvikling og vedligeholdelse af XML skemaer i OIOXML regi. Publikationer i denne serie afdækker emner som bl.a.:

- definitioner af centrale begreber for OIOXML skemaer
- regelgrundlag for udvikling af OIOXML skemaer
- vedligeholdelse og versionering af OIOXML skemaer
- brugervejledninger til udvikling af OIOXML skemaer

## 3. Version

Denne publikation er 3. version af OIOXML NDR.

## 4. Målgruppe

Denne publikation henvender sig til de personer, som har til opgave at udvikle og standardisere datamodeller og grænseflader mellem informationssystemer i form af XML skemaer.

## 5. Forudsætninger

Denne publikation forudsætter et vist kendskab til IT, herunder XML og især *XML Schema anbefalingen* fra W3C. Derudover er følgende læsning anbefalet og påkrævet.

### 5.1. Anbefalet læsning

Følgende publikationer anbefales læst for at opnå den fulde gavn af NDR:

- *OIO-1: OIOXML Paradigmet - Overblik* : Denne informative publikation giver et generelt overblik over OIOXML paradigmet.
- *OIO-4: Infostrukturbasen - Overblik* : Denne informative publikation giver et generelt overblik over Infostrukturbasen.
- *OIO-7: OIOXML Metadata – Definitioner* : Denne normative publikation gennemgår alle de metadataoplysninger, der er for et OIOXML skema.

### 5.2. Påkrævet læsning

Følgende publikationer kræves læst for at have fuld gavn af NDR'en:

- *OIO-2: OIOXML Skemaer - Definitioner* : Denne normative publikation definerer de grundlæggende termer/begreber for et OIOXML skema.
- *OIO-3: OIOXML Klasser - Definitioner* : Denne normative publikation definerer de grundlæggende termer/begreber for OIOXML klasserne.
- *OIO-5: OIOXML Skemaaflevering - Definitioner* : Denne normative publikation definerer de grundlæggende termer/begreber for en OIOXML skemaaflevering.

### 5.3. Afgrænsning

Denne publikation er ikke tænkt som en lærebog i XML eller i XML skemadesign, da dette ligger udenfor rammerne af formålet med NDR. NDR er primært tænkt som et referenceværk for XML skemaudvikleren, der ønsker at udvikle skemaer, der kan indgå i den fællesoffentlig datamodel, baseret på genbrug af eksisterende OIOXML skemaer.

## 6. Kommentarer

Kommentarer, spørgsmål og forslag til forbedringer bedes rettet til XML Sekretariatet <xml@itst.dk>.

---

# Kapitel 1. Introduktion

## 1.1. Formål

*OIOXML NDR* er udarbejdet på foranledning af den fællesoffentlige XML Komité, hvor den er et vigtigt instrument i XML Komitéens standardiseringsarbejde. NDR skal bruges til at afgøre, om et XML skema kan godkendes som et OIOXML skema.

Formålet med denne version 3.0 af OIOXML NDR har været at opdatere reglerne, så de er mere repræsentative og i bedre overensstemmelse med nutidige krav for udvikling af OIOXML skemaer. Dernæst har formålet også været, at NDR som helhed fremstår som mere klar, entydig og brugervenlig for skemaudviklerne, så de er i stand til at skabe resultater med en højere kvalitet til gavn for standardisering og dataudveksling i offentligt såvel som i privat regi. Fælles regler øger læsbarheden og letter fortolkningen af de data, der udveksles, samt hjælper med til at sikre værktøjsunderstøttelse for udviklede skemaer. Udviklere af OIOXML skemaer vil, ved genbrug af eksisterende godkendte OIOXML skemaer og ved overholdelse af NDRs regler, kunne udvikle nye skemaer, der også kan opnå OIOXML godkendelse og dermed bidrage til standardiseringsprocessen.

## 1.2. Status

Denne publikation er godkendt af XML Komitéen med *normativ* status.

## 1.3. Terminologi og notationer

### 1.3.1. Klassifikation af regler

Reglernes vægtning er gradueret med inspiration fra Internet Engineering Task Force RFC 2119 "Key words for use in RFCs to Indicate Requirement Levels" [<http://www.ietf.org/rfc/rfc2119.txt>]. Nøgleordene "SKAL", "MÅ IKKE", "BØR", "BØR IKKE" og "MÅ" skal i definition af reglerne tillægges følgende betydning:

SKAL	Dette ord betyder, at det definerede er et absolut krav.
MÅ IKKE	Denne frase betyder, at det definerede er fuldstændig forbudt.
BØR	Dette ord betyder, at det definerede er et krav. Kun hvis der findes vægtige grunde (der skal accepteres som gyldige), hvor de fulde konsekvenser skal være forstået og nøje overvejet, kan man se bort fra det definerede.
BØR IKKE	Denne frase betyder, at det definerede forbydes. Kun hvis der findes vægtige grunde (der skal accepteres som gyldige), hvor de fulde konsekvenser skal være forstået og nøje overvejet og hvor det som defineres som frarådeligt, er acceptabelt eller sågar nyttigt, kan man anvende det definerede.
MÅ	Dette ord, eller ordet "KAN", betyder, at det definerede er valgfrit.

### 1.3.2. Regelprefix værdier

Alle reglerne er grupperet og navngivet både med en kode, en tekst og en titel. Navngivning af koderne er baseret på de prefix som anvendes i de Navngivnings- og designregler der er udviklet til Universal Business Language (UBL) [<http://www.oasis-open.org/committees/ubl/>].

Regelprefix	Regelemne
ATD	Attributerklæring
ATN	Attributnavngivning
CTD	Kompleks typedefinition
DOC	Skemadokumentation
ELD	Elementerklæring
ELN	Elementnavngivning
FNR	Filnavngivning
GNR	Generel navngivning
GTD	Generel typedefinition
GXS	Generel XML Schema
LNR	Sprogrelateret navngivning
MTA	Metadata
NMS	Namespace
OIO	OIOXML modellering
STD	Simpel typedefinition
TPN	Typenavngivning
VER	Versionering

### 1.3.3. Brugen af eksempler

Det skal bemærkes, at de anvendte skemaeksempler i denne version af NDR er både opdigtede og mindre komplicerede end i tidligere versioner med det formål at skabe bedre eksempler, der kun formidler det aspekt, som eksemplet skal illustrere.

**BEMÆRK:** når der i eksemplerne i reglerne refereres til “eksisterende” elementer og typer i diverse OIOXML klasser, menes der hermed helt præcis de elementer og typer, der er angivet her i dette afsnit. Der refereres IKKE til eksisterende elementer og typer, der reelt findes i Infostrukturbasen. Skulle der være et overlap mellem eksempelskemaerne i dette kapitel og skemaer i Infostrukturbasen er det et rent tilfælde.

#### 1.3.3.1. Foruddefinerede skemaer i Kerneklassen

Følgende eksempel-skemaer hører under Kerneklassen:

*FirstNameText.xsd:*

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples">

  <element name="FirstNameText" type="string"/>

</schema>
```

*MiddleNameText.xsd:*

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
```



```
targetNamespace="http://www.oioxml.dk/examples">
  <element name="MiddleNameText" type="string"/>
</schema>
```

### *LastNameText.xsd:*

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples">
  <element name="LastNameText" type="string"/>
</schema>
```

### *TelephoneIdentifier.xsd:*

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">
  <element name="TelephoneIdentifier" type="xmp:TelephoneIdentifierType"/>
  <simpleType name="TelephoneIdentifierType">
    <restriction base="string">
      <pattern value="(\+)?[0-9]{5,20}" />
    </restriction>
  </simpleType>
</schema>
```

### 1.3.4. Specielle bemærkninger

Reglerne i denne publikation starter ofte med formuleringen:

- “Et OIOXML skema tilhørende Kerneklassen...”
- “Et OIOXML skema tilhørende Domæneklassen...”
- “Et OIOXML skema tilhørende NDR-klassen...”

Dette kan være årsag til lidt forvirring, idet et NDR-skema, som er blevet ophøjet til Kerneschema, stadig vil tilhøre dets grundklasse, nemlig NDR-klassen og teoretisk set kunne gå under betegnelsen et NDR-skema (et Kerneschema og Domæneschema vil altid vil tilhøre en af grundklasserne NDR-klassen eller Adoptionsklassen).

Hvis formuleringen “*Et OIOXML skema tilhørende NDR-klassen...*” benyttes, skal det i denne publikation forstås på den måde, at der henvises til et OIOXML skema, som kun tilhører NDR-klassen og ikke en af de to genbrugsklasser, Kerneklassen eller Domæneklassen.

---

## Kapitel 2. OIOXML regler

OIOXML reglerne vedrører den overordnede anvendelse af OIOXML skemaer og dækker følgende områder:

- Genbrug af OIOXML skemaer
- Organisering af OIOXML skemaer

### 2.1. Genbrug af eksisterende OIOXML elementer og typer [OIO-1] [OIO-2]

Genbrugstanken er central for OIOXML. Af den grund er de to følgende regler (specielt den første) og selvfølgelig overholdelsen af dem særdeles vigtige for at etablere et højt niveau af skemagenbrug.

**[OIO-1]** Et OIOXML skema SKAL genbruge eksisterende elementer eller typer fra *Kerneklassen (OIOXML Core Class)* og *Domæneklassen (OIOXML Domain Class)*.

Denne regel kræver, at man genbruger eksisterende kerne- og domænekomponenter. Man må med andre ord ikke definere en ny type eller erklære et nyt element, hvis tilsvarende element eller type allerede eksisterer i Kerneklassen eller Domæneklassen og som repræsenterer den samme information.

#### Eksempel: Definér en sammensat type, der repræsenterer en persons fulde navn

*Første løsning:* Uden hensyn til eksisterende komponenter i Kerne- og Domæneklassen defineres i første omgang den sammensatte type *PersonNameType* i følgende skema:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <element name="PersonName" type="xmp:PersonNameType" />

  <complexType name="PersonNameType">
    <sequence>
      <element name="FirstName" type="string"/>
      <element name="MiddleName" type="string"/>
      <element name="LastName" type="string"/>
    </sequence>
  </complexType>
</schema>
```

*PersonNameType* er en repræsentation af en persons fulde navn med lokalt erklærede underelementer til repræsentation af fornavn, mellemnavn og efternavn.

Dette skema er ikke et OIOXML skema, idet [OIO-1] bl.a. ikke overholdes. I Kerneklassen (se Afsnit 1.3.3, "Brugen af eksempler") eksisterer der nemlig 3 skemaer, som hver definerer kerneelementer til angivelse af fornavn, mellemnavn og efternavn på en person. Disse kerneelementer skal ifølge [OIO-1] genbruges.

*Anden løsning:* Med denne oplysning kan skemaet i første løsning rettes til følgende korrekte

OIOXML skema, hvor [OIO-1] er overholdt:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="FirstNameText.xsd"/>
  <include schemaLocation="MiddleNameText.xsd"/>
  <include schemaLocation="LastNameText.xsd"/>

  <element name="PersonName" type="xmp:PersonNameType"/>

  <complexType name="PersonNameType">
    <sequence>
      <element ref="xmp:FirstNameText"/>
      <element ref="xmp:MiddleNameText"/>
      <element ref="xmp:LastNameText"/>
    </sequence>
  </complexType>
</schema>
```

En konsekvens af denne tilretning er også, at de før frit valgte elementnavne for fornavn, mellemnavn og efternavn nu er ændret til de standardiserede kerneelementers navne. Desuden er den sammensatte type's underelementerne ikke længere lokalt erklærede, men referencer til globalt erklærede elementer. Hermed er skemaet heller ikke i modstrid med andre regler, bl.a. [ELD-1]/[ELD-2] og [OIO-6]/[OIO-7].

**[OIO-2]** Et OIOXML skema BØR genbruge eksisterende elementer eller typer fra *NDR-klassen (OIOXML NDR Class)*.

Man bør også genbruge NDR-klassens komponenter og specielt skal man være opmærksom på, at optimal genbrug mellem en skemaafleverings egne skemaer gennemføres. Generelt vil genbrug altid reducere antallet af skemaer og dermed reducere mængden af redundant information og samlet skabe et bedre overblik i en større skemaaflevering med mange skemaer.

## 2.2. Genbrug af de indbyggede simple typer [OIO-3]

**[OIO-3]** Et OIOXML skema SKAL genbruge de indbyggede simple typer i XML Schema frem for at definere egne typer til at repræsentere den samme information.

Denne regel forebygger, at man ikke definerer sine egne versioner af de indbyggede simple typer i XML Schema. Det er f.eks. ikke tilladt at definere følgende skema til repræsentation af en dato:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <element name="MyDate" type="xmp:DateType"/>

  <simpleType name="DateType">
    <restriction base="string">
      <pattern value="[0-9]{8}"/>
    </restriction>
  </simpleType>
</schema>
```

Den indbyggede simple type *date* eksisterer allerede til at repræsentere en dato og skal altid anvendes. Den korrekte udformning er:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <element name="MyDate" type="date"/>

</schema>
```

### 2.3. Genbrug af element fremfor type [OIO-4]

Et skema (specielt i Kerne- og Domæneklassen) vil som oftest indeholde to genbrugelige komponenter, som reelt set er to forskellige måder at repræsentere den samme information på, nemlig et element og elementets tilsvarende type. Det er imidlertid ikke altid helt klart, hvad man skal vælge og der kan være en tendens til, at typen bliver genbrugt mere end elementet, også selvom det ikke altid er nødvendigt. Følgende regel balancerer denne tendens, så elementet bliver genbrugt i alle de situationer, det er muligt.

**[OIO-4]** Et OIOXML skema BØR genbruge et element frem for dets type, hvis elementets navn og anvendelse er entydig i den konkrete sammenhæng.

Når et element eller en type genbruges, vil det oftest ske i forbindelse med definition af indholdsmodellen for en sammensat type. Indholdsmodellen for den sammensatte type vil enten referere direkte til det genanvendelige element eller indirekte til den genanvendelige type (via et til situationen erklæret element, hvis type netop er den genanvendelige type).

#### Entydig i den konkrete sammenhæng

Den kontekst, hvor entydighed er relevant, kan være på ethvert niveau ligefra det minimale omfang med entydighed i en konkret indholdsmodel (som vist i eksemplerne forned) til et større omfang med entydighed på tværs af en hel skemaaflevering med multiple skemaer eller endda helt overordnet for et helt fagområde (domæne) indeholdende flere skemaafleveringer. Det optimale er, hvis man som skemaudvikler kan skabe entydighed i så stort et omfang som muligt. I de følgende eksempler illustreres genbrug af både element og type med fokus på entydighed i indholdsmodellen.

#### Eksempel: Genbrug af Kerneelement

I Kerneklassen er der defineret et Kerneelement *TelephoneIdentifier* samt Kernetype *TelephoneIdentifierType* til repræsentation af telefonnumre (se Afsnit 1.3.3, "Brugen af eksempler"). Genbrug af Kerneelementet vil se således ud:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="TelephoneIdentifier.xsd"/>

  <element name="ContactStructure" type="xmp:ContactStructureType"/>

  <complexType name="ContactStructureType">
    <sequence>
      ...
      <element ref="xmp:TelephoneIdentifier"/>
      ...
    </sequence>
  </complexType>

</schema>
```

I dette skema sker genbruget med en reference til det globalt erklærede element *TelephoneIdentifier*.

### Eksempel: Genbrug af kernetype

Ved genbrug af kernetypen fås to skemaer med følgende udseende.

Først et nyt skema med navnet *MyPhoneIdentifier.xsd*, som netop genbruger typen:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="TelephoneIdentifier.xsd" />

  <element name="MyPhoneIdentifier" type="xmp:TelephoneIdentifierType" />

</schema>
```

Det nye element *MyPhoneIdentifier* erklæres med genbrug af kernetypen *TelephoneIdentifierType*. Dette element får følgende anvendelse i *ContactStructureType*:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="MyPhoneIdentifier.xsd" />

  <element name="ContactStructure" type="xmp:ContactStructureType" />

  <complexType name="ContactStructureType">
    <sequence>
      ...
      <element ref="xmp:MyPhoneIdentifier" />
      ...
    </sequence>
  </complexType>

</schema>
```

### Hvornår skal man genbruge elementet fremfor typen?

Genbrug af et element forudsætter, at elementet er entydigt i den givne sammenhæng. Det mest konkrete eksempel på dette er i indholdsmodellen for en sammensat type. Hvis man i en sammensat type kun har ét element, der angiver et telefonnummer, er det oplagt at anvende kerneelementet *TelephoneIdentifier*, fordi det i denne sammenhæng er entydigt. Der er jo kun ét telefonnummer. Hvis det derimod er nødvendigt med flere telefonnumre i den sammensatte type, f.eks. hjemme-, mobil- og faxnummer, så er *TelephoneIdentifier* elementet ikke længere entydigt nok og genbrug af typen bliver nødvendigt. Dette er illustreret i skemaet forneden:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="MyHomePhoneIdentifier.xsd" />
  <include schemaLocation="MyMobilePhoneIdentifier.xsd" />
  <include schemaLocation="MyFaxPhoneIdentifier.xsd" />

  <element name="ContactStructure" type="xmp:ContactStructureType" />

  <complexType name="ContactStructureType">
    <sequence>
      <element ref="xmp:MyHomePhoneIdentifier" />
      <element ref="xmp:MyMobilePhoneIdentifier" />
      <element ref="xmp:MyFaxPhoneIdentifier" />
    </sequence>
  </complexType>

</schema>
```

hvor de 3 elementer *MyHomePhoneIdentifjer*, *MyMobilePhoneIdentifjer* og *MyFaxPhoneIdentifjer* erklæres i hvert sit skema med genbrug af den samme kernetype *TelephoneIdentifjerType* (som i *MyPhoneIdentifjer.xsd* vist foroven).

Hermed overholdes reglerne [OIO-6] og [OIO-7], sammenholdt med genbrug af typen resulterer i 3 ekstra skemaer (se under [OIO-6] og [OIO-7] for mere information). Da tendensen er, at typen bliver genbrugt langt oftere end elementet, er følgerne deraf et større antal skemaer end det nogen gange er nødvendigt. Dette balanceres således via [OIO-4], så man genbruger elementet, når man kan, og derved kun genbruger typen, når det er nødvendigt.

## 2.4. Genbrug af seneste skemaversion [OIO-5]

[OIO-5] Et OIOXML skema SKAL genbruge nyeste version af et andet eksisterende OIOXML skema, hvis dette andet skema forekommer i flere versioner.

Denne regel sikrer, at skemaudviklere altid genbruger de nyeste versioner af et skema og dermed er opdateret i forhold til den udvikling af enkelte skemaer, som naturligt må forekomme.

## 2.5. Et OIOXML skema's indhold [OIO-6] [OIO-7]

[OIO-6] Et OIOXML skema tilhørende *Kerneklassen (OIOXML Core Class)* eller *Domæneklassen (OIOXML Domain Class)* SKAL indeholde én elementerklæring og, hvis elementerklæringen ikke genbruger en eksisterende type fra et andet skema, én typedefinition for elementet samt eventuelt en eller flere definitioner af *støttetyper* nødvendige for etablering af elementets typedefinition.

[OIO-7] Et OIOXML skema tilhørende *NDR-klassen (OIOXML NDR Class)* BØR indeholde én elementerklæring og, hvis elementerklæringen ikke genbruger en eksisterende type fra et andet skema, én typedefinition for elementet samt eventuelt en eller flere definitioner af *støttetyper* nødvendige for etablering af elementets typedefinition.

Centralt og vigtigt for OIOXML er at holde tingene så simple som overhovedet muligt for at balancere den uundgåelige kompleksitet, der naturligt vil opstå i forbindelse med de mange mulige relationer og afhængigheder mellem godkendte OIOXML skemaer. Denne strategi virkeliggøres ved, at ét OIOXML skema kun tillades at repræsentere ét informationsobjekt eller informationsbegreb ad gangen. Helt konkret betyder det, at et OIOXML skema kun må indeholde én elementerklæring og evt. dets typedefinition til repræsentation af dette informationsobjekt, et forhold som er et krav for Kerneklassen og Domæneklassen og bør ske for NDR-klassen. Bemærk at et OIOXML skema ikke behøver at definere en type til sin elementerklæring, men lige så naturligt kan genbruge en eksisterende type i et andet skema.

Fordelen ved at skemaet med én elementerklæring kun dækker over ét begreb er, at dette begreb fyldestgørende kan dokumenteres i OIOXML skemaets metadata (der tilknyttes kun ét sæt metadata til et OIOXML skema). Hvis et OIOXML skema repræsenterede et større antal informationsobjekter, ville det bl.a. gøre dokumentationen i metadata meget kompleks.

Denne 1-1 relation mellem skemaets ene elementerklæring og metadata sikrer dels den fulde forståelse af skemaets indhold og dels bedre fremsøgning i Infostrukturbasen.

Et OIOXML skema kan udover dets ene elementerklæring og typedefinition indeholde et antal simple typedefinitioner, som fungerer som *støttetyper* elementets typedefinition, hvis en sådan er angivet. Sådanne støttetyper kan f.eks. være nødvendige ved anvendelse af lokale attributerklæringer. Se [GTD-9] og [GTD-10] for mere information om støttetyper og [TPN-4] vedrørende navngivning af støttetyper.

Følgende eksempel illustrerer anvendelsen af [OIO-6] og [OIO-7], hvor udgangspunktet er nedenstående skema, en sammensat type *ContactStructureType*, der repræsenterer telefonnumre i en kontaktstruktur:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="TelephoneIdentifier.xsd"/>

  <element name="MyHomePhoneIdentifier" type="xmp:TelephoneIdentifierType"/>
  <element name="MyMobilePhoneIdentifier" type="xmp:TelephoneIdentifierType"/>
  <element name="MyFaxPhoneIdentifier" type="xmp:TelephoneIdentifierType"/>

  <element name="ContactStructure" type="xmp:ContactStructureType"/>

  <complexType name="ContactStructureType">
    <sequence>
      <element ref="xmp:MyHomePhoneIdentifier"/>
      <element ref="xmp:MyMobilePhoneIdentifier"/>
      <element ref="xmp:MyFaxPhoneIdentifier"/>
    </sequence>
  </complexType>

</schema>
```

Ved anvendelse af [OIO-6] og [OIO-7] ændres ovenstående skema til følgende skemaer:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="MyHomePhoneIdentifier.xsd"/>
  <include schemaLocation="MyMobilePhoneIdentifier.xsd"/>
  <include schemaLocation="MyFaxPhoneIdentifier.xsd"/>

  <element name="ContactStructure" type="xmp:ContactStructureType"/>

  <complexType name="ContactStructureType">
    <sequence>
      <element ref="xmp:MyHomePhoneIdentifier"/>
      <element ref="xmp:MyMobilePhoneIdentifier"/>
      <element ref="xmp:MyFaxPhoneIdentifier"/>
    </sequence>
  </complexType>

</schema>
```

med følgende 3 skemaer.

### *MyHomePhoneIdentifier.xsd*

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="TelephoneIdentifier.xsd"/>

  <element name="MyHomePhoneIdentifier" type="xmp:TelephoneIdentifierType"/>

</schema>
```

### *MyMobilePhoneIdentifier.xsd*

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="TelephoneIdentifier.xsd"/>

  <element name="MyMobilePhoneIdentifier" type="xmp:TelephoneIdentifierType"/>

</schema>
```

### *MyFaxPhoneIdentifier.xsd*

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oioxml.dk/examples"
  xmlns:xmp="http://www.oioxml.dk/examples">

  <include schemaLocation="TelephoneIdentifier.xsd"/>

  <element name="MyFaxPhoneIdentifier" type="xmp:TelephoneIdentifierType"/>

</schema>
```

## 2.6. Skemareferencer [OIO-8]

**[OIO-8]** En OIOXML skemaaflevering SKAL i dets enkelte skemaer altid referere enten til godkendte OIOXML skemaer i Infostrukturbasen eller til skemaer indeholdt i selve skemaafleveringen.

For at sikre at en skemaaflevering er konsistent og komplet i sig selv, er det vigtigt, at alle for skemaafleveringen relevante skemaer enten indgår i skemaafleveringen selv eller allerede er godkendte OIOXML skemaer tilstede i Infostrukturbasen (dvs. er enten NDR- eller Adoptionsskemaer). Denne regel forhindrer effektivt referencer til skemaer, som ikke er under kontrol af OIOXML, og undgår dermed, at der opstår afleveringer med "løse ender".

## 2.7. OIOXML skemaer i Infostrukturbasen [OIO-9]

**[OIO-9]** Et OIOXML skema SKAL placeres i Infostrukturbasen.

For at sikre en ensartet tilgang til godkendte OIOXML skemaer og deres metadata skal alle skemaer gemmes i Infostrukturbasen. Hermed har alle altid fri adgang til OIOXML skemaernes definitioner, erklæringer og databeskrivelser.

## 2.8. Systemuafhængige OIOXML skemaer [OIO-10] [OIO-11]

**[OIO-10]** Et OIOXML skema tilhørende *Kerneklassen (OIOXML Core Class)* eller *Domæneklassen (OIOXML Domain Class)* MÅ IKKE i dets udformning være influeret af begrænsninger i et bagvedliggende fagsystem.

**[OIO-11]** Et OIOXML skema tilhørende *NDR-klassen (OIOXML NDR Class)* BØR IKKE i dets udformning være influeret af begrænsninger i et bagvedliggende fagsystem.



OIOXML skemaer tilhørende Kerneklassen og Domæneklassen skal være systemuafhængige og bør være det for NDR-skemaer. Desto mindre binding OIOXML skemaer har til enkelte fagsystemer, desto større er den generelle anvendelse af disse skemaer.

Kerne- og Domæneskemaer er specielle i den forstand, at de skal være generelt anvendelige på tværs af den offentlige og private sektor. Som sådan skal de udformes uafhængigt af eksisterende systemer og begrænsninger. Kerne- og Domæneskemaer er en del af en "ideel facade", som kan opstilles omkring eksisterende fagsystemer.

Et eksempel er registrering af navne for personer af forskellig nationalitet. Et eksisterende fagsystem kunne være begrænset til understøttelse af ISO 8859-1 tegnsæt, der indeholder alle vesteuropæiske specialtegn. Personnavne med specialtegn, som ikke er understøttet af ISO 8859-1, vil ikke kunne få sit navn registreret korrekt i et sådant fagsystem. Derfor skal et skema til repræsentation af navne være uafhængigt af ethvert fagsystem og være dækkende for enhver anvendelse. Eksempelvis kunne typen for et fornavn være defineret som:

```
<simpleType name="FirstNameText">  
  <restriction base="string">  
    <pattern value="\p{L}*" />  
  </restriction>  
</simpleType>
```

Denne type tillader navne udtrykt i alverdens tegnsæt og kunne være et eksempel på en Kerntype, der potentielt kan bruges globalt.

De centrale Kerne- og Domæneskemaer udgør som sagt en ideel datamodel, som i visse tilfælde ikke harmonerer med eksisterende fagsystemer. Ejere af fagsystemet tvinges til at tage stilling til, hvordan den ideelle datamodel konverteres til det eksisterende fagsystem.

## 2.9. Klare og entydige skemaer [OIO-12]

**[OIO-12]** Et OIOXML skema SKAL designes så enkelt og entydigt som overhovedet muligt uden unødvendig kompleksitet og overflødige konstruktioner.

Et OIOXML skema må kun indeholde lige præcis de konstruktioner, som er nødvendige for skemaets funktion. Skemakonstruktioner, som hverken bidrager til skemaets funktion eller på anden måde, i den bredest mulige anvendelse, øger skemaets værdi, er ikke tilladt. Med andre ord hvis et skema kan normaliseres til en enklere form, uden at skemaets repræsentationelle værdi forringes, så skal denne enklere udformning vælges fremfor enhver anden udformning.

Denne regel har, som det også er angivet, til formål at gøre OIOXML skemaer nemmere at læse og forstå for *alle* parter, specielt med tanke for dem, som ikke selv har udviklet skemaerne og som ikke har den indsigt i udformningen, som udviklerne naturligt vil have.

---

## Kapitel 3. Generelle XML Skema regler

### 3.1. Valg af skemasprog [GXS-1]

**[GXS-1]** Et OIOXML skema SKAL defineres i overensstemmelse med W3C XML Schema anbefalingen (version 1.0) af 2. maj 2001: *XML Schema Part 1: Structures* og *XML Schema Part 2: Datatypes*.

Det er vigtigt at sikre interoperabilitet ved, at alle offentlige systemer bruger samme skemasprog. Ellers vil der være en risiko for, at syntaksspecifikke definitioner af fælleskomponenter ikke vil kunne genbruges.

### 3.2. Version af XML [GXS-2]

**[GXS-2]** Et OIOXML skema SKAL anvende version 1.0 af W3C XML anbefalingen af 4. februar 2004 *Extensible Markup Language (XML) 1.0 (Third Edition)*.

Alle OIOXML skemaer skal som udgangspunkt anvende version 1.0 af XML. Da der er risiko for manglende værktøjsunderstøttelse, er det ikke tilladt at bruge den nyere XML version 1.1 til OIOXML skemaer.

XML versionen angives i XML erklæringen først i skemaet:

```
<?xml version="1.0" ... ?>
<schema>
  ...
</schema>
```

### 3.3. Valg af encoding scheme [GXS-3]

**[GXS-3]** Et OIOXML skema SKAL anvende UTF-8 som encoding scheme.

Alle OIOXML skemaer skal anvende UTF-8 som encoding scheme. UTF-8 er det transformationsformat til Unicode karaktersættet, der fylder mindst, og vælges her på grund af den brede accept og anvendelse i eksisterende XML værktøjer.

Bemærk at ISO Latin 1 (ISO-8859-1), som også understøtter danske tegn, ikke må anvendes.

UTF-8 encoding scheme angives i XML erklæringen først i skemaet:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema>
  ...
</schema>
```

### 3.4. Tilknytning til namespace [GXS-4]

**[GXS-4]** Et OIOXML skema SKAL tilknyttes et namespace.

Namespaces udgør ryggraden i OIOXML versioneringsstrategien og bruges således både til versionering af enkelte OIOXML skemaer og til at skelne hele skemaafleveringer fra hinanden i separate domæner. Denne anvendelse af namespace sikrer dels, at man tydeliggør skemaernes tilknytningsforholdet samt sikrer sig imod utilsigtet navnekollision.

Namespace for et skema angives ved hjælp af *targetNamespace* attributten i *schema* rodelementet.

### 3.5. Skemareferencer [GXS-5] [GXS-6]

Et XML skema kan referere til andre skemaer på 3 forskellige måder via konstruktionerne *include*, *import* og *redefine*. Selvom disse 3 konstruktioner fungerer på forskellig måde, har de alle overordnet til opgave at tillade skemakomponenter defineret i andre OIOXML skemaer at indgå i det OIOXML skema, som refererer til dem. Anvendelsen af disse konstruktioner er angivet i følgende regler.

**[GXS-5]** Et OIOXML skema MÅ IKKE benytte *import* konstruktionen til at referere til et andet OIOXML skema, der anvender samme namespace, som skemaet der refereres fra.

Hvis man ønsker at referere til et underskema med samme namespace, som skemaet man refererer fra, skal man altid anvende *include* konstruktionen. Hvis underskemaet har et andet namespace, kan man kun benytte *import* konstruktionen.

**[GXS-6]** Et OIOXML skema MÅ IKKE benytte *redefine* konstruktionen.

Konstruktionen *redefine* inkluderer et underskema på samme måde som *include* gør, men giver også samtidig mulighed for at redefinere en i underskemaet eksisterende type's indholdsmodel uden, at typens navn af den grund ændres. Da typens navn ikke ændres, vil et skema med en sådan redefineret type slå igennem alle de steder, hvor denne type benyttes. Da dette meget nemt kan have utilsigtede sideeffekter og da entydighed er en særdeles vigtig egenskab blandt OIOXML skemaer, tillades *redefine* ikke. OIOXML garanterer herved, at en eksisterende type altid er den samme i alle sammenhænge; en garanti, der ikke bliver undermineret med redefinitioner.

### 3.6. Anvendelse af *notation* [GXS-7]

**[GXS-7]** Et OIOXML skema MÅ IKKE benytte *notation* konstruktionen.

Da denne konstruktion laver applikationsspecifikke bindinger til instanser, må den ikke benyttes. Notationer er et levn fra DTD'er, som er forsøgt taget med i XML Schema for bagudkompatibilitet uden egentlig at opnå det.

### 3.7. Anvendelse af *schemaLocation* [GXS-8]

**[GXS-8]** Et OIOXML skema SKAL angive alle dets *schemaLocation* attributter med en absolut og gyldig URL til det refererede OIOXML skemas placering i Infostrukturbasen.

Denne regel sikrer, at Infostrukturbasen kan opretholde en korrekt afdækning af afhængigheder skemaer imellem samt at skemaernes sammenhæng og fortsatte gyldighed bevares ved flytning eller kopiering fra Infostrukturbasen til en alternativ placering (f.eks. en lokal udviklingsplatform). Fordelen ved den absolutte URL er, at selv OIOXML skemaer, som er lokaliseret et andet sted end Infostrukturbasen, stadig har korrekte referencer.

## Kapitel 4. Navngivning

### 4.1. Generelle navngivningsregler

#### 4.1.1. Unik navngivning [GNR-1]

**[GNR-1]** Et OIOXML skema SKAL navngive alle dets globalt erklærede komponenter (elementer, og typer) unikt indenfor skemaets namespace og BØR navngive unikt på tværs af alle namespaces for godkendte OIOXML skemaer.

Navngivning af elementer og typer under et givet namespace skal være unik, således at to forskellige semantiske begreber ikke navngives ens og for at stimulere genbrug mest mulig.

#### 4.1.2. Navngivningsmodellen for elementer, attributter og typer [GNR-2]

**[GNR-2]** Et OIOXML skema BØR navngive alle dets globalt og lokalt erklærede komponenter (elementer, attributter samt typer) efter *ObjektEgenskabRepræsentation* navngivningsmodellen, som specificeret i følgende underregler.

OIOXML element-, attribut- og typenavne er opbygget af en sammensætningen af de tre termer *Objekt*, *Egenskab* og *Repræsentation*. Denne opbygning følger de anbefalinger, som ebXML har for navngivning og som selv er baseret på på ISO standarden 11179 "Information technology — Specification and standardization of data elements". Der er meget bred konsensus for denne navngivningskonvention internationalt. ebXMLs anvendelse er beskrevet i ebXML TR - Naming Convention for Core Components 10 May 2001 11 Version 1.04 [<http://www.ebxml.org/specs/ebCCNAM.pdf>]. ISO 11179 standarden er under videreudvikling og har nu titlen "Information technology — Metadata registries". Den seneste udgave kan findes på arbejdsgruppens hjemmeside [<http://metadata-stds.org/11179/index.html>].

Nedenstående tabel viser eksempler på navne med denne navngivningsmodel:

Objekt	Egenskab	Repræsentation	Elementnavn
Bank	Account	Identifier	BankAccountIdentifier
Person	Birth	Date	PersonBirthDate
Tax		Percentage	TaxPercentage
Amount	Type		AmountType
Person	Gender	Code	PersonGenderCode
Communication	Mode	Code	CommunicationModeCode
Country		Code	CountryCode
Currency	Exchange	Rate	CurrencyExchangeRate
Location	Type	Code	LocationTypeCode
Transport	Method	Code	TransportMethodCode
Street	Name		StreetName
OrganizationRegistration		Date	OrganizationRegistrationDate

Objekt	Egenskab	Repræsentation	Elementnavn
Address	Type	Code	AddressTypeCode

#### 4.1.2.1. Termen *Objekt* [GNR-2a] [GNR-2b]

[GNR-2a] Et navn SKAL i dets *Objekt* term beskrive det dataobjekt, som et element og dets type repræsenterer i en bestemt sammenhæng.

Termen *Objekt* er et eller flere nøgleord, som beskriver objektet, enheden eller konceptet, som det centrale dataelement eller -type relaterer til, f.eks. Bank, Person, Building, Country, Organization.

[GNR-2b] Et navn KAN udelade dets *Objekt* term i det tilfælde, hvor et element og dets type optræder i en kontekst af et objekt eller objektet er ukendt.

I de tilfælde, hvor elementet eller typen optræder i en bestemt sammenhæng, hvor objektrelationen nemt kan identificeres, eller i de tilfælde hvor objektet er ukendt, kan *Objekt* termen udelades. I følgende eksempel er det ikke nødvendigt at angive *Objekt* for *TelephoneNumber* elementet, da elementet optræder i konteksten af *Employee* elementet:

```
<Employee>
  <TelephoneNumber>+45 11223344</TelephoneNumber>
  <DepartmentName>IT Strategic Office</DepartmentName>
</Employee>
```

#### 4.1.2.2. Termen *Egenskab* [GNR-2c]

[GNR-2c] Et navn SKAL i dets *Egenskab* term ved hjælp af en eller flere kvalificerende ord beskrive en fremtrædende egenskab ved et elements og dets types *Objekt* term.

Termen *Egenskab* er et eller flere kvalificerende ord, som beskriver en egenskab ved objektet. Hvis der er flere ord, skal hver enkelt være af betydning for objektet, som bliver beskrevet. Eksempler er “Account” som i *BankAccountIdentifier*, “Birth” som i *PersonBirthDate* eller “Exit” som i *BuildingExitQuantity*. Beskrivelsen kan udelades, hvis indholdet i elementet eller typen direkte vedrører objektet og ikke en egenskab ved objektet.

#### 4.1.2.3. Termen *Repræsentation* [GNR-2d]

[GNR-2d] Et navn SKAL i dets *Repræsentation* term beskrive et elements og dets types repræsentative kategori og SKAL antage en af værdierne i OIOXML's liste over repræsentationstermer.

Termen *Repræsentation* er et nøgleord, som associerer til en simpel type. Eksempler er “Identifier” som i *BankAccountIdentifier*, “Date” som i *PersonBirthDate* eller “Quantity”

som i *BuildingExitQuantity*. Se følgende tabel for mulige repræsentationer for både engelsk og dansk navngivning:

Repræsentationsterm		Beskrivelse
Engelsk	Dansk	
Amount	Beloeb	Et antal af monetære enheder specificeret i en møntfod, hvor møntfoden enten er eksplicit eller implicit.
Code	Kode	En karakterstreng der kan bruges til at repræsentere eller erstatte en definitiv værdi eller tekst, f.eks. som forkortelse og/eller for at opnå sproguafhængighed . Koder er normalt vedligeholdt i en kodeliste per type, f.eks. farve.
Date	Dato	En dato indenfor et bestemt kalenderår, baseret på ISO 8601.
DateTime (*)	DatoTid	Et bestemt tidspunkt.
Identifier	Identifikator	En karakterstreng der, indenfor en identifikationsramme, bruges til at identificere og unikt udpege én instans af et objekt blandt alle objekter indenfor samme ramme.
Reference	Reference	En karakterstreng som bruges til at referere til en bestemt objektinstans identificeret med en Identifier (dette er en tilføjelse til de repræsentationstermer, ebXML definerer).
Indicator	Indikator	En liste af to og kun to værdier, som indikerer en tilstand som f.eks. "on/off" eller "true/false". Synonymt med "boolean".
Measure	Maal	En numerisk værdi bestemt ved at måle et objekt. Dette mål kan angives med en måleenhed; de mulige måleenheder tages fra UN/ECE Rec. 20.
Name	Navn	Et ord eller en frase som indeholder den præcise betegnelse for en person, et sted, ting eller koncept.
Percent	Procent	En rate udtrykt som en hundrededel mellem to værdier med samme måleenhed.
Quantity	Kvantitet	Et antal af ikke monetære enheder. Den er associeret med indikationen af objekter. Mængder kan angives med en eksplicit mængdeenhed.
Rate	Rate	En mængde eller et antal målt i forhold til en anden mængde eller antal, eller en fast eller passende ladning, omkostning eller værdi, f.eks. "kroner i timen", "danske kroner per EURO" eller "kilometer per liter".
Text	Tekst	En karakterstreng udtrykt generelt som ord i et sprog.
Time	Tid	Tidspunktet indenfor en ikke specificeret dag. Baseret på ISO 8601:1988.

(\*) Repræsentationstermene foroven er taget fra ebXML og er stort set alle identiske med deres termer. Undtagelserne er *DateTime*, som i ebXML betegnes *DateAndTime*, og *Reference*, som slet ikke indgår i ebXML's termer.

#### 4.1.2.4. Identiske eller enslydende fraser [GNR-2e]

[GNR-2e] Et navn SKAL, hvis det har en frase i termen *Egenskab*, som er synonymt med en frase i termen *Repræsentation*, fjerne frasen fra *Egenskab* og bibeholde den i *Repræsentation*.

Med denne regel undgås uheldige navneformuleringer, såsom "IdentificationIdentifier" som skal ændres til "Identifier".

#### 4.1.2.5. Entalsform af navne [GNR-2f]

[GNR-2f] Et navn SKAL angives på entalsform, med mindre navneordet er en flertalsform.

Et eksempel på en undtagelse er "Goods". Navne skal beskrive enkelte elementer. Hvis der er tale om flere forekomster af det samme element, skal man følge [TPN-2].

#### 4.1.2.6. Forkortelser og akronymer [GNR-2g]

[GNR-2g] Et navn BØR IKKE benytte forkortelser og akronymer.

Da forkortelser og akronymer hæmmer læsbarheden, bør de ikke benyttes. Hvis der er behov for at lave forkortelser, bør man anvende eksisterende forkortelser. Hvis der ikke eksisterer en almindelig brugt forkortelse, så find på én, som giver mening og er entydig.

Forkortelser skal skrives med store bogstaver.

#### 4.1.2.7. Bindeord med mere [GNR-2h]

[GNR-2h] Et navn SKAL opbygges af udsagnsord, navneord og tillægsord.

Det er vigtigt, at navne er så præcise som muligt og dermed tager udgangspunkt i, at både semantik og syntaks skal være så præcis som mulig. Det er ikke meningen, at navnet skal være en sætning og derfor må ord som "and", "of" og "the" ikke benyttes.

#### 4.1.2.8. Anvendelse af tegn i navne [GNR-2i]

[GNR-2i] Et navn MÅ IKKE i dets opbygning benytte underscore (\_), punktum (.) og bindestreg (-).

XML anbefalingen tillader disse tegn, men da *UpperCamelCase* er valgt som metodik for elementer og typer (se [ELN-2] og [TPN-3] for mere information), er der ikke brug for disse tegn.

### 4.2. Engelsk/dansk sprogvalg i navngivning



Som generel retningslinie for valg af sprog i forbindelse med navngivning af elementer, attributter og typer i OIOXML skemaer anbefales det, at der anvendes engelsk navngivning. Hvis det er uhensigtsmæssigt af administrative, semantiske eller andre væsentlige hensyn, kan der i stedet anvendes dansk navngivning.

De følgende 13 regler for engelsk/dansk sprogvalg præciserer anvendelsen og skal opfattes og anvendes som en helhed. Reglerne er organiseret i 3 grupper, som vedrører:

- Sprogvalg for et OIOXML skema
- Sprogvalg i relation til OIOXML klasserne
- Kontrolleret sprogsift i et OIOXML skema

### 4.2.1. Sprogvalg for et OIOXML skema

#### 4.2.1.1. Engelsk og dansk OIOXML skema [LNR-1]

[LNR-1] Et OIOXML skema SKAL navngive alle dets globalt og lokalt erklærede komponenter (elementer, attributter samt typer) i ét og samme sprog. Et OIOXML skema med engelsk navngivne komponenter betegnes et *engelsk* OIOXML skema og et OIOXML skema med dansk navngivne komponenter betegnes et *dansk* OIOXML skema.

For at sikre en entydig sproganvendelse i et OIOXML skema, tillades der kun ét sprog (engelsk eller dansk) for navngivning af erklærede komponenter i et OIOXML skema. Med erklærede komponenter menes mere præcist globale elementerklæringer, lokale attributerklæringer samt globale simple og komplekse typedefinitioner. Hvis lokalt erklærede komponenter (udover attributerklæringer, som altid er lokale og følger skemaets sprog) benytter et andet sprog end de globalt erklærede, skal reglerne [LNR-10], [LNR-11], [LNR-12] og [LNR-13] om sprogsift følges.

#### 4.2.1.2. Sprogangivelse med *xml:lang* [LNR-2]

[LNR-2] Et dansk OIOXML skema SKAL tildele attributten *xml:lang* i rodelementet *schema* værdien "DA". Et engelsk OIOXML skema KAN tildele attributten *xml:lang* i rodelementet *schema* værdien "EN". Attributten *xml:lang* MÅ IKKE antage andre værdier end "DA" og "EN" og hvis attributten ikke er specificeret, antages værdien at være "EN", dvs. skemaet antages at være et engelsk OIOXML skema.

Det skal entydigt kunne identificeres, om et skema er et engelsk eller et dansk OIOXML skema. Denne regel viser, hvorledes dette gøres ved at angive skemaets sprog ved hjælp af *xml:lang* attributten i rodelementet *schema*.

#### 4.2.1.3. Ord- og fagbøger for et sprog [LNR-3] [LNR-4]

[LNR-3] Navngivning på engelsk BØR ske i overensstemmelse med *Oxford English*

*Dictionary* eller relevante fagbøger.

**[LNR-4]** Navngivning på dansk BØR ske i overensstemmelse med *Dansk Retskrivningsordbog* eller relevante fagbøger.

Navngivning i et bestemt sprog skal følge det pågældende sprog's regler for syntaks og korrekt stavning, som specificeret i relevante ord- og fagbøger.

#### 4.2.1.4. Anvendelse af Æ, Ø og Å [LNR-5]

**[LNR-5]** Et dansk OIOXML skema må ikke anvende de danske specialtegn æ, ø og å på grund af problemer i forbindelse med manglende værktøjsunderstøttelse.

På grund af problemerne med manglende værktøjsunderstøttelse for de danske specialtegn æ, ø og å må disse specialtegn i forbindelse med dansk navngivning ikke anvendes. I stedet for anbefales det at anvende kombinationerne *ae*, *oe* og *aa* for at repræsentere henholdsvis æ, ø og å.

#### 4.2.2. Sprogvalg i relation til OIOXML klasserne

##### 4.2.2.1. Sprogvalg i Kerneklassen [LNR-6]

**[LNR-6]** Et OIOXML Skema tilhørende *Kerneklassen (OIOXML Core Class)* SKAL være et engelsk OIOXML skema.

For at bevare en ensartet anvendelse og understøtte bredest muligt genbrug både nationalt og internationalt kræves det, at skemaer tilhørende Kerneklassen altid navngives på engelsk.

##### 4.2.2.2. Sprogvalg i Domæne- og NDR-klassen [LNR-7] [LNR-8] [LNR-9]

**[LNR-7]** Et OIOXML skema tilhørende *Domæneklassen (OIOXML Domain Class)* eller *NDR-klassen (OIOXML NDR Class)* BØR være et engelsk OIOXML skema.

Det anbefales, at alle NDR- og Domæneskemaer navngives på engelsk for at støtte den internationale anvendelse af OIOXML skemaer. Hvis det ikke er muligt at navngive på engelsk, da skal [LNR-8] følges.

**[LNR-8]** Et OIOXML skema tilhørende *Domæneklassen (OIOXML Domain Class)* eller *NDR-klassen (OIOXML NDR Class)* KAN være et dansk OIOXML skema, hvis det indgår i en sammenhæng eller i et domæne eller subdomæne, hvor dansk skaber en større begrebsmæssig klarhed end engelsk.

Udover at dansk navngivning i NDR- og Domæneskemaer følger samme overordnede retningslinier i denne regel, kræver Domæneskemaer dog særlig opmærksomhed ifølge [LNR-9].

**[LNR-9]** For et OIOXML skema tilhørende *Domæneklassen (OIOXML Domain Class)* er det en domænekomité for skemaets respektive domæne, der beslutter, hvilket sprog skemaet navngives i.

Domæneskemaer har speciel status som velegnede til genbrug og det er af den grund vigtigt, at sprogvalget i et Domæneskema tager højde herfor. Det enkelte fagområde må, via dets domænekomité, ud fra egne kriterier (i forhold til fagområdet og den sammenhæng skemaet indgår i) afgøre, hvilket sprog Domæneskemaet navngives i. Dog skal retningslinierne i [LNR-8] være vejledende for beslutningen. Hvis der ikke er etableret en domænekomité for det pågældende fagområde, så vil XML Komitéen varetage denne opgave.

#### **4.2.3. Kontrolleret sprogsift i et OIOXML skema**

Der er vigtigt at sikre, at sprogsift OIOXML skemaer imellem ikke underminerer en eventuel international indsats. Derfor tillades der kun én sprogovergang mellem engelsk og dansk mellem OIOXML skemaer, der refererer og genbruger hinanden. Den vigtigste pointe i nedenstående regler er, at et engelsk OIOXML skema aldrig inkluderer eller importerer (og dermed genbruger fra) et dansk OIOXML skema. Derved eksisterer der altid en sproglig konsistens mellem et engelsk skema og alle dets underskemaer. Hvis et engelsk skema refererer til danske skemaer, vil skemaet ikke kunne anvendes i en international sammenhæng.

##### **4.2.3.1. Reference fra engelsk til dansk OIOXML skema [LNR-10]**

**[LNR-10]** Et engelsk OIOXML skema MÅ IKKE via *include* konstruktionen inkludere eller via *import* konstruktionen importere et dansk OIOXML skema.

Af ovennævnte internationale hensyn må et engelsk OIOXML skema ikke referere til (dvs. inkludere eller importere) et dansk OIOXML skema. Derved sikres sproglig konsistens mellem et engelsk skema og alle dets underskemaer på alle indlejrte niveauer. Bemærk at alle andre sprogkombinationer er tilladte, dvs. et dansk skema kan referere til et dansk eller engelsk skema og et engelsk skema kan selvfølgelig referere til et engelsk skema.

De følgende regler [LNR-11], [LNR-12] og [LNR-13] uddyber konsekvensen af [LNR-10].

##### **4.2.3.2. Genbrug i engelske elementer og attributer [LNR-11]**

**[LNR-11]** Et engelsk OIOXML skema MÅ IKKE erklære et element eller en attribut, der genbruger en dansk type defineret i et dansk OIOXML skema.

Genbrug af en dansk type i en element- eller attributerklæring i et engelsk OIOXML skema er ikke tilladt og kan i øvrigt kun ske, hvis det engelske skema refererer til et dansk skema, hvilket ikke er tilladt ifølge [LNR-10].

### 4.2.3.3. Genbrug i engelske typer [LNR-12] [LNR-13]

[LNR-12] Et engelsk OIOXML skema MÅ IKKE i en kompleks type genbruge danske komponenter (elementer og typer) definerede eller erklærede i et dansk OIOXML skema.

Genbrug af danske komponenter (dvs. elementer og typer) i en kompleks type i et engelsk OIOXML skema er ikke tilladt og kan i øvrigt kun ske, hvis det engelske skema refererer til et dansk skema, hvilket ikke er tilladt ifølge [LNR-10].

[LNR-13] Et engelsk OIOXML skema MÅ IKKE i en simpel eller kompleks type nedarve fra en dansk type erklæret i et dansk OIOXML skema.

Nedarvning fra danske typer i et engelsk OIOXML skema er ikke tilladt og kan i øvrigt kun ske, hvis det engelske skema refererer til et dansk skema, hvilket ikke er tilladt ifølge [LNR-10].

### 4.2.3.4. Vilkaarligt sprogskit måske mulig via *any* konstruktionen

På grund af reglerne [LNR-10], [LNR-11], [LNR-12] og [LNR-13] tillades et sprogskit fra engelsk til dansk ikke eksplicit i et OIOXML skema. Der kan dog alligevel være mulighed for vilkaarligt sprogskit i overgangen mellem to OIOXML dialekter (se definition af OIOXML dialekt i Afsnit 5.3.5, “Anvendelse af *any* [CTD-7] [CTD-8] [CTD-9]”) i en XML instans, uden at dette er synligt i OIOXML skemaerne.

Dette muliggøres ved hjælp af *any* konstruktionen, som fungerer som en slags “fribillet”, der i indholdsmodellen for en kompleks type repræsenterer anvendelsen af et vilkaarligt unavngivent element fra et helt andet, ikke nødvendigvis kendt, namespace, dvs. et element fra et vilkaarligt andet skema. Selve effekten af en *any* konstruktion i et skema, er kun synlig i XML instansen, da det er her XML strukturen realiseres. Konstruktionen muliggør en “kontaktflade” eller overgang mellem flere OIOXML dialekter, hvor man ikke nødvendigvis kender de skemaer, der skal anvendes på forhånd. Det er her i denne overgang, at et sprogskit er muligt, idet det skema, som erklærer det element, der anvendes på *any* konstruktionens plads i XML instansen, kan være på et andet sprog end det skema, som har specificeret *any* konstruktionen i sin indholdsmodel.

Da OIOXML ikke sætter nogen sprogkrav i forbindelse med overgangen mellem to OIOXML dialekter via *any* konstruktionen, er det tilladt, at OIOXML skemaerne bag de to dialekter anvender forskellige sprog. Men bemærk derimod, at OIOXML sætter nogle krav, som begrænser anvendelsen af *any* konstruktionen og at dette kan kollidere med et ønsket sprogskitte. Disse krav har en ren strukturel baggrund og har intet med sprog at gøre.

Så, med andre ord, hvis det er tilladt at anvende *any* konstruktionen, da er et sprogskit i den forbindelse muligt og tilladt. Se reglerne for anvendelsen af *any* konstruktionen i Afsnit 5.3.5,

“Anvendelse af any [CTD-7] [CTD-8] [CTD-9]” for mere information.

### 4.3. Navngivning af typer

#### 4.3.1. Type suffix [TPN-1]

[TPN-1] Et OIOXML skema SKAL afslutte navnet på en simpel eller kompleks type med suffix “Type”.

Navnene på datatyper skal slutte med tekststrengen “Type”. Dette suffix fjernes til det respektive elementnavn (se [ELN-1]). Dette sikrer, at man altid kan kende forskel på typer og elementer.

#### 4.3.2. Navne for komplekse typer [TPN-2]

[TPN-2] Et OIOXML skema MÅ IKKE anvende termen *Repræsentation* i navnet for en kompleks type. Termen *Egenskab* i typenavnet SKAL være “Collection” i et engelsk OIOXML skema og “Samling” i et dansk OIOXML skema, hvis typen indeholder præcist ét element og mindst 2 forekomster af dette. I alle andre tilfælde SKAL termen *Egenskab* i typenavnet være “Structure” i et engelsk OIOXML skema og “Struktur” i et dansk OIOXML skema eller helt undlades.

Navngivningen for komplekse typer håndteres anderledes end for simple typer. *Repræsentation* giver bl.a. ikke mening for sammensatte typer og skal helt udelades. Termen *Egenskab* styres af selve indholdsmodellens opbygning. Hvis indholdsmodellen for en type består af multiple forekomster af ét element, betegnes dette en liste eller samling og betegnes “Collection” (“Samling” på dansk). Ved alle andre former for indholdsmodel er det valgfrit, om man angiver “Structure” (eller “Struktur” på dansk) som *Egenskab* eller helt undlader den.

#### 4.3.3. Anvendelse af UpperCamelCase [TPN-3]

[TPN-3] Et OIOXML skema SKAL navngive dets simple og komplekse typer med *UpperCamelCase*.

*UpperCamelCase* betyder, at navnet skal begynde med et stort bogstav, herefter begynder hvert nyt ord med et stort bogstav, et eksempel:

```
<element name="StreetBuildingIdentifier" type="dkcc:StreetBuildingIdentifierType">
```

Hvis en forkortelse består udelukkende af store bogstaver, som f.eks. DK, bør det efterfølgende ord starte med et lille bogstav. F.eks. *DKAddressType* i modsætning til *DKAddressType*.

#### 4.3.4. Navne for støttetyper [TPN-4]

[TPN-4] Et OIOXML skema SKAL anvende ét underscore (\_) som prefix til en type's navn, hvis typen er en *støttetype*.

Navngivningen for støttetyper håndteres lidt anderledes end for typer generelt, idet en støttetype's navn, der ellers opbygges på sædvanlig vis, altid skal foranstilles med ét underscore (\_). Dette gøres for at signalere en forskel mellem skemaets ene typedefinition og dets evt. støttetyper. En støttetype kan kun genbruges lokalt i skemaet selvom, at den reelt er globalt erklæret. En elementerklæring, der refererer til en støttetype i et andet skema, vil blive opfattet som en fejl. Se [GTD-9] og [GTD-10] for mere information om støttetyper samt [OIO-6] og [OIO-7].

## 4.4. Navngivning af elementer

### 4.4.1. Sammenhæng mellem element- og typenavn [ELN-1]

[ELN-1] Et OIOXML skema BØR navngive dets element identisk med elementets type (uden typens "Type" suffix), hvis elementerklæring og typedefinition forekommer sammen i skemaet.

Grundlæggende skal navnet på elementet direkte afspejle navnet på den brugte type. Dette gøres ved at fjerne "Type" suffix fra typenavnet og bruge det resulterende navn som elementnavn. Denne regel er kun relevant for skemaer, som indeholder både elementerklæring og dets tilsvarende typedefinition (se også [TPN-1]).

### 4.4.2. Anvendelse af UpperCamelCase [ELN-2]

[ELN-2] Et OIOXML skema SKAL navngive dets element med *UpperCamelCase*.

*UpperCamelCase* betyder, at navnet skal begynde med et stort bogstav, herefter begynder hvert nyt ord med et stort bogstav, et eksempel:

```
<element name="StreetBuildingIdentifier" type="dkcc:StreetBuildingIdentifierType">
```

Hvis en forkortelse består udelukkende af store bogstaver, som f.eks. DK, bør det efterfølgende ord starte med et lille bogstav. F.eks. *DKaddress* i modsætning til *DKAddress*.

## 4.5. Navngivning af attributter

### 4.5.1. Anvendelse af lowerCamelCase [ATN-1]

[ATN-1] Et OIOXML skema SKAL navngive dets attributter med *lowerCamelCase*.

Når man skriver i *lowerCamelCase* skal navnet begynde med lille bogstav, herefter begynder hvert nyt ord med et stort bogstav.

```
<complexType name="CustomerType">
  <sequence>
    <element name="GivenName" type="prefix:GivenNameType"/>
    ...
  </sequence>
  <attribute name="customerIdentifier" type="prefix:CustomerIdentifierType"/>
</complexType>
```

## 4.6. Navngivning af OIOXML skema- og metadatafiler

### 4.6.1. Navngivning af OIOXML skema fil [FNR-1]

**[FNR-1]** Et OIOXML skema SKAL navngive dets fil efter modellen: *<namespace-prefix med store bogstaver>+“\_”+<elementnavn>+“.xsd”* .

Anvendelse af beskrivende navne er ikke alene vigtigt internt i et skema, men giver også konsistens og sammenhæng ved filnavngivning af XML Skemaer og andre typer af XML-dokumenter.

Fordelen ved navngivningskonventionen i denne regel er, at man umiddelbart fra filnavnet både kan se hvilket overordnet namespace skemaet kommer fra og hvilket element og eventuel tilknyttet type, der er i skemaet.

Eksempel:

```
CPR_CivilRegistrationNumber.xsd
```

### 4.6.2. Navngivning af metadata fil [FNR-2]

**[FNR-2]** Et OIOXML skema SKAL navngive dets metadatafil efter modellen: *<skemafilnavn>+“.meta.xml”* .

Denne regel sikrer, at man automatisk kan finde et OIOXML skemas metadata alene ud fra dets filnavn.

---

## Kapitel 5. Regler for typedefinitioner

### 5.1. Generelle regler for typedefinitioner

#### 5.1.1. Stærke datatyper [GTD-1]

[GTD-1] Et OIOXML skema SKAL definere alle dets simple og komplekse typer stærkest muligt.

XML Schema anbefalingen er valgt netop, fordi det giver mulighed for stærke datatyper, altså at man præcist kan beskrive udfaldsrummet af lovlige værdier. Dette giver mulighed for lave validering af instanser inden videre behandling og reducerer dermed de anstrengelser, man ellers vil have med datavalidering i applikationer.

I de tilfælde, hvor man kan beskrive det samme på flere måder, bør man vælge den form, som er mest simpel og logisk. Eksemplet nedenfor viser først en for "løs" og præsentationsorienteret definition af temperaturmålingsenheden Celsius. Der er et maksimum på 7 cifre for udfaldsrummet for Celsius, hvilket kan være uhensigtsmæssigt.

```
<simpleType name="CelsiusType">
  <restriction base="string">
    <pattern value="[+-][0-9]{1,7}" />
  </restriction>
</simpleType>
```

Celsius kunne defineres stærkere og på en måde, som er simplere, ved at basere sig på en type, der logisk danner baggrund. Der er en klar definition på det minimum, som Celsius kan antage og ingen værdi for maksimum.

```
<simpleType name="CelciusType">
  <restriction base="decimal">
    <minExclusive value="-273" />
  </restriction>
</simpleType>
```

I følgende eksempel er *CivilRegistrationNumberType* defineret som 10 cifre mellem 0 og 9.

```
<simpleType name="CivilRegistrationNumberType">
  <restriction base="string">
    <pattern value="[0-9]{10}" />
  </restriction>
</simpleType>
```

Det betyder, at *CivilRegistrationNumberType* bl.a. kan antage værdien 1313131313. Heraf kan det ses, at vedkommende med dette CPR-nummer er født i den 13. måned. Dette er som bekendt ikke muligt, men XML skemaparseren vil ikke fange fejlen. Datatypen er for svag.

Nedenstående er et forsøg på at gøre *CivilRegistrationNumberType* til en stærk datatype.

```
<simpleType name="CivilRegistrationNumberType">
  <restriction base="string">
    <pattern value="(((0[1-9]|1[0-9]|2[0-9]|3[0-9])
      (01|03|05|07|08|10|12))|
      ((0[1-9]|1[0-9]|2[0-9]|30)(04|06|09|11))|
      ((0[1-9]|1[0-9]|2[0-9])(02)))[0-9]{6}" />
  </restriction>
</simpleType>
```



```
</restriction>  
</simpleType>
```

Ved hjælp af det nye regulære udtryk for *CivilRegistrationNumberType* er det sikret, at kun valide CPR-numre kommer igennem parseren med undtagelse af CPR-numre, der indeholder datoen 29.02. for andet end skudår. I dette tilfælde blev typen mere kompliceret, da der ikke er noget alternativ. Det er vigtigst at gøre typen stærk fremfor letlæselig.

### 5.1.2. Globale typedefinitioner [GTD-2] [GTD-3]

**[GTD-2]** Et OIOXML skema tilhørende *Kerneklassen (OIOXML Core Class)* eller *Domæneklassen (OIOXML Domain Class)* SKAL definere alle dets simple og komplekse typer globalt.

**[GTD-3]** Et OIOXML skema tilhørende *NDR-klassen (OIOXML NDR Class)* BØR definere alle dets simple eller komplekse typer globalt.

Globale typedefinitioner befordrer genbrug og fjerner mulig redundans, hvilket er et meget vigtigt aspekt af OIOXML. Desuden bliver OIOXML skemaerne mere overskuelige og ensartede, og der sikres imod navnekollision.

Disse regler er også ensbetydende med, at typedefinitioner altid skal navngives og anonyme typer ikke er tilladt.

### 5.1.3. Ny definition af en eksisterende type [GTD-4]

**[GTD-4]** Et OIOXML skema MÅ IKKE definere en ny simpel eller kompleks type identisk med en simpel eller kompleks type fra et andet eksisterende OIOXML skema.

For at begrænse redundans og prioritere genbrug, er det ikke tilladt at definere en ny type identisk med en eksisterende, hvis formålet kun er at etablere den samme type med et til situationen mere sigende navn. Man må ikke definere en ny type ud fra en eksisterende uden reelt også at ændre typens indhold i forhold til den eksisterende type.

### 5.1.4. *anyType* og *anySimpleType* [GTD-5]

**[GTD-5]** Et OIOXML skema MÅ IKKE genbruge de indbyggede ur-typer *anyType* og *anySimpleType*.

Da disse typer er de svagest mulige datatyper, må de ikke benyttes.

### 5.1.5. Håndtering af binært indhold [GTD-6]

**[GTD-6]** Et OIOXML skema BØR benytte de indbyggede simple typer *anyURI* eller *base64Binary* til at håndtere binært indhold.

Da XML er karakterbaseret, er der ingen mulighed for at indlejre rå binær data i XML. Dette løses ved enten at referere til det dokument, man ønsker at videresende, eller ved at formatere dokumentet til base64 formattet.

```
<element name="ImageURI" type="xsd:anyURI"/>
<element name="Image" type="xsd:base64Binary"/>
```

Man bør definere et element eller en attribut, som tilkendegiver dokumentets type baseret på `MimeType`.

### 5.1.6. Abstrakte typer [GTD-7]

[GTD-7] Et OIOXML skema KAN benytte attributten *abstract* i simple og komplekse typedefinitioner.

Hvis modelleringen angiver en afledningsstruktur, f.eks. med en fælles datatype som ikke selv skal instantieres, kan man anvende attributten *abstract*.

### 5.1.7. Kontrol af typeafledninger [GTD-8]

[GTD-8] Et OIOXML skema BØR IKKE begrænse typeafledninger, dvs. attributterne *finalDefault* og *blockDefault* i rodelementet *schema* samt attributterne *block* og *final* i simple og komplekse typedefinitioner BØR IKKE benyttes.

Genbrug er centralt i OIOXML og derfor bør man ikke begrænse muligheden for typeafledninger.

### 5.1.8. Anvendelse af støttetyper [GTD-9] [GTD-10]

[GTD-9] Et OIOXML skema SKAL anvende dets støttetyper udelukkende som en hjælp til at etablere skemaets ene typedefinition, hvis en sådan defineres i skemaet og ikke kan udtrykkes i en enkelt definition.

[GTD-10] Et OIOXML skema SKAL definere dets støttetyper som *simple typer*.

Der er situationer, hvor det ikke er muligt at udtrykke typen for et OIOXML skema's elementerklæring udelukkende med én typedefinition. I disse tilfælde er det nødvendigt at definere en eller flere simple typer og genbruge disse i skemaets typedefinition. Sådanne typer til støtte for skemaets ene typedefinition betegnes *støttetyper*. Behovet opstår specielt i forbindelse med anvendelse af attributter. Følgende eksempel illustrerer anvendelsen af støttetyper.

Følgende skemauddrag erklærer et element *AText* med typen *string*:

```
<element name="AText" type="string"/>
```

Ønskes et attribut *myAttr* tilknyttet element *AText*, kan skemaet få følgende udseende:

```
<element name="AText" type="ATextType" />
<complexType name="ATextType">
  <simpleContent>
    <extension base="string">
      <attribute name="myAttr" type="string" />
    </extension>
  </simpleContent>
</complexType>
```

Bemærk at attributter skal være lokalt erklærede og derfor altid vil indgå direkte i en kompleks type. I ovenstående konstruktion er både elementets og attributtens type den simple indbyggede type *string* og det er ikke nødvendigt med støttetyper. Men hvis f.eks. elementets type var et *mønster* (via *pattern* konstruktionen) og/eller attributtens type f.eks. en enumereret kodeliste, så er det nødvendigt at omstrukturere skemaet og anvende støttetyper, idet informationsstrukturen ikke længere kan udtrykkes med kun én typedefinition. Følgende viser repræsentationen af begge disse udvidelser:

```
<element name="AText" type="ATextType" />
<complexType name="ATextType">
  <simpleContent>
    <extension base="_ATextType">
      <attribute name="myAttr" type="_myAttrType" />
    </extension>
  </simpleContent>
</complexType>
<!-- Støttetyper -->
<simpleType name="_ATextType">
  <restriction base="string">
    <pattern value="[0-9]{5}" />
  </restriction>
</simpleType>
<simpleType name="_myAttrType">
  <restriction base="string">
    <enumeration value="a" />
    <enumeration value="b" />
    <enumeration value="c" />
  </restriction>
</simpleType>
```

De to simple typer *\_ATextType* og *\_myAttrType* er nødvendige for at etablere skemaets typedefinition *ATextType* og er derfor reelt støttetyper. Se i øvrigt [TPN-4] for mere information om navngivning af støttetyper samt [OIO-6] og [OIO-7].

## 5.2. Regler for simple typedefinitioner

### 5.2.1. Anvendelse af *list* [STD-1]

[STD-1] Et OIOXML skema MÅ IKKE benytte konstruktionen *list*.

*list* muliggør repræsentation af multiple værdier af en bestemt type i selve den tekstuelle værdi af ét element eller én attribut. Dvs. et element eller attribut er ikke tilknyttet én entydig værdi, men flere mulige. Da entydighed er et væsentligt princip i OIOXML, tillades det ikke at anvende *list* konstruktionen.

### 5.2.2. Anvendelse af *union* [STD-2]

**[STD-2]** Et OIOXML skema MÅ IKKE benytte konstruktionen *union*.

*union* består af sammenlægning af to eller flere datatyper i samme element. Dette gør det dermed uklart, hvilken type der egentlig anvendes, og må derfor ikke benyttes.

Eksempel:

```
<xsd:element name='size'>
  <xsd:simpleType>
    <xsd:union>
      <xsd:simpleType>
        <xsd:restriction base='integer' />
      </xsd:simpleType>
      <xsd:simpleType>
        <xsd:restriction base='string' />
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>
</xsd:element>
```

I forhold til ovenstående eksempel vil det første “size” element herunder validere som *integer* og de to øvrige som *string*.

```
<size>1</size>
<size>large</size>
<size xsi:type='xsd:string'>1</size>
```

### 5.2.3. Længden af *string* [STD-3]

**[STD-3]** Et OIOXML skema BØR IKKE begrænse længden af den simple indbyggede type *string*, hvis der ikke eksisterer en fælles vedtagen længde.

Denne regel tillader kun, at man begrænser længden af *string* for en bestemt tekstinformation, hvis der eksisterer en reel årsag, såsom en lovbestemt årsag, et direktiv eller cirkulære, eller en fælles vedtagen strenglængde for denne information for et givent domæne (længden af *string* kan begrænses med facetterne *Length* og *maxLength*).

Denne regel skal befordre, at man, i stedet for at repræsentere en bestemt tekstinformation med forskellige strenglængder i OIOXML skemaer, løser problemet på et højere niveau enten ved, at man ikke begrænser tekststrengen eller endnu bedre, at man indenfor et domæne bliver enige om en fast strenglængde for en bestemt tekstinformation, så alle parter kan udveksle denne information uden problemer og uden tab af data..

Hvis ikke der er en fælles konsensus om længden af en bestemt tekstinformation og hvis hvert fagsystem definerer deres eget OIOXML skema med egen strenglængde for samme information, da kan situationen opstå, hvor et fagsystem via XML overfører en tekstinformation med f.eks. strenglængden 40 til et andet system, der kun kan acceptere strenglængden 30 for samme information. Enten kan overførslen ikke gennemføres eller også er konsekvensen tab af data.

### 5.2.4. Repræsentation af kodelister [STD-4]

**[STD-4]** Et OIOXML skema BØR udtrykke kodelister via *enumeration* konstruktionen.

Hvis der findes naturlige *name token* værdier, bør *enumeration* konstruktionen anvendes. Alternativt kan begrænsning af heltal, typisk positive, og mere sjældent *pattern* anvendes. Det er meningen, at det skal forblive en kode og en fuld tekstuel repræsentation.

Et eksempel er landekoden, hvor værdien næsten er selvsigende, modsat f.eks. et heltal, der typisk er blottet for semantik og blot identificerer en entydig værdi.

Se iverdigt dokumentationsreglen [DOC-6].

### 5.2.5. Værdier i *enumeration* [STD-5] [STD-6]

[STD-5] Et OIOXML skema BØR udelukkende udtrykke værdier i *enumeration* konstruktioner med små bogstaver.

[STD-6] Et OIOXML skema KAN udtrykke værdier i *enumeration* konstruktioner både på dansk eller engelsk.

Udgangspunktet bør være små bogstaver uden mellemrum med mindre, det er værdier, som reelt begynder med stort bogstav (f.eks. navneord).

I det omfang typen repræsenterer reelle dataværdier, bør man bruge dansk som sprog. Mere abstrakte koder bør benytte sig af engelsk.

Dersom der indenfor et domæne allerede er defineret et udfaldsrum, som falder udenfor ovenstående regel, kan der dispenseres. Et eksempel kunne være et udfaldsrum, der har været anvendt i EDIFACT-baserede meddelelser gennem en årrække. Hvis der er bred enighed blandt domæneeksperter om at bibeholde det eksisterende udfaldsrum, kan dette implementeres i OIOXML.

### 5.2.6. Anvendelse af *whitespace* facetten og dertil koblede typer [STD-7]

[STD-7] Et OIOXML skema MÅ IKKE benytte *whitespace* facetten samt de indbyggede simple typer *token* og *normalizedString*.

Anvendelse af *whitespace*, *token* og *normalizedString* har den konsekvens, at en XML instans' indhold ikke er entydig på forhånd, men kræver tilstedeværelsen af et skema for at kunne etablere dets fulde kanoniserede indhold. Af denne grund må disse konstruktioner ikke benyttes. En XML instans' indhold skal være entydig i sig selv. Se også samme problematik i [ELD-6], [ELD-7], [ATD-3] og [ATD-4].

## 5.3. Regler for komplekse typedefinitioner

### 5.3.1. Opbygning af komplekse typer [CTD-1] [CTD-2]

[CTD-1] Et OIOXML skema BØR definere en kompleks type ved anvendelse af konstruktionerne *sequence* og *choice*.

**[CTD-2]** Et OIOXML skema MÅ IKKE definere en kompleks type ved anvendelse af konstruktionen *all*.

Man bør definere nye komplekse typer ved direkte anvendelse af *sequence* og *choice*, da disse giver en forudsigelig struktur. Konstruktionen *all* derimod bruges på indholdsmodeller uden krav til rækkefølgen af elementerne, men stiller dog krav til kardinaliteten, idet hvert element enten skal forkomme præcis én gang eller være valgfrit (*minOccurs=0* og *maxOccurs=1*).

Konstruktionen *all* kan kun ligge øverst i indholdsmodellen for en sammensat type og kan kun indeholde elementerklæringer, ikke *choice* eller *sequence*. Af denne grund skal *sequence* og *choice* benyttes, da der ønskes en klar og explicit model. Risikoen for en non-deterministisk indholdsmodel fjernes og værktøjsunderstøttelsen sikres.

Bemærk at visse udviklingsværktøjer på nuværende tidspunkt ikke i stand til at konsumere WSDL'er, der anvender *choice* konstruktionen.

### 5.3.2. Definition af komplekse typer ved afledning [CTD-3] [CTD-4]

**[CTD-3]** Et OIOXML skema KAN definere en kompleks type via *extension* konstruktionen.

Ud fra et overordnede designprincip om at gøre tingene så simple som muligt, skal man prioritere at anvende aggregering via en sammensat type fremfor at benytte *extension*. Dog er det helt gyldigt at benytte *extension* i OIOXML skemaer. Valget afgøres normalt ud fra, hvorledes man ønsker at repræsentere ens datamodel i et skema.

**[CTD-4]** Et OIOXML skema MÅ IKKE definere en kompleks type via *restriction* konstruktionen.

Anvendelsen af *restriction* i komplekse typer er ikke tilladt pga. de komplikationer, der kan nemt opstå. Kombinationer af *restriction* med *extension* kan hurtigt skabe svært fortolkelige indholdsmodeller, hvilket modarbejder princippet i OIOXML, om at skemaer skal defineres så simpelt og forståeligt som muligt.

### 5.3.3. Anvendelse af blandet indholdsmodel [CTD-5]

**[CTD-5]** Et OIOXML skema KAN benytte den blandede indholdsmodel (mixed content) til strukturer, hvor tekst og elementer frit kan blandes.

Den blandede indholdsmodel tillader elementer at indeholde både tekst og andre elementer. Dette sker typisk i strukturer, der primært indeholder tekst fremfor data.

### 5.3.4. Anvendelse af tom indholdsmodel [CTD-6]

**[CTD-6]** Et OIOXML skema KAN benytte den tomme indholdsmodel (empty content) til strukturer, hvor tekst og elementer frit kan blandes.

Eksempler er f.eks. <br/> og <hr/> fra XHTML.

### 5.3.5. Anvendelse af *any* [CTD-7] [CTD-8] [CTD-9]

**[CTD-7]** Et OIOXML skema KAN benytte *any* konstruktionen til at binde to relaterede, og eventuelt uafhængige, OIOXML dialekter sammen.

En *OIOXML dialekt* repræsenterer en del af et samlet XML sprog og vil typisk relatere til en bestemt funktionsgruppe eller klart afgrænset informationsmængde i et helt XML sprog, som kan stå for sig selv. En OIOXML dialekt kan i sin yderste konsekvens repræsentere et helt XML sprog, f.eks. en hel grænseflade i en OIOXML skemaaflevering eller et helt adopteret OIOXML sprog såsom f.eks. XHTML. Uanset omfanget af en OIOXML dialekt, vil dialekten *aldrig* være repræsenteret af kun ét XML element, men *altid* af et større eller mindre antal elementer, som er strukturelt og logisk forbundet. Uanset antallet af elementer og deres struktur i dialekten, vil det være fælles for alle elementer i en OIOXML dialekt, at de alle er erklærede i godkendte OIOXML skemaer.

Fordelen ved at anvende *any* konstruktionen i forbindelse med OIOXML dialekter er, at man forholdsvis nemt kan koble to eller flere måske uafhængige gruppers arbejde sammen for at skabe en større sammenhæng. Disse grupper har som antydning ikke nødvendigvis kontakt til hinanden, men kan selvfølgelig have det, idet deres samarbejde kan være alt ligefra svagt koblet til stærkt koblet. OIOXML dialekter kan f.eks. være udviklet af forskellige nationale og internationale grupper; internationale OIOXML dialekter vil typisk være adopterede i OIOXML.

Enhver anden anvendelse af *any* konstruktionen frarådes direkte eller forbydes (se regel [CTD-8]).

**[CTD-8]** Et OIOXML skema MÅ IKKE benytte *any* konstruktionen til at skabe en bevidst svag indholdsmodel indenfor samme OIOXML dialekt eller til generelt at skabe en uforpligtende overgang til et eller flere elementer, som enten er statisk kendt på forhånd eller som ikke reelt repræsenterer en selvstændig OIOXML dialekt.

Denne regel forbyder enhver anvendelse af *any* konstruktionen, som ikke overholder [CTD-7] og ikke lever op til kravene til en OIOXML dialekt. Vigtigst er det, at *any* konstruktionen ikke må anvendes til direkte at skabe en svag indholdsmodel (dette strider mod [GTD-1]) for at opnå en bestemt effekt.

**[CTD-9]** Et OIOXML skema SKAL i en *any* konstruktion sætte værdien af attributten *processContents* til "strict".

Det er vigtigt, at der i forbindelse med anvendelsen af *any* konstruktioner i OIOXML skemaer sikres, at hele indholdet af en XML instans, der indeholder to eller flere OIOXML dialekter koblet sammen med *any* konstruktioner, altid valideres fuldstændigt.

Der sættes ingen krav til anvendelsen af *namespace* attributten i *any* konstruktionen, men det anbefales, at man så vidt muligt altid specificerer det namespace eller de namespaces, som *any* konstruktionen benytter sig af og undgår *namespace* værdier som "##any" og "##other".

### 5.3.6. Anvendelse af *anyAttribute* [CTD-10]

[CTD-10] Et OIOXML skema MÅ IKKE benytte *anyAttribute* konstruktionen i indholdsmodellen for en kompleks type.

Da attributter kun anvendes til metadata og der ønskes en klar indholdsmodel, må man ikke benytte *anyAttribute*.

### 5.3.7. Komplekse typer i Kerneschemaer [CTD-11]

[CTD-11] Et OIOXML skema tilhørende *Kerneklassen (OIOXML Core Class)* SKAL kun genbruge eksisterende OIOXML Kerneelementer i indholdsmodellen for dets komplekse typedefinition.

Denne regel sikrer, at en sammensat (Kerne)type i et Kerneschema altid selv vil være opbygget af Kerneelementer. Dermed bevares en 100% kernestatus for både den komplekse Kernetype og dens indholdsmodel.



---

## Kapitel 6. Regler for element- og attributerklæringer

### 6.1. Regler for elementerklæringer

#### 6.1.1. Globale elementerklæringer [ELD-1] [ELD-2]

[ELD-1] Et OIOXML skema tilhørende *Kerneklassen (OIOXML Core Class)* eller *Domæneklassen (OIOXML Domain Class)* SKAL erklære dets element globalt.

[ELD-2] Et OIOXML skema tilhørende *NDR-klassen (OIOXML NDR Class)* BØR erklære dets element globalt.

Disse regler er med til at sikre så høj en grad af genbrug som muligt. OIOXML skemaerne gøres mere overskuelige med en logisk fladere struktur, idet lokalt erklærede elementer altid skal implementeres som en reference til et globalt erklæret element.

Se også [OIO-1], [OIO-2], [GTD-2] og [GTD-3] for mere information om genbrug.

#### 6.1.2. Namespace for elementer [ELD-3]

[ELD-3] Et OIOXML skema SKAL tildele et namespace til dets element, dvs. attributten *elementFormDefault* i rodelementet *schema* SKAL tildeles værdien *qualified* og attributten *form* i elementerklæringer MÅ IKKE benyttes.

Da alle OIOXML skemaer skal være tilknyttet et namespace og elementer erklæres globalt, skal der bruges kvalificerede elementer i instanser. Dette er også en fordel i forhold til erklæring af default namespaces i instanser.

Eksempel:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema elementFormDefault="qualified" ... >
...
</schema>
```

#### 6.1.3. Anvendelse af substitution groups [ELD-4]

[ELD-4] Et OIOXML skema MÅ IKKE benytte substitution groups i dets elementerklæring.

For at opretholde entydighed i anvendelsen af elementer, er det ikke tilladt at anvende substitution groups. Hvis anvendelsen af alternative elementer ønskes i en bestemt sammenhæng, skal dette helt og holdent specificeres i indholdsmodellen for den relevante type og ikke gennem substitution groups.

#### 6.1.4. Anvendelse af *nillable* [ELD-5]

XML Schema indeholder en facilitet som tillader, at man kan repræsentere valgfri XML information ved hjælp af null-værdier. Med en boolsk værdi (*true* eller *false*) styrer attributten *nillable* i *element* konstruktionen, hvorvidt det er tilladt at angive en null-værdi for det pågældende element i XML instansen. Med andre ord tillades det, at angivelse af indhold for et element i XML-instansen helt kan undlades, selvom elementets indholdsmodel kræver et indhold. Faciliteten kan anvendes, uanset om elementets indholdsmodel kræver ren tekst (via en simpel type) eller andre underelementer (via en kompleks type).

Det frarådes helt at anvende denne konstruktion indenfor rammerne af OIOXML. For det første tillader XML Schema allerede repræsentation af valgfri XML information via indholdsmodellen ved at anvende "minOccurs=0" i en elementerklæring. For det andet skaber anvendelsen af *nillable* nemt skemaer, som uheldigvis ligger sig meget tæt opad den tilknyttede applikation. Denne applikationsnærhed eller -afhængighed er uhensigtsmæssig netop for OIOXML skemaer, som er underlagt langsigtede krav om fælles standardisering og størst mulig genbrugelighed på tværs af fagsystemer.

Derudover henvises også til det gennemgående designprincip om, at OIOXML skemaer skal skabes med entydighed og enkelthed for øje samt med mindst mulig kompleksitet (se [OIO-12]). *Nillable*, anvendt til repræsentation af valgfri XML information, er et langt mere komplekst alternativ (endda med variabel værktøjsunderstøttelse) til den (selv i XML DTD'er) simple indbyggede mulighed for specifikation af valgfrie elementer via indholdsmodellen.

De følgende 3 regler, der af ovennævnte grunde forbyder eller stærkt fraråder anvendelsen af *nillable*, skal opfattes og anvendes sammen som en helhed.

**[ELD-5]** Et OIOXML skema tilhørende *Kerneklassen (OIOXML Core Class)* eller *Domæneklassen (OIOXML Domain Class)* MÅ IKKE sætte *nillable* attributten til værdien *true* i dets elementerklæring.

Anvendelsen af *nillable* attributten forbydes helt for Kerne- og Domæneelementer, idet netop disse er specielt udvalgte og godkendte med henblik på genbrug og derfor skal være så applikationsneutrale som overhovedet muligt.

**[ELD-6]** Et OIOXML skema MÅ IKKE sætte *nillable* attributten til værdien *true* i dets elementerklæring, hvis det direkte eller indirekte genbruger en *Kernetype* eller *Domænetype*.

Det forbydes tillige at anvende *nillable* for elementerklæringer, der *genbruger* en Kerne- eller Domænetype enten direkte eller indirekte, f.eks. ved nedarvning. Anvendelsen af *nillable* signalerer en karakterændring af et eksisterende Kerne- eller Domæneelement, som er vigtigt at undgå på grund af disses skemaers specielle genbrugsstatus. Det er vigtigt, at elementer og typer i Kerne- og Domæneskemaer altid fremstår entydige og stabile i alle sammenhænge og anvendelser.

**[ELD-7]** Et OIOXML skema tilhørende *NDR-klassen (OIOXML NDR Class)* BØR IKKE sætte *nillable* attributten til værdien *true* i dets elementerklæring og i givet fald kun hvis elementets type er en simpel type eller hvis elementets type er en kompleks type, der udelukkende indeholder attributerklæringer.

Derudover frarådes det stærkt at anvende *nillable* i NDR-skemaer generelt. Hvis konstruktionen anvendes, må det kun ske for et element, der udelukkende indeholder en tekstværdi og ikke for et element, hvis indholdsmodel tillader yderligere underelementer. Attributter er tilladt for elementet.

### 6.1.5. Default- og konstantværdier [ELD-8] [ELD-9]

[ELD-8] Et OIOXML skema MÅ IKKE benytte attributten *default* i dets elementerklæring.

[ELD-9] Et OIOXML skema MÅ IKKE benytte attributten *fixed* i dets elementerklæring.

Problemet ved at benytte *default* og *fixed* er, at en XML Schema processor vil implicit tilføje værdierne til de tomme elementer og dermed skabe en intern repræsentation, som ikke er identisk med den oprindelige XML instans. Konsekvensen er, at XML instansen ikke er entydig på forhånd, men kræver tilstedeværelsen af et skema for at etablere dets fulde kanoniserede indhold. OIOXML skemaer skal udformes således, at de tilsvarende XML instanser altid er entydige på egen hånd, hvilket betyder, at *default* og *fixed* ikke må anvendes.

## 6.2. Regler for attributerklæringer

### 6.2.1. Anvendelsen af attributter [ATD-1]

[ATD-1] Et OIOXML skema SKAL benytte dets elements attributter til metadata.

Attributter skal reserveres til metadata, dvs. data som yderligere beskriver hoveddata. Elementet som attributterne er tilknyttet, rummer de egentlige hoveddata. Dette giver en simpel indholdsmodel og giver ingen problemer med, at attributter kun kan være baseret på simple typer.

I følgende eksempler er det på sin plads at anvende attributter, fordi attributterne kvalificerer elementerne ved f.eks. at angive en enhed:

```
<weight measureUnit="kg">230</weight>
```

### 6.2.2. Attributter, genbrug og namespaces [ATD-2] [ATD-3]

Det er ikke forventningen, at attributter vil være genstand for genbrug, da deres anvendelse udelukkende er til metadata, der kvalificerer et element og dets indhold. Dette svarer til den normale anvendelse og ligger i tråd med generelle anbefalinger internationalt. Af denne grund skal alle attributter i OIOXML skemaer være lokalt erklærede samt ikke tilhøre et namespace. Disse krav er præciseret i følgende regler.

[ATD-2] Et OIOXML skema SKAL erklære alle dets attributter lokalt.

Ved at fjerne muligheden for at erklære en attribut globalt, kan attributter ikke genbruges af andre skemaer.

[ATD-3] Et OIOXML skema MÅ IKKE tildele et namespace til dets attributter, dvs. attributten *attributeFormDefault* i rodelementet *schema* SKAL tildeles værdien *unqualified*, eller slet ikke specificeres (*unqualified* er defaultværdien), og attributten *form* i attributerklæringer MÅ IKKE benyttes.

Herved fjernes muligheden helt for at tildele et namespace til en lokal attributerklæring. Attributter eksisterer kun i kontekst af et element og har derfor ikke behov for et namespace.

### Eksempel:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema attributeFormDefault="unqualified" ... >
...
</schema>
```

### 6.2.3. Default- og konstantværdier [ATD-4] [ATD-5]

[ATD-4] Et OIOXML skema MÅ IKKE benytte attributten *default* i dets attributerklæringer.

[ATD-5] Et OIOXML skema MÅ IKKE benytte attributten *fixed* i dets attributerklæringer.

Problemstillingen er den samme som i elementer (se [ELD-6] og [ELD-7]). Skemaet laver om på indholdet efter validering, dvs. XML instansen er ikke entydig.

---

## Kapitel 7. Versionerings- og namespaceregler

### 7.1. Namespaceversionering [VER-1]

[VER-1] Et OIOXML skema SKAL angive dets version ved hjælp af et namespace.

OIOXML skemaer vil altid være udsat for ændringer og derfor er det nødvendigt med en entydig og klar versioneringsstrategi, der nemt kan vedligeholdes. Denne strategi er derfor baseret på versionering gennem namespaces, da dette tillader bevarelsen af det lokale navn, mens det kvalificerede navn opdateres.

### 7.2. Låsning af et OIOXML skema [VER-2]

[VER-2] Et OIOXML skema, som er godkendt og optaget i en af OIOXML klasserne, MÅ IKKE ændres.

For ikke at skabe tvetydighed eller usikkerhed om versioner, kan man ikke ændre et skema i Infostrukturbasen som er godkendt. Dette er rent praktisk, da man derved er sikker på at få det samme indhold hver gang, man bruger et godkendt skema.

### 7.3. Bagudkompatible udvidelser [VER-3]

[VER-3] Et OIOXML skema KAN benytte et eksisterende namespace ved bagudkompatible udvidelser.

Hvis man har et større projekt, som er opdelt i flere faser, kan man vælge at benytte det samme namespace til alle faser, hvis ellers der er tale om udvidelser, der ikke forstyrrer de allerede etablerede skemaer. Man skal altså ikke erklære et nyt namespace, hver gang man har nogle nye skemaer, men modsat bør skemaer, der deler namespace have en sammenhæng.

### 7.4. Anvendelse af attributten *version* [VER-4]

[VER-4] Et OIOXML skema KAN frit benytte attributten *version* i rodelementet *schema*.

XML Schema anbefalingen har kun i ringe omfang taget højde for versioneringsproblematikken. I *schema* rodelementet er der en attribut *version*, men standarden understøtter ikke, at man refererer til en bestemt version af et skema indenfor et givet namespace eller instansdokument.

Da versionering foregår gennem erklæring af namespace og metadata, kan attributten højest anvendes vejledende i forhold til både andre skemaer og instansdokumenter. Dvs. der er ingen specifikke regler for anvendelse og det er helt op til skemaudvikleren, hvorledes man vil benytte denne attribut.

### 7.5. Namespace repræsentation [NMS-1]

[NMS-1] Et OIOXML skema SKAL anvende et namespace, som repræsenterer en gyldig URL i Infostrukturbasen og har følgende opbygning: "http://rep.oio.dk/<internetdomæne>/xml/schemas/<YYYY>/<MM>/<DD>/".

Filosofien ved den valgte namespace navngivningsstrategi er, at namespaceet herved både er unikt globalt, rummer versionering samt er en gyldig URL, der ved opslag i en browser lister alle de skemaer, der er erklæret under det givne namespace.

Eksempelvis har undervisningsministeriet erklæret følgende namespace <http://rep.oio.dk/uvm.dk/xml/schemas/2003/06/23/>

Man bør ikke oprette namespaces, som er baseret på en fremtidig dato.

## 7.6. Navngivning af prefix i namespaceerklæringer [NMS-2]

[NMS-2] Et OIOXML skema SKAL basere sit namespace-prefix på navnet af internetdomænet i namespaceet uden punktum og roddomæne.

Sammenhæng mellem prefix og namespace giver læsbarhed, da alle elementer er tilknyttet et namespace, og derfor skal ses som et samlet hele.

Hvis man har et meget langt domænenavn skal man anvende en forkortelse, men typisk vælger man også relativt korte domænenavne for at gøre dem lette at huske, hvilket passer godt ind i dette formål. Det er desuden værd at bemærke at et namespace prefix ikke må begynde med bogstavsækkefølgen *xml*, da denne bogstavsekvens er reserveret af XML-standardens.

Eksempelvis har undervisningsministeriet domænet *uvm.dk* og deres prefix bliver derfor *uvm*

Hvis man anvender to namespaces som tilhører samme internetdomæne skal man vælge en strategi der adskiller dem, som f.eks. *uvm1/uvm2*, *uvm-a/uvm-b* eller *uvm-230623/uvm-20040311*

---

## Kapitel 8. Dokumentation og metadata

### 8.1. Dokumentation

I OIOXML sammenhæng er grundig dokumentation essentiel, da skemaerne skal formidles til mange forskellige parter. Skemaerne skal sagsbehandles i XML sekretariatet, evt. i høring hos andre interessenter, indgå som underskemaer i fremtidige skemaer og endeligt bruges af forskellige softwareudviklere. Derudover er det valgte sprog (engelsk eller dansk) for dokumentationen vigtig.

Det anbefales, at der generelt dokumenteres på engelsk. Hvis det er u hensigtsmæssigt af administrative, semantiske, eller af andre væsentlige hensyn, kan der i stedet anvendes dansk dokumentation.

Engelsk dokumentation anbefales, bl.a. fordi skemaer direkte kan indgå i internationale sammenhænge eller softwareudviklere ansat i (internationale eller danske) firmaer placeret i Danmark ofte er specialister fra udlandet med varierende kendskab til dansk. Det er heller ikke ualmindeligt, at udviklingsafdelingen i et internationalt softwarefirma i Danmark er placeret i udlandet.

Det er meget vigtigt at dokumentere OIOXML skemaer parallelt med, at disse udvikles. Et XML skema er ren syntaks for sammensætningen af tags eller elementer i en datastruktur. Selvom navngivningen tilstræbes at gøres meningsfuld, skal skemaets fulde semantik eller forretningslogik, dvs. meningen og anvendelsen af skemaet altid dokumenteres.

#### 8.1.1. Angivelse af dokumentation i metadata [DOC-1]

**[DOC-1]** Et OIOXML skema SKAL placere al dets dokumentation i skemaets metadata.

Ved at placere hele dokumentationen for et OIOXML skema i skemaets metadata (i modsætning til at dokumentationen placeres i selve skemaet) kan dokumentationen opdateres/forældes samtidig med, at skemaet forbliver uændret. Dette er muligt, fordi et OIOXML skema og dets metadata er repræsenteret i hver sin fil (se i publikationen “*OIO-7: OIOXML Metadata – Definitioner*” for mere information om metadata).

Følgende regler beskriver de krav, som stilles til de metadata-elementer, der er reserveret til skemadokumentation. Der anvendes 3 specifikke metadata-elementer med det formål at dokumentere funktionen af et OIOXML skema, nemlig: *Title*, *DescriptionEnglish* og *DescriptionDanish*. Derudover kan der suppleres med yderligere oplysninger i *LabelEnglish* og *LabelDanish* i forbindelse med brugervenlige termer for et skema.

##### 8.1.1.1. *Title* [DOC-2]

**[DOC-2]** Et OIOXML skema SKAL udfylde metadata-elementet *Title* med en overordnet titel for skemaet.

Denne titel er typisk navnet på rodelementet eller evt. den mest betydende type i skemaet.

### 8.1.1.2. *DescriptionEnglish* og *DescriptionDanish* [DOC-3] [DOC-4]

[DOC-3] Hvis metadata-elementet *DescriptionEnglish* udfyldes, SKAL metadata-elementet indeholde en udførlig beskrivelse af skemaets funktion på *engelsk*.

[DOC-4] Hvis metadata-elementet *DescriptionDanish* udfyldes, SKAL metadata-elementet indeholde en udførlig beskrivelse af skemaets funktion på *dansk*.

Med beskrivelse menes en fyldestgørende redegørelse for den information, skemaet repræsenterer samt den funktion skemaet skal udføre med denne information. Beskrivelsen skal være tilstrækkelig uddybende, så en relativ udenforstående kan danne sig en ide om skemaets anvendelse og en tekniker kan få et præcist billede af skemaets funktion i forhold til en reel implementering af skemaet i en løsning.

Beskrivelserne i *DescriptionEnglish* og *DescriptionDanish* skal altid være specificeret i det respektive sprog, henholdsvis engelsk og dansk.

Hvilke af metadata-elementerne *DescriptionEnglish* og *DescriptionDanish*, som skal anvendes for forskellige typer OIOXML skemaer, er angivet i de følgende regler [DOC-5] og [DOC-6].

#### 8.1.2. Dokumentation for OIOXML klasserne

##### 8.1.2.1. Dokumentation for Kerneklassen [DOC-5]

[DOC-5] Et OIOXML skema tilhørende *Kerneklassen (OIOXML Core Class)* SKAL udfylde metadata-elementerne *DescriptionEnglish* og *DescriptionDanish*.

Det gælder specielt for et Kerneschema, at det skal beskrives både på engelsk og dansk. I kraft af, at et Kerneschema er godkendt til bredest muligt genbrug, er det vigtigt, at både en engelsk og dansk beskrivelse foreligger for ikke alene at fremme forståelsen af skemaets funktion i både en dansk og international sammenhæng, men også for at imødekomme softwareudviklere af enhver nationalitet.

##### 8.1.2.2. Dokumentation for Domæne- og NDR-klassen [DOC-6]

[DOC-6] Et OIOXML skema tilhørende *Domæneklassen (OIOXML Domain Class)* eller *NDR-klassen (OIOXML NDR Class)* SKAL udfylde enten metadata-elementet *DescriptionEnglish* eller metadata-elementet *DescriptionDanish*, og KAN udfylde begge metadata-elementer, hvilket anbefales.

Et Domæne- eller NDR-skema skal altid have en beskrivelse af dets funktion, men i



modsatning til et Kerneschema behøver beskrivelsen kun at foreligge enten på dansk eller engelsk. Det anbefales dog, at beskrivelsen foreligger på begge sprog af samme årsager, som angivet i [DOC-5].

### 8.1.3. Anvendelse af Labels metadata

#### 8.1.3.1. Angivelse af brugervenlig term [DOC-7]

Man kan frivilligt angive metadata-elementerne *LabelEnglish* og *LabelDanish* for et OIOXML skema til at specificere en brugervenlig term på dansk og/eller engelsk, som bedre egner sig til anvendelse og præsentation i en brugergrænseflade end det reelle navn anvendt i skemaets komponenter (element- eller typenavnet). Dette er et tilbud, som skemaudviklerne sammen med applikationsudviklerne kan benytte sig af til at bygge bedre grænseflader.

Følgende regler præciserer anvendelsen af Labels i metadata.

**[DOC-7]** Et OIOXML skema KAN udfylde metadata-elementerne *LabelEnglish* og/eller *LabelDanish* til at angive en brugervenlig term, som egner sig til præsentation i en brugergrænseflade.

Man kan vælge at angive enten én af metadata-elementerne *LabelEnglish* og *LabelDanish* eller begge to helt uafhængigt af sproget for skemaet (se i øvrigt Afsnit 4.2, "Engelsk/dansk sprogetvalg i navngivning" om sprogetvalg).

#### 8.1.3.2. *LabelEnglish* og *LabelDanish* [DOC-8] [DOC-9]

**[DOC-8]** Hvis metadata-elementet *LabelEnglish* udfyldes, SKAL termen være på *engelsk*.

**[DOC-9]** Hvis metadata-elementet *LabelDanish* udfyldes, SKAL termen være på *dansk*.

Beskrivelserne i *LabelEnglish* og *LabelDanish* skal altid være specificeret i det respektive sprog, henholdsvis engelsk og dansk.

Et eksempel på en anvendelse af *LabelEnglish* og *LabelDanish* er vist forneden.

Uddrag af skema:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" ...>
...
  <element name="TelephoneSubscriberIdentifier" ... />
...
</schema>
```

Skemaet erklærer et element, som har et langt teknisk navn, der repræsenterer et telefonnummer. Et eksempel på Labels metadata for dette skema kunne så være:

Uddrag af metadata for ovenstående skema:

```
<?xml version="1.0" encoding="UTF-8"?>
<Metadata>
...
<LabelDanish>Telefonnummer</LabelDanish>
<LabelEnglish>Telephone number</LabelEnglish>
...
</Metadata>
```

Den danske og engelske brugervenlige term, henholdsvis “Telefonnummer” og “Telephone number” kan så anvendes i en brugergrænseflade i stedet for det mere omstændelige, men korrekte, elementnavn “TelephoneSubscriberIdentifier”.

### 8.1.4. Anvendelse af *appinfo* [DOC-10]

**[DOC-10]** Et OIOXML skema MÅ IKKE benytte *appinfo* konstruktionen.

XML skemaer benyttes til beskrivelse og validering af data og skal derfor ikke indeholde applikationsspecifikke informationer

### 8.1.5. Dokumentation af kodeliste [DOC-11]

**[DOC-11]** Et OIOXML skema SKAL dokumentere alle værdierne i en kodeliste enten lokalt eller med en reference til et andet fyldestgørende dokument i Infostrukturbasen.

En kodeliste er en sekvens af individuelle værdier, som anvendes i en bestemt sammenhæng. En kodeliste kunne f.eks. beskrive forskellige mulige værdier for en egenskab. F.eks. hvis egenskaben er *farve* kunne kodelisten være: “rød”, “grøn”, “blå”, “gul”, etc.. Det kunne også være en liste af heltal, hvor hvert tal er en kode med en bestemt betydning. Specielt for kodelister opbygget af heltal er dokumentation af hver værdi obligatorisk for forståelsen. Se iøvrigt også [STD-4].

### 8.1.6. Afleveringsdokument [DOC-12]

**[DOC-12]** En OIOXML aflevering SKAL indeholde afleveringsdokumentation, der overordnet beskriver afleveringens skemaer.

Se "*OIO-5: OIOXML Skemaaflevering - Definitioner*" angående de specifikke detaljer i afleveringsdokumentationen.

## 8.2. Metadata

Metadata er data om data – dvs. forskellige oplysninger, der beriger et OIOXML skema med forskellige typer af information i form af en struktureret beskrivelse af skemaets indhold, type og form. Tilføjelsen af metadata giver slutbrugeren ekstra oplysninger samt gør skemaet lettere at genfinde.

En stor mængde skemaer øger behovet for at anvende metadata med henblik på at lette genfindingen af skemaerne. Da én af ideerne med Infostrukturbasen er udveksling af data mellem institutionerne, er det vigtigt, at man anvender metadata-elementerne.

### 8.2.1. Metadata for OIOXML skemaer [MTA-1]

**[MTA-1]** Et OIOXML skema SKAL have tilknyttet metadata.

Metadata for et OIOXML skema indeholder udover skemaets dokumentation (se [DOC-1]) også andre informationer, som er centrale for skemaets anvendelse indenfor OIOXML paradigmet. Derfor skal et OIOXML skema have tilknyttet metadata.

Metadata for et OIOXML skema er repræsenteret adskilt fra skemaet i en separat metadata-fil. Der er en klar entydig forbindelse mellem et OIOXML skema og dets metadata-fil. Har man en reference til et OIOXML skema, vil man altid entydigt kunne etablere en reference til dets metadata-fil og omvendt.

Der henvises til publikationen "*OIO-7: OIOXML Metadata – Definitioner*" for mere information om anvendelse og opbygning af OIOXML metadata samt en liste over eksisterende metadata-elementer.

## Appendiks A. OIOXML NDR Quick Reference

Regel kode	Regeltitel	Regeltekst
<b>OIOXML regler</b>		
OIO-1	Genbrug af eksisterende OIOXML elementer og typer	Et OIOXML skema SKAL genbruge eksisterende elementer eller typer fra <i>Kerneklassen (OIOXML Core Class)</i> og <i>Domæneklassen (OIOXML Domain Class)</i> .
OIO-2	Genbrug af eksisterende OIOXML elementer og typer	Et OIOXML skema BØR genbruge eksisterende elementer eller typer fra <i>NDR-klassen (OIOXML NDR Class)</i> .
OIO-3	Genbrug af de indbyggede simple typer	Et OIOXML skema SKAL genbruge de indbyggede simple typer i XML Schema frem for at definere egne typer til at repræsentere den samme information.
OIO-4	Genbrug af element fremfor type	Et OIOXML skema BØR genbruge et element frem for dets type, hvis elementets navn og anvendelse er entydig i den konkrete sammenhæng.
OIO-5	Genbrug af seneste skemaversion	Et OIOXML skema SKAL genbruge nyeste version af et andet eksisterende OIOXML skema, hvis dette andet skema forekommer i flere versioner.
OIO-6	Et OIOXML skema's indhold	Et OIOXML skema tilhørende <i>Kerneklassen (OIOXML Core Class)</i> eller <i>Domæneklassen (OIOXML Domain Class)</i> SKAL indeholde én elementerklæring og, hvis elementerklæringen ikke genbruger en eksisterende type fra et andet skema, én typedefinition for elementet samt eventuelt en eller flere definitioner af <i>støttetyper</i> nødvendige for etablering af elementets typedefinition.
OIO-7	Et OIOXML skema's indhold	Et OIOXML skema tilhørende <i>NDR-klassen (OIOXML NDR Class)</i> BØR indeholde én elementerklæring og, hvis elementerklæringen ikke genbruger en eksisterende type fra et andet skema, én typedefinition for elementet samt eventuelt en eller flere definitioner af <i>støttetyper</i> nødvendige for etablering af elementets typedefinition.
OIO-8	Skemareferencer	En OIOXML skemaaflevering SKAL i dets enkelte skemaer altid referere enten til godkendte OIOXML skemaer i Infostrukturbasen eller til skemaer indeholdt i selve skemaafleveringen.
OIO-9	OIOXML skemaer i Infostrukturbasen	Et OIOXML skema SKAL placeres i Infostrukturbasen.
OIO-10	Systemuafhængige OIOXML skemaer	Et OIOXML skema tilhørende <i>Kerneklassen (OIOXML Core Class)</i> eller <i>Domæneklassen (OIOXML Domain Class)</i> MÅ IKKE i dets udformning være influeret af begrænsninger i et bagvedliggende fagsystem.

OIOXML NDR Quick  
Reference

Regel kode	Regeltitel	Regeltekst
<b>OIOXML regler</b>		
OIO-11	Systemuafhængige OIOXML skemaer	Et OIOXML skema tilhørende <i>NDR-klassen (OIOXML NDR Class)</i> BØR IKKE i dets udformning være influeret af begrænsninger i et bagvedliggende fagsystem.
OIO-12	Klare og entydige skemaer	Et OIOXML skema SKAL designes så enkelt og entydigt som overhovedet muligt uden unødvendig kompleksitet og overflødige konstruktioner.

Regel kode	Regeltitel	Regeltekst
<b>Generelle XML Skema regler</b>		
GXS-1	Valg af skemasprog	Et OIOXML skema SKAL defineres i overensstemmelse med W3C XML Schema anbefalingen (version 1.0) af 2. maj 2001: <i>XML Schema Part 1: Structures</i> og <i>XML Schema Part 2: Datatypes</i> .
GXS-2	Version af XML	Et OIOXML skema SKAL anvende version 1.0 af W3C XML anbefalingen af 4. februar 2004 <i>Extensible Markup Language (XML) 1.0 (Third Edition)</i> .
GXS-3	Valg af encoding scheme	Et OIOXML skema SKAL anvende UTF-8 som encoding scheme.
GXS-4	Tilknytning til namespace	Et OIOXML skema SKAL tilknyttes et namespace.
GXS-5	Skemareferencer	Et OIOXML skema MÅ IKKE benytte <i>import</i> konstruktionen til at referere til et andet OIOXML skema, der anvender samme namespace, som skemaet der refereres fra.
GXS-6	Skemareferencer	Et OIOXML skema MÅ IKKE benytte <i>redefine</i> konstruktionen.
GXS-7	Anvendelse af notation	Et OIOXML skema MÅ IKKE benytte <i>notation</i> konstruktionen.
GXS-8	Anvendelse af schemaLocation	Et OIOXML skema SKAL angive alle dets <i>schemaLocation</i> attributter med en absolut og gyldig URL til det refererede OIOXML skemas placering i Infostrukturbasen.

OIOXML NDR Quick  
Reference

Regel kode	Regeltitel	Regeltekst
<b>Navngivning</b>		
<b>Generelle navngivningsregler</b>		
GNR-1	Unik navngivning	Et OIOXML skema SKAL navngive alle dets globalt erklærede komponenter (elementer, og typer) unikt indenfor skemaets namespace og BØR navngive unikt på tværs af alle namespaces for godkendte OIOXML skemaer.
GNR-2	Navngivningsmodellen for elementer, attributter og typer	Et OIOXML skema BØR navngive alle dets globalt og lokalt erklærede komponenter (elementer, attributter samt typer) efter <i>ObjektEgenskabRepræsentation</i> navngivningsmodellen, som specificeret i følgende underregler.
GNR-2a	Termen Objekt	Et navn SKAL i dets <i>Objekt</i> term beskrive det dataobjekt, som et element og dets type repræsenterer i en bestemt sammenhæng.
GNR-2b	Termen Objekt	Et navn KAN udelade dets <i>Objekt</i> term i det tilfælde, hvor et element og dets type optræder i en kontekst af et objekt eller objektet er ukendt.
GNR-2c	Termen Egenskab	Et navn SKAL i dets <i>Egenskab</i> term ved hjælp af en eller flere kvalificerende ord beskrive en fremtrædende egenskab ved et elements og dets types <i>Objekt</i> term.
GNR-2d	Termen Repræsentation	Et navn SKAL i dets <i>Repræsentation</i> term beskrive et elements og dets types repræsentative kategori og SKAL antage en af værdierne i OIOXML's liste over repræsentationstermer.
GNR-2e	Identiske eller enslydende fraser	Et navn SKAL, hvis det har en frase i termen <i>Egenskab</i> , som er synonymt med en frase i termen <i>Repræsentation</i> , fjerne frasen fra <i>Egenskab</i> og bibeholde den i <i>Repræsentation</i> .
GNR-2f	Entalsform af navne	Et navn SKAL angives på entalsform, med mindre navneordet er en flertalsform.
GNR-2g	Forkortelser og akronymer	Et navn BØR IKKE benytte forkortelser og akronymer.
GNR-2h	Bindeord med mere	Et navn SKAL opbygges af udsagnsord, navneord og tillægsord.
GNR-2i	Anvendelse af tegn i navne	Et navn MÅ IKKE i dets opbygning benytte underscore (_), punktum (.) og bindestreg (-).
<b>Engelsk/dansk sprogvvalg i navngivning</b>		
<b>Sprogvvalg for et OIOXML skema</b>		
LNR-1	Engelsk og dansk OIOXML skema	Et OIOXML skema SKAL navngive alle dets globalt og lokalt erklærede komponenter (elementer, attributter

OIOXML NDR Quick  
Reference

Regel kode	Regeltitel	Regeltekst
<b>Navngivning</b>		
		samt typer) i ét og samme sprog. Et OIOXML skema med engelsk navngivne komponenter betegnes et <i>engelsk</i> OIOXML skema og et OIOXML skema med dansk navngivne komponenter betegnes et <i>dansk</i> OIOXML skema.
LNR-2	Sprogangivelse med <code>xml:lang</code>	Et dansk OIOXML skema SKAL tildele attributten <code>xml:lang</code> i rodelementet <i>schema</i> værdien DA. Et engelsk OIOXML skema KAN tildele attributten <code>xml:lang</code> i rodelementet <i>schema</i> værdien EN. Attributten <code>xml:lang</code> MÅ IKKE antage andre værdier end DA og EN og hvis attributten ikke er specificeret, antages værdien at være EN, dvs. skemaet antages at være et engelsk OIOXML skema.
LNR-3	Ord- og fagbøger for et sprog	Navngivning på engelsk BØR ske i overensstemmelse med <i>Oxford English Dictionary</i> eller relevante fagbøger.
LNR-4	Ord- og fagbøger for et sprog	Navngivning på dansk BØR ske i overensstemmelse med <i>Dansk Retskrivningsordbog</i> eller relevante fagbøger.
LNR-5	Anvendelse af Æ, Ø og Å	Et dansk OIOXML skema må ikke anvende de danske specialtegn <i>æ</i> , <i>ø</i> og <i>å</i> på grund af problemer i forbindelse med manglende værktøjsunderstøttelse.
<b>Sprogvalg i relation til OIOXML klasser navngivningsprogvalg for OIOXML klasser</b>		
LNR-6	Sprogvalg i Kerneklassen	Et OIOXML Skema tilhørende <i>Kerneklassen</i> ( <i>OIOXML Core Class</i> ) SKAL være et engelsk OIOXML skema.
LNR-7	Sprogvalg i Domæne- og NDR-klassen	Et OIOXML skema tilhørende <i>Domæneklassen</i> ( <i>OIOXML Domain Class</i> ) eller <i>NDR-klassen</i> ( <i>OIOXML NDR Class</i> ) BØR være et engelsk OIOXML skema.
LNR-8	Sprogvalg i Domæne- og NDR-klassen	Et OIOXML skema tilhørende <i>Domæneklassen</i> ( <i>OIOXML Domain Class</i> ) eller <i>NDR-klassen</i> ( <i>OIOXML NDR Class</i> ) KAN være et dansk OIOXML skema, hvis det indgår i en sammenhæng eller i et domæne eller subdomæne, hvor dansk skaber en større begrebsmæssig klarhed end engelsk.
LNR-9	Sprogvalg i Domæne- og NDR-klassen	For et OIOXML skema tilhørende <i>Domæneklassen</i> ( <i>OIOXML Domain Class</i> ) er det en domænekomité for skemaets respektive domæne, der beslutter, hvilket sprog skemaet navngives i.
<b>Kontrolleret sprogsift i et OIOXML skema</b>		
LNR-10	Reference fra engelsk til	Et engelsk OIOXML skema MÅ IKKE via <i>include</i>

OIOXML NDR Quick  
Reference

Regel kode	Regeltitel	Regeltekst
<b>Navngivning</b>		
	dansk OIOXML skema	konstruktionen inkludere eller via <i>import</i> konstruktionen importere et dansk OIOXML skema.
LNR-11	Genbrug i engelske elementer og attributer	Et engelsk OIOXML skema MÅ IKKE erklære et element eller en attribut, der genbruger en dansk type defineret i et dansk OIOXML skema.
LNR-12	Genbrug i engelske typer	Et engelsk OIOXML skema MÅ IKKE i en kompleks type genbruge danske komponenter (elementer og typer) definerede eller erklærede i et dansk OIOXML skema.
LNR-13	Genbrug i engelske typer	Et engelsk OIOXML skema MÅ IKKE i en simpel eller kompleks type nedarve fra en dansk type erklæret i et dansk OIOXML skema.
<b>Navngivning af typer</b>		
TPN-1	Type suffix	Et OIOXML skema SKAL afslutte navnet på en simpel eller kompleks type med suffix <i>Type</i> .
TPN-2	Navne for komplekse typer	Et OIOXML skema MÅ IKKE anvende termen <i>Representation</i> i navnet for en kompleks type. Termen <i>Egenskab</i> i typenavnet SKAL være <i>Collection</i> i et engelsk OIOXML skema og <i>Samling</i> i et dansk OIOXML skema, hvis typen indeholder præcist ét element og mindst 2 forekomster af dette. I alle andre tilfælde SKAL termen <i>Egenskab</i> i typenavnet være <i>Structure</i> i et engelsk OIOXML skema og <i>Struktur</i> i et dansk OIOXML skema eller helt undlades.
TPN-3	Anvendelse af UpperCamelCase	Et OIOXML skema SKAL navngive dets simple og komplekse typer med <i>UpperCamelCase</i> .
TPN-4	Navne for støttetyper	Et OIOXML skema SKAL anvende ét underscore (_) som prefix til en type's navn, hvis typen er en <i>støttetype</i> .
<b>Navngivning af elementernavngivningselementer</b>		
ELN-1	Sammenhæng mellem element- og typenavn	Et OIOXML skema BØR navngive dets element identisk med elementets type (uden typens <i>Type</i> suffix), hvis elementerklæring og typedefinition forekommer sammen i skemaet.
ELN-2	Anvendelse af UpperCamelCase	Et OIOXML skema SKAL navngive dets element med <i>UpperCamelCase</i> .
<b>Navngivning af attributternavngivningattributer</b>		
ATN-1	Anvendelse af lowerCamelCase	Et OIOXML skema SKAL navngive dets attributter med <i>lowerCamelCase</i> .



Regel kode	Regeltitel	Regeltekst
<b>Navngivning</b>		
<b>Navngivning af OIOXML skema- og metadatafiler</b>		
FNR-1	Navngivning af OIOXML skema fil	Et OIOXML skema SKAL navngive dets fil efter modellen: <i>&lt;namespace-prefix med store bogstaver&gt;+_+&lt;elementnavn&gt;+.xsd.</i>
FNR-2	Navngivning af metadata fil	Et OIOXML skema SKAL navngive dets metadatafil efter modellen: <i>&lt;skemafilnavn&gt;+.meta.xml.</i>

Regel kode	Regeltitel	Regeltekst
<b>Regler for typedefinitioner</b>		
<b>Generelle regler for typedefinitioner</b>		
GTD-1	Stærke datatyper	Et OIOXML skema SKAL definere alle dets simple og komplekse typer stærkest muligt.
GTD-2	Globale typedefinitioner	Et OIOXML skema tilhørende <i>Kerneklassen (OIOXML Core Class)</i> eller <i>Domæneklassen (OIOXML Domain Class)</i> SKAL definere alle dets simple og komplekse typer globalt.
GTD-3	Globale typedefinitioner	Et OIOXML skema tilhørende <i>NDR-klassen (OIOXML NDR Class)</i> BØR definere alle dets simple eller komplekse typer globalt.
GTD-4	Ny definition af en eksisterende type	Et OIOXML skema MÅ IKKE definere en ny simpel eller kompleks type identisk med en simpel eller kompleks type fra et andet eksisterende OIOXML skema.
GTD-5	anyType og anySimpleType	Et OIOXML skema MÅ IKKE genbruge de indbyggede ur-typer <i>anyType</i> og <i>anySimpleType</i> .
GTD-6	Håndtering af binært indhold	Et OIOXML skema BØR benytte de indbyggede simple typer <i>anyURI</i> eller <i>base64Binary</i> til at håndtere binært indhold.
GTD-7	Abstrakte typer	Et OIOXML skema KAN benytte attributten <i>abstract</i> i simple og komplekse typedefinitioner.
GTD-8	Kontrol af typeafledninger	Et OIOXML skema BØR IKKE begrænse typeafledninger, dvs. attributterne <i>finalDefault</i> og <i>blockDefault</i> i rodelementet <i>schema</i> samt attributterne <i>block</i> og <i>final</i> i simple og komplekse typedefinitioner BØR IKKE benyttes.
GTD-9	Anvendelse af støttetyper	Et OIOXML skema SKAL anvende dets støttetyper udelukkende som en hjælp til at etablere skemaets ene typedefinition, hvis en sådan defineres i skemaet og

OIOXML NDR Quick  
Reference

Regel kode	Regeltitel	Regeltekst
<b>Regler for typedefinitioner</b>		
		ikke kan udtrykkes i en enkelt definition.
GTD-10	Anvendelse af støttetyper	Et OIOXML skema SKAL definere dets støttetyper som <i>simple typer</i> .
<b>Regler for simple typedefinitioner</b>		
STD-1	Anvendelse af list	Et OIOXML skema MÅ IKKE benytte konstruktionen <i>list</i> .
STD-2	Anvendelse af union	Et OIOXML skema MÅ IKKE benytte konstruktionen <i>union</i> .
STD-3	Længden af string	Et OIOXML skema BØR IKKE begrænse længden af den simple indbyggede type <i>string</i> , hvis der ikke eksisterer en fælles vedtagen længde.
STD-4	Repræsentation af kodelister	Et OIOXML skema BØR udtrykke kodelister via <i>enumeration</i> konstruktionen.
STD-5	Værdier i enumerationer	Et OIOXML skema BØR udelukkende udtrykke værdier i <i>enumeration</i> konstruktioner med små bogstaver.
STD-6	Værdier i enumerationer	Et OIOXML skema KAN udtrykke værdier i <i>enumeration</i> konstruktioner både på dansk eller engelsk.
STD-7	Anvendelse af whitespace facetten og dertil koblede typer	Et OIOXML skema MÅ IKKE benytte <i>whitespace</i> facetten samt de indbyggede simple typer <i>token</i> og <i>normalizedString</i> .
<b>Regler for komplekse typedefinitionertyperkomplekse</b>		
CTD-1	Opbygning af komplekse typer	Et OIOXML skema BØR definere en kompleks type ved anvendelse af konstruktionerne <i>sequence</i> og <i>choice</i> .
CTD-2	Opbygning af komplekse typer	Et OIOXML skema MÅ IKKE definere en kompleks type ved anvendelse af konstruktionen <i>all</i> .
CTD-3	Definition af komplekse typer ved afledning	Et OIOXML skema KAN definere en kompleks type via <i>extension</i> konstruktionen.
CTD-4	Definition af komplekse typer ved afledning	Et OIOXML skema MÅ IKKE definere en kompleks type via <i>restriction</i> konstruktionen.
CTD-5	Anvendelse af blandet indholdsmodel	Et OIOXML skema KAN benytte den blandede indholdsmodel (mixed content) til strukturer, hvor tekst og elementer frit kan blandes.
CTD-6	Anvendelse af tom indholdsmodel	Et OIOXML skema KAN benytte den tomme indholdsmodel (empty content) til strukturer, hvor tekst og elementer frit kan blandes.
CTD-7	Anvendelse af any	Et OIOXML skema KAN benytte <i>any</i> konstruktionen til at binde to relaterede, og eventuelt uafhængige,

Regel kode	Regeltitel	Regeltekst
<b>Regler for typedefinitioner</b>		
		OIOXML dialekter sammen.
CTD-8	Anvendelse af any	Et OIOXML skema MÅ IKKE benytte <i>any</i> konstruktionen til at skabe en bevidst svag indholdsmodel indenfor samme OIOXML dialekt eller til generelt at skabe en uforpligtende overgang til et eller flere elementer, som enten er statisk kendt på forhånd eller som ikke reelt repræsenterer en selvstændig OIOXML dialekt.
CTD-9	Anvendelse af any	Et OIOXML skema SKAL i en <i>any</i> konstruktion sætte værdien af attributten <i>processContents</i> til <i>strict</i> .
CTD-10	Anvendelse af anyAttribute	Et OIOXML skema MÅ IKKE benytte <i>anyAttribute</i> konstruktionen i indholdsmodellen for en kompleks type.
CTD-11	Komplekse typer i Kerneschemaer	Et OIOXML skema tilhørende <i>Kerneklassen (OIOXML Core Class)</i> SKAL kun genbruge eksisterende OIOXML Kerneelementer i indholdsmodellen for dets komplekse typedefinition.

Regel kode	Regeltitel	Regeltekst
<b>Regler for element- og attributerklæringer</b>		
<b>Regler for elementerklæringerelementer</b>		
ELD-1	Globale elementerklæringer	Et OIOXML skema tilhørende <i>Kerneklassen (OIOXML Core Class)</i> eller <i>Domæneklassen (OIOXML Domain Class)</i> SKAL erklære dets element globalt.
ELD-2	Globale elementerklæringer	Et OIOXML skema tilhørende <i>NDR-klassen (OIOXML NDR Class)</i> BØR erklære dets element globalt.
ELD-3	Namespace for elementer	Et OIOXML skema SKAL tildele et namespace til dets element, dvs. attributten <i>elementFormDefault</i> i rodelementet <i>schema</i> SKAL tildeles værdien <i>qualified</i> og attributten <i>form</i> i elementerklæringer MÅ IKKE benyttes.
ELD-4	Anvendelse af substitution groups	Et OIOXML skema MÅ IKKE benytte substitution groups i dets elementerklæring.
ELD-5	Anvendelse af nillable	Et OIOXML skema tilhørende <i>Kerneklassen (OIOXML Core Class)</i> eller <i>Domæneklassen (OIOXML Domain Class)</i> MÅ IKKE sætte <i>nillable</i> attributten til værdien <i>true</i> i dets elementerklæring.
ELD-6	Anvendelse af nillable	Et OIOXML skema MÅ IKKE sætte <i>nillable</i>

Regel kode	Regeltitel	Regeltekst
<b>Regler for element- og attributerklæringer</b>		
		attributten til værdien <i>true</i> i dets elementerklæring, hvis det direkte eller indirekte genbruger en <i>Kernetype</i> eller <i>Domænetype</i> .
ELD-7	Anvendelse af <i>nillable</i>	Et OIOXML skema tilhørende <i>NDR-klassen (OIOXML NDR Class)</i> BØR IKKE sætte <i>nillable</i> attributten til værdien <i>true</i> i dets elementerklæring og i givet fald kun hvis elementets type er en simpel type eller hvis elementets type er en kompleks type, der udelukkende indeholder attributerklæringer.
ELD-8	Default- og konstantværdier	Et OIOXML skema MÅ IKKE benytte attributten <i>default</i> i dets elementerklæring.
ELD-9	Default- og konstantværdier	Et OIOXML skema MÅ IKKE benytte attributten <i>fixed</i> i dets elementerklæring.
<b>Regler for attributerklæringattributter</b>		
ATD-1	Anvendelsen af attributter	Et OIOXML skema SKAL benytte dets elements attributter til metadata.
ATD-2	Attributter, genbrug og namespaces	Et OIOXML skema SKAL erklære alle dets attributter lokalt.
ATD-3	Attributter, genbrug og namespaces	Et OIOXML skema MÅ IKKE tildele et namespace til dets attributter, dvs. attributten <i>attributeFormDefault</i> i rodelementet <i>schema</i> SKAL tildeles værdien <i>unqualified</i> , eller slet ikke specificeres ( <i>unqualified</i> er defaultværdien), og attributten <i>form</i> i attributerklæringer MÅ IKKE benyttes.
ATD-4	Default- og konstantværdier	Et OIOXML skema MÅ IKKE benytte attributten <i>default</i> i dets attributerklæringer.
ATD-5	Default- og konstantværdier	Et OIOXML skema MÅ IKKE benytte attributten <i>fixed</i> i dets attributerklæringer.

Regel kode	Regeltitel	Regeltekst
<b>Versionerings- og namespace regler</b>		
VER-1	Namespaceversionering	Et OIOXML skema SKAL angive dets version ved hjælp af et namespace.
VER-2	Låsning af et OIOXML skema	Et OIOXML skema, som er godkendt og optaget i en af OIOXML klasserne, MÅ IKKE ændres.
VER-3	Bagudkompatible udvidelser	Et OIOXML skema KAN benytte et eksisterende namespace ved bagudkompatible udvidelser.
VER-4	Anvendelse af attributten	Et OIOXML skema KAN frit benytte attributten

OIOXML NDR Quick  
Reference

Regel kode	Regeltitel	Regeltekst
<b>Versionerings- og namespaceregler</b>		
	version	<i>version</i> i rodelementet <i>schema</i> .
NMS-1	Namespace repræsentation	Et OIOXML skema SKAL anvende et namespace, som repræsenterer en gyldig URL i Infostrukturbasen og har følgende opbygning: http://rep.oio.dk/<internetdomæne>/xml/schemas/<YY YY>/<MM>/<DD>/.
NMS-2	Navngivning af prefix i namespaceerklæringer	Et OIOXML skema SKAL basere sit namespace-prefix på navnet af internetdomænet i namespaceet uden punktum og roddomæne.

Regel kode	Regeltitel	Regeltekst
<b>Dokumentation og metadata</b>		
<b>Dokumentation</b>		
DOC-1	Angivelse af dokumentation i metadata	Et OIOXML skema SKAL placere al dets dokumentation i skemaets metadata.
DOC-2	Title	Et OIOXML skema SKAL udfylde metadata-elementet <i>Title</i> med en overordnet titel for skemaet.
DOC-3	DescriptionEnglish og DescriptionDanish	Hvis metadata-elementet <i>DescriptionEnglish</i> udfyldes, SKAL metadata-elementet indeholde en udførlig beskrivelse af skemaets funktion på <i>engelsk</i> .
DOC-4	DescriptionEnglish og DescriptionDanish	Hvis metadata-elementet <i>DescriptionDanish</i> udfyldes, SKAL metadata-elementet indeholde en udførlig beskrivelse af skemaets funktion på <i>dansk</i> .
<b>Dokumentation for OIOXML klasserne</b>		
DOC-5	Dokumentation for Kerneklassen	Et OIOXML skema tilhørende <i>Kerneklassen</i> ( <i>OIOXML Core Class</i> ) SKAL udfylde metadata-elementerne <i>DescriptionEnglish</i> og <i>DescriptionDanish</i> .
DOC-6	Dokumentation for Domæne- og NDR-klassen	Et OIOXML skema tilhørende <i>Domæneklassen</i> ( <i>OIOXML Domain Class</i> ) eller <i>NDR-klassen</i> ( <i>OIOXML NDR Class</i> ) SKAL udfylde enten metadata-elementet <i>DescriptionEnglish</i> eller metadata-elementet <i>DescriptionDanish</i> , og KAN udfylde begge metadata-elementer, hvilket anbefales.
<b>Anvendelse af Labels metadata</b>		
DOC-7	Angivelse af	Et OIOXML skema KAN udfylde metadata-

OIOXML NDR Quick  
Reference

Regel kode	Regeltitel	Regeltekst
<b>Dokumentation og metadata</b>		
	brugervenlig term	elementerne <i>LabelEnglish</i> og/eller <i>LabelDanish</i> til at angive en brugervenlig term, som egner sig til præsentation i en brugergrænseflade.
DOC-8	LabelEnglish og LabelDanish	Hvis metadata-elementet <i>LabelEnglish</i> udfyldes, SKAL termen være på <i>engelsk</i> .
DOC-9	LabelEnglish og LabelDanish	Hvis metadata-elementet <i>LabelDanish</i> udfyldes, SKAL termen være på <i>dansk</i> .
DOC-10	Anvendelse af appinfo	Et OIOXML skema MÅ IKKE benytte <i>appinfo</i> konstruktionen.
DOC-11	Dokumentation af kodeliste	Et OIOXML skema SKAL dokumentere alle værdierne i en kodeliste enten lokalt eller med en reference til et andet fyldestgørende dokument i Infostrukturbasen.
DOC-12	Afleveringsdokument	En OIOXML aflevering SKAL indeholde afleveringsdokumentation, der overordnet beskriver afleveringens skemaer.
<b>Metadata</b>		
MTA-1	Metadata for OIOXML skemaer	Et OIOXML skema SKAL have tilknyttet metadata.

---

# Stikordsregister

## A

abstrakte typer, 28  
Adoptionsklassen, 3  
afleveringsdokumentation, 44  
akronymer  
  entalsform, 18  
attributter, 37  
  defaultværdi, 38  
  genbrug, 37  
  konstantværdi, 38  
  metadata, 37  
  namespace, 37

## B

binært indhold, 28

## D

dokumentation, 41  
  dansk, 41  
  DescriptionDanish, 41, 42  
  DescriptionEnglish, 41, 42  
  engelsk, 41  
  kodeliste, 44  
  metadata, 41  
  metadataelementer, 41  
  skemaaflevering, 44  
  Title, 41, 41  
Domæneklassen, 3, 4  
Domæneskema, 3

## E

egenskabsterm (se navngivning)  
elementer, 35  
  defaultværdi, 37  
  genbrug, 35  
  globale erklæringer, 35  
  konstantværdi, 37  
  namespace, 35  
  nillable, 36  
  substitution groups, 35  
encoding scheme  
  Unicode, 12  
  UTF-8, 12  
Extensible Markup Language (XML), 12

## G

genbrug, 4, 15  
  af element fremfor type, 6, 7

  af Kerneelement, 6  
  af Kernetype, 7  
  af seneste skemaversion, 8  
  attributter, 37  
genbrugsklasser, 3  
genbrugstanken, 4  
grundklasser, 3

## I

indbyggede typer, 5  
  xsd:date, 5  
Infostrukturbasen, vi, 14  
  OIOXML skema, 10

## K

Kerneklassen, 3, 4  
Kerneskema, 3  
kodeliste, 44

## L

lowerCamelCase for attributter, 24

## M

metadata, vi, 37  
  brugervenlig term, 43  
  dokumentation, 41  
  LabelDanish, 43, 43  
  LabelEnglish, 43, 43  
  navngivning af fil, 25  
  OIOXML skema, 8, 44

## N

namespace, 35  
  OIOXML skema, 13  
  opbygning, 40  
  prefix, 40  
  versionering, 39  
navnekollision, 13  
navngivning  
  "Type" suffix, 23, 24  
  attributter, 24  
  bindeord, 18  
  dansk, 19  
  ebXML Naming Convention, 15  
  egenskabsterm, 15, 16  
  elementer, 24  
  engelsk, 19  
  entalsform, 18  
  forkortelser, 18  
  ISO 11179, 15  
  komplekse typer, 23  
  kontrolleret sprogsift, 21

- 
- lowerCamelCase for attributter, 24
  - metadataafil, 25
  - navngivningsmodel, 15
  - ObjektEgenskabRepræsentation, 15
  - objektterm, 15, 16
  - repræsentationsterm, 15, 16
  - retningslinie for sprogvvalg, 19
  - skemafil, 25
  - specialtegn, 18
  - sprogskift via xsd:any, 22
  - sprogvvalg for OIOXML klasser, 20
  - støttetyper, 24
  - unik, 15
  - UpperCamelCase for elementer, 24
  - UpperCamelCase for typer, 23
  - æ, ø, å, 20
  - navngivningsmodel, 15
  - NDR, v
    - regelprefix, 1
  - NDR-klassen, 3, 5
  - NDR regler
    - [ATD-1], 37
    - [ATD-2], 37
    - [ATD-3], 38
    - [ATD-4], 38
    - [ATD-5], 38
    - [ATN-1], 24
    - [CTD-1], 31
    - [CTD-10], 34
    - [CTD-11], 34
    - [CTD-2], 32
    - [CTD-3], 32
    - [CTD-4], 32
    - [CTD-5], 32
    - [CTD-6], 32
    - [CTD-7], 33
    - [CTD-8], 33
    - [CTD-9], 33
    - [DOC-1], 41
    - [DOC-10], 44
    - [DOC-11], 44
    - [DOC-12], 44
    - [DOC-2], 41
    - [DOC-3], 42
    - [DOC-4], 42
    - [DOC-5], 42
    - [DOC-6], 42
    - [DOC-7], 43
    - [DOC-8], 43
    - [DOC-9], 43
    - [ELD-1], 35
    - [ELD-2], 35
    - [ELD-3], 35
    - [ELD-4], 35
    - [ELD-5], 36
    - [ELD-6], 36
    - [ELD-7], 36
    - [ELD-8], 37
    - [ELD-9], 37
    - [ELN-1], 24
    - [ELN-2], 24
    - [FNR-1], 25
    - [FNR-2], 25
    - [GNR-1], 15
    - [GNR-2], 15
    - [GNR-2a], 16
    - [GNR-2b], 16
    - [GNR-2c], 16
    - [GNR-2d], 16
    - [GNR-2e], 18
    - [GNR-2f], 18
    - [GNR-2g], 18
    - [GNR-2h], 18
    - [GNR-2i], 18
    - [GTD-1], 26
    - [GTD-10], 28
    - [GTD-2], 27
    - [GTD-3], 27
    - [GTD-4], 27
    - [GTD-5], 27
    - [GTD-6], 27
    - [GTD-7], 28
    - [GTD-8], 28
    - [GTD-9], 28
    - [GXS-1], 12
    - [GXS-2], 12
    - [GXS-3], 12
    - [GXS-4], 12
    - [GXS-5], 13
    - [GXS-6], 13
    - [GXS-7], 13
    - [GXS-8], 13
    - [LNR-1], 19
    - [LNR-10], 21
    - [LNR-11], 21
    - [LNR-12], 22
    - [LNR-13], 22
    - [LNR-2], 19
    - [LNR-3], 19
    - [LNR-4], 20
    - [LNR-5], 20
    - [LNR-6], 20
    - [LNR-7], 20
    - [LNR-8], 20
-



- [LNR-9], 21
  - [MTA-1], 45
  - [NMS-1], 40
  - [NMS-2], 40
  - [OIO-1], 4
  - [OIO-10], 10
  - [OIO-11], 10
  - [OIO-12], 11
  - [OIO-2], 5
  - [OIO-3], 5
  - [OIO-4], 6
  - [OIO-5], 8
  - [OIO-6], 8
  - [OIO-7], 8
  - [OIO-8], 10
  - [OIO-9], 10
  - [STD-1], 29
  - [STD-2], 30
  - [STD-3], 30
  - [STD-4], 30
  - [STD-5], 31
  - [STD-6], 31
  - [STD-7], 31
  - [TPN-1], 23
  - [TPN-2], 23
  - [TPN-3], 23
  - [TPN-4], 23
  - [VER-1], 39
  - [VER-2], 39
  - [VER-3], 39
  - [VER-4], 39
  - NDR-skema, 3
- O**
- objektterm (se navngivning)
  - OIO publikation, v
  - OIOXML dialekt, 33
  - OIOXML grundlaget, v
  - OIOXML klasser, vi
  - OIOXML metadata, vi
  - OIOXML Naming and Design Rules, v
  - OIOXML Navngivnings- og Design Regler, v
  - OIOXML NDR, v, 1
  - OIOXML paradigmet, v, v, vi
  - OIOXML skema, vi, 1
    - bagudkompatibel, 39
    - dansk, 19
    - engelsk, 19
    - indhold, 8
    - Infostrukturbasen, 10
    - klare og entydige, 11
    - låsning, 39
    - metadata, 8, 44
    - namespace, 13
    - navngivning af fil, 25
    - sprogangivelse med xml:lang, 19
    - støttetyper, 8
    - systemuafhængig, 11
    - version, 39
  - OIOXML skemaaflevering, vi
- R**
- repræsentationsterm (se navngivning)
- S**
- schemaLocation, 14
  - skemaeksempler, 2
  - skemagenbrug, 4
  - skemaversionering, 8
    - namespace, 13
  - skemaaflevering, vi, 13
  - støttetyper, 8
  - substitution groups, 35
- T**
- targetNamespace, 13
  - typer
    - abstrakte, 28
    - afledning, 32
    - binært indhold, 28
    - empty content, 33
    - generelle regler, 26
    - globale typedefinitioner, 27
    - kodelister, 31
    - komplekse, 31
    - komplekse typer, 26
    - mixed content, 32
    - simple typer, 26
    - stærke datatyper, 26
    - typeafledning, 28
- U**
- Unicode (se encoding scheme)
  - unik navngivning, 15
  - UpperCamelCase for elementer, 24
  - UpperCamelCase for typer, 23
- V**
- version, 8
  - versionering
    - namespace, 39

## X

xml:lang, 19  
XML Komité, 1  
XML Schema anbefalingen, vi, 12  
xsd:abstract, 28  
xsd:all, 32  
xsd:any, 22, 33  
xsd:anyAttribute, 34  
xsd:anySimpleType, 27  
xsd:anyType, 27  
xsd:anyURI, 28  
xsd:appinfo, 44  
xsd:attributeFormDefault, 38  
xsd:base64Binary, 28  
xsd:block, 28  
xsd:blockDefault, 28  
xsd:choice, 32  
xsd:date, 5  
xsd:default, 37, 38  
xsd:elementFormDefault, 35  
xsd:enumeration, 31, 31  
xsd:extension, 32  
xsd:final, 28  
xsd:finalDefault, 28  
xsd:fixed, 37, 38  
xsd:form, 35, 38  
xsd:import, 13  
xsd:include, 13  
xsd:length, 30  
xsd:list, 29  
xsd:maxLength, 30  
xsd:maxOccurs, 32  
xsd:minOccurs, 32, 36  
xsd:namespace, 33  
xsd:nillable, 36  
xsd:normalizedString, 31  
xsd:notation, 13  
xsd:pattern, 31  
xsd:processContents, 33  
xsd:redefine, 13  
xsd:restriction, 32  
xsd:schema, 13, 35  
xsd:sequence, 32  
xsd:string, 30  
xsd:token, 31  
xsd:union, 30  
xsd:version, 39  
xsd:whitespace, 31