XML Schema Documentation

Table of Contents

- Schema Document Properties
- Global Definitions
  - Complex Type: **ACK_Declarant_Type**
  - Complex Type: **ACK_Representative_Type**
  - Complex Type: **Declarant_Type**
  - Complex Type: **DeclarantBE_Type**
  - Complex Type: **FileAttachment_Type**
  - Complex Type: **MTN_Declarant_Type**
  - Complex Type: **Representative_Type**
  - Complex Type: **RepresentativeID_Type**
  - Complex Type: **VR_Declarant_Type**
  - Complex Type: **VR_Representative_Type**
  - Simple Type: **BEVATNumber**
  - Simple Type: **City_Type**
  - Simple Type: **Comment_Type**
  - Simple Type: **DeclarantReference_Type**
  - Simple Type: **EMail_Type**
  - Simple Type: **EUVATNumber**
  - Simple Type: **FileTypeCode**
  - Simple Type: **IntervatConsignmentReference_Type**
  - Simple Type: **IntervatDeclarationReference_Type**
  - Simple Type: **IntervatOrSTIRINTDeclReference_Type**
  - Simple Type: **Month_Type**
  - Simple Type: **Name_Type**
  - Simple Type: **PositiveAmount_Type**
  - Simple Type: **PostCode_Type**
  - Simple Type: **Quarter_Type**
  - Simple Type: **RepresentativeIDType_Type**
  - Simple Type: **RepresentativeReference_Type**
  - Simple Type: **RestrictedDate_Type**
  - Simple Type: **RestrictedDateTime_Type**
  - Simple Type: **SignedAmount_Type**
  - Simple Type: **STIRINTReference_Type**
  - Simple Type: **Street_Type**
  - Simple Type: **UnlimitedSignedAmount_Type**
  - Simple Type: **Year_Type**
  - Simple Type: **YesNo_Type**

top

Schema Document Properties

Target Namespace	**http://www.minfin.fgov.be/InputCommon**
Version	1.0
Language	en
Element and Attribute Namespaces	• Global element and attribute declarations belong to this schema's target namespace. • By default, local element declarations belong to this schema's target namespace. • By default, local attribute declarations have no namespace.

Schema Composition	- This schema imports schema(s) from the following namespace(s): ○ *http://www.minfin.fgov.be/IsoTypes* (at IntervatIsoTypes_v0_7.xsd) ○ *urn:ec.europa.eu:taxud:fiscalis:common:v1* (at commontypes_v1.xsd)
Documentation	=== HISTORY=== 1.1 => import commonTypes_v1.xsd and add the phoneNumber to the Representative_Type

Declared Namespaces

Prefix	Namespace
Default namespace	**http://www.minfin.fgov.be/InputCommon**
xml	http://www.w3.org/XML/1998/namespace
xs	http://www.w3.org/2001/XMLSchema
iso	http://www.minfin.fgov.be/IsoTypes
cm	urn:ec.europa.eu:taxud:fiscalis:common:v1

Schema Component Representation

```
<xs:schema xml:lang="en" targetNamespace="http://www.minfin.fgov.be
/InputCommon" elementFormDefault="qualified" version="1.0">
   <xs:import namespace="http://www.minfin.fgov.be/IsoTypes"
   schemaLocation="IntervatIsoTypes_v0_7.xsd"/>
   <xs:import namespace="urn:ec.europa.eu:taxud:fiscalis:common:v1"
   schemaLocation="commontypes_v1.xsd"/>
   ...
</xs:schema>
```

top

Global Definitions

Complex Type: ACK_Declarant_Type

Super-types:	None
Sub-types:	None

Name	ACK_Declarant_Type
Abstract	no

XML Instance Representation

```
<...>
   <VATNumber> BEVATNumber </VATNumber> [1]  ?
   <Name> Name_Type </Name> [1]  ?
   <Street> Street_Type </Street> [1]  ?
   <PostCode> PostCode_Type </PostCode> [0..1]  ?
   <City> City_Type </City> [1]  ?
   <CountryCode> iso:MSCountryCode </CountryCode> [1]  ?
   <EmailAddress> EMail_Type </EmailAddress> [0..1]  ?
   <Phone> cm:PhoneNumber_Type </Phone> [0..1] ?
</...>
```

Schema Component Representation

```
<xs:complexType name="ACK_Declarant_Type">
  <xs:sequence>
    <xs:element name="VATNumber" type="BEVATNumber"/>
    <xs:element name="Name" type="Name_Type"/>
    <xs:element name="Street" type="Street_Type"/>
    <xs:element name="PostCode" type="PostCode_Type" minOccurs="0"/>
    <xs:element name="City" type="City_Type"/>
    <xs:element name="CountryCode" type="iso:MSCountryCode"/>
    <xs:element name="EmailAddress" type="EMail_Type" minOccurs="0"/>
    <xs:element name="Phone" type="cm:PhoneNumber_Type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

top

Complex Type: ACK_Representative_Type

Super-types:	None
Sub-types:	None

Name	ACK_Representative_Type
Abstract	no

XML Instance Representation

```
<...>
   <RepresentativeID> RepresentativeID_Type </RepresentativeID> [1] ?
   <Name> Name_Type </Name> [1] ?
   <Street> Street_Type </Street> [0..1] ?
   <PostCode> PostCode_Type </PostCode> [0..1] ?
   <City> City_Type </City> [1] ?
   <CountryCode> iso:CountryCode </CountryCode> [1] ?
   <EmailAddress> EMail_Type </EmailAddress> [1] ?
   <Phone> cm:PhoneNumber_Type </Phone> [1] ?
</...>
```

Schema Component Representation

```
<xs:complexType name="ACK_Representative_Type">
  <xs:sequence>
    <xs:element name="RepresentativeID" type="RepresentativeID_Type"/>
    <xs:element name="Name" type="Name_Type"/>
    <xs:element name="Street" type="Street_Type" minOccurs="0"/>
    <xs:element name="PostCode" type="PostCode_Type" minOccurs="0"/>
    <xs:element name="City" type="City_Type"/>
    <xs:element name="CountryCode" type="iso:CountryCode"/>
    <xs:element name="EmailAddress" type="EMail_Type"/>
    <xs:element name="Phone" type="cm:PhoneNumber_Type"/>
  </xs:sequence>
</xs:complexType>
```

top

Complex Type: Declarant_Type

Super-types:	None
Sub-types:	None

Name	Declarant_Type
Abstract	no
Documentation	Déclarant d'un envoi de type TVA, listing clients, relevé intracommunautaire.

XML Instance Representation

```
<...>
   <VATNumber> BEVATNumber </VATNumber> [1] ?
   <Name> Name_Type </Name> [0..1] ?
   <Street> Street_Type </Street> [0..1] ?
   <PostCode> PostCode_Type </PostCode> [0..1] ?
   <City> City_Type </City> [0..1] ?
   <CountryCode> iso:MSCountryCode </CountryCode> [0..1] ?
   <EmailAddress> EMail_Type </EmailAddress> [0..1] ?
   <Phone> cm:PhoneNumber_Type </Phone> [0..1] ?
</...>
```

Schema Component Representation

```
<xs:complexType name="Declarant_Type">
   <xs:sequence>
      <xs:element name="VATNumber" type="BEVATNumber"/>
      <xs:element name="Name" type="Name_Type" minOccurs="0"/>
      <xs:element name="Street" type="Street_Type" minOccurs="0"/>
      <xs:element name="PostCode" type="PostCode_Type" minOccurs="0"/>
      <xs:element name="City" type="City_Type" minOccurs="0"/>
      <xs:element name="CountryCode" type="iso:MSCountryCode"
      minOccurs="0"/>
      <xs:element name="EmailAddress" type="EMail_Type" minOccurs="0"/>
      <xs:element name="Phone" type="cm:PhoneNumber_Type" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

top

Complex Type: DeclarantBE_Type

Super-types:	None
Sub-types:	None

Name	DeclarantBE_Type
Abstract	no
Documentation	Déclarant d'un envoi de type déclaration spéciale 629, VATRefund ou modification du prorata VATRefund (pays de l'adresse en Belgique).

XML Instance Representation

```
<...>
   <VATNumber> BEVATNumber </VATNumber> [1] ?
   <Name> Name_Type </Name> [0..1] ?
   <Street> Street_Type </Street> [0..1]  ?
   <PostCode> PostCode_Type </PostCode> [0..1]  ?
   <City> City_Type </City> [0..1]  ?
   <CountryCode> iso:BECountryCode </CountryCode> [0..1]  ?
   <EmailAddress> EMail_Type </EmailAddress> [0..1]  ?
   <Phone> cm:PhoneNumber_Type </Phone> [0..1]  ?
</...>
```

Schema Component Representation

```
<xs:complexType name="DeclarantBE_Type">
   <xs:sequence>
      <xs:element name="VATNumber" type="BEVATNumber"/>
      <xs:element name="Name" type="Name_Type" minOccurs="0"/>
      <xs:element name="Street" type="Street_Type" minOccurs="0"/>
      <xs:element name="PostCode" type="PostCode_Type" minOccurs="0"/>
      <xs:element name="City" type="City_Type" minOccurs="0"/>
      <xs:element name="CountryCode" type="iso:BECountryCode"
      minOccurs="0"/>
      <xs:element name="EmailAddress" type="EMail_Type" minOccurs="0"/>
      <xs:element name="Phone" type="cm:PhoneNumber_Type" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

top

Complex Type: FileAttachment_Type

Super-types:	None
Sub-types:	None

Name	FileAttachment_Type
Abstract	no

XML Instance Representation

```
<...>
   <FileType> FileTypeCode </FileType> [1]  ?
   <FileName> xs:token </FileName> [1]  ?
   <FileDescription> xs:token </FileDescription> [0..1]  ?
</...>
```

Schema Component Representation

```
<xs:complexType name="FileAttachment_Type">
   <xs:sequence>
      <xs:element name="FileType" type="FileTypeCode"/>
      <xs:element name="FileName" type="xs:token"/>
      <xs:element name="FileDescription" type="xs:token" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

top

Complex Type: MTN_Declarant_Type

Super-types:	None
Sub-types:	None

Name	MTN_Declarant_Type
Abstract	no
Documentation	Déclarant d'un envoi de type MTN.

XML Instance Representation

```
<...>
  <VATNumber> BEVATNumber </VATNumber> [1]  ?
  <Name> Name_Type </Name> [0..1]  ?
  <Street> Street_Type </Street> [0..1]  ?
  <PostCode> PostCode_Type </PostCode> [0..1]  ?
  <City> City_Type </City> [0..1]  ?
  <CountryCode> iso:MSCountryCode </CountryCode> [0..1]  ?
  <EmailAddress> EMail_Type </EmailAddress> [1]  ?
  <Phone> cm:PhoneNumber_Type </Phone> [0..1]  ?
</...>
```

Schema Component Representation

```
<xs:complexType name="MTN_Declarant_Type">
  <xs:sequence>
    <xs:element name="VATNumber" type="BEVATNumber"/>
    <xs:element name="Name" type="Name_Type" minOccurs="0"/>
    <xs:element name="Street" type="Street_Type" minOccurs="0"/>
    <xs:element name="PostCode" type="PostCode_Type" minOccurs="0"/>
    <xs:element name="City" type="City_Type" minOccurs="0"/>
    <xs:element name="CountryCode" type="iso:MSCountryCode"
    minOccurs="0"/>
    <xs:element name="EmailAddress" type="EMail_Type"/>
    <xs:element name="Phone" type="cm:PhoneNumber_Type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

top

Complex Type: Representative_Type

Super-types:	None
Sub-types:	None

Name	Representative_Type
Abstract	no
Documentation	Mandataire d'un envoi de type TVA, listing clients, relevé intracommunautaire, déclaration spéciale 629 ou MTN.

XML Instance Representation

```
<...>
  <RepresentativeID> RepresentativeID_Type </RepresentativeID> [1]  ?
  <Name> Name_Type </Name> [1]  ?
  <Street> Street_Type </Street> [1]  ?
  <PostCode> PostCode_Type </PostCode> [1]  ?
  <City> City_Type </City> [1]  ?
  <CountryCode> iso:CountryCode </CountryCode> [1]  ?
  <EmailAddress> EMail_Type </EmailAddress> [1]  ?
  <Phone> cm:PhoneNumber_Type </Phone> [1]  ?
</...>
```

Schema Component Representation

```
<xs:complexType name="Representative_Type">
  <xs:sequence>
    <xs:element name="RepresentativeID" type="RepresentativeID_Type"/>
    <xs:element name="Name" type="Name_Type"/>
    <xs:element name="Street" type="Street_Type"/>
    <xs:element name="PostCode" type="PostCode_Type"/>
    <xs:element name="City" type="City_Type"/>
    <xs:element name="CountryCode" type="iso:CountryCode"/>
    <xs:element name="EmailAddress" type="EMail_Type"/>
    <xs:element name="Phone" type="cm:PhoneNumber_Type"/>
  </xs:sequence>
</xs:complexType>
```

top

Complex Type: RepresentativeID_Type

Super-types:	xs:token < **RepresentativeID_Type** (by extension)
Sub-types:	None

Name	RepresentativeID_Type
Abstract	no

XML Instance Representation

```
<...
 issuedBy="iso:MSCountryCode [1]  ?"
 identificationType="RepresentativeIDType_Type [1]  ?"
 otherQlf="xs:token [0..1]  ?">
  xs:token
</...>
```

Schema Component Representation

```
<xs:complexType name="RepresentativeID_Type">
  <xs:simpleContent>
    <xs:extension base="xs:token">
      <xs:attribute name="issuedBy" type="iso:MSCountryCode"
      use="required"/>
      <xs:attribute name="identificationType"
      type="RepresentativeIDType_Type" use="required"/>
      <xs:attribute name="otherQlf" type="xs:token" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

top

Complex Type: VR_Declarant_Type

Super-types:	None
Sub-types:	None

Name	VR_Declarant_Type
Abstract	no

XML Instance Representation

```
<...>
   <VATNumber> BEVATNumber </VATNumber> [1] ?
   <Name> Name_Type </Name> [0..1] ?
   <Street> Street_Type </Street> [0..1] ?
   <PostCode> PostCode_Type </PostCode> [0..1] ?
   <City> City_Type </City> [0..1] ?
   <CountryCode> iso:BECountryCode </CountryCode> [0..1] ?
   <EmailAddress> EMail_Type </EmailAddress> [1] ?
   <Phone> cm:PhoneNumber_Type </Phone> [0..1] ?
</...>
```

Schema Component Representation

```
<xs:complexType name="VR_Declarant_Type">
   <xs:sequence>
      <xs:element name="VATNumber" type="BEVATNumber"/>
      <xs:element name="Name" type="Name_Type" minOccurs="0"/>
      <xs:element name="Street" type="Street_Type" minOccurs="0"/>
      <xs:element name="PostCode" type="PostCode_Type" minOccurs="0"/>
      <xs:element name="City" type="City_Type" minOccurs="0"/>
      <xs:element name="CountryCode" type="iso:BECountryCode"
      minOccurs="0"/>
      <xs:element name="EmailAddress" type="EMail_Type"/>
      <xs:element name="Phone" type="cm:PhoneNumber_Type" minOccurs="0"/>
   </xs:sequence>
   <-- Specificity vs. Declarant_Type: e-mail required + countrycode of
   address always "BE" -->
</xs:complexType>
```

top

Complex Type: VR_Representative_Type

Super-types:	None
Sub-types:	None

Name	VR_Representative_Type
Abstract	no
Documentation	Mandataire d'un envoi de type VATRefund ou modification du prorata VATRefund (pays de l'adresse limité aux Etats Membres)

XML Instance Representation

```
<...>
   <RepresentativeID> RepresentativeID_Type </RepresentativeID> [1] ?
   <Name> Name_Type </Name> [1] ?
   <Street> Street_Type </Street> [1]  ?
   <PostCode> PostCode_Type </PostCode> [1]  ?
   <City> City_Type </City> [1]  ?
   <CountryCode> iso:MSCountryCode </CountryCode> [1]  ?
   <EmailAddress> EMail_Type </EmailAddress> [1]  ?
   <Phone> cm:PhoneNumber_Type </Phone> [1]  ?
</...>
```

Schema Component Representation

```
<xs:complexType name="VR_Representative_Type">
  <xs:sequence>
    <xs:element name="RepresentativeID" type="RepresentativeID_Type"/>
    <xs:element name="Name" type="Name_Type"/>
    <xs:element name="Street" type="Street_Type"/>
    <xs:element name="PostCode" type="PostCode_Type"/>
    <xs:element name="City" type="City_Type"/>
    <xs:element name="CountryCode" type="iso:MSCountryCode"/>
    <xs:element name="EmailAddress" type="EMail_Type"/>
    <xs:element name="Phone" type="cm:PhoneNumber_Type"/>
  </xs:sequence>
</xs:complexType>
```

top

Simple Type: BEVATNumber

Super-types:	xs:string < **BEVATNumber** (by restriction)
Sub-types:	None

Name BEVATNumber

Content

- Base XSD Type: string
- *pattern* = [0-9]{10}

Schema Component Representation

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

top

Simple Type: City_Type

Super-types:	xs:string < **City_Type** (by restriction)
Sub-types:	None

Name City_Type

Content

- Base XSD Type: string
- *length* <= 184

Schema Component Representation

```
<xs:simpleType name="City_Type">
  <xs:restriction base="xs:string">
    <xs:maxLength value="184"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: Comment_Type

Super-types:	xs:string < **Comment_Type** (by restriction)
Sub-types:	None

Name Comment_Type

Content

- Base XSD Type: string

Schema Component Representation

```
<xs:simpleType name="Comment_Type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
```

top

Simple Type: DeclarantReference_Type

Super-types:	xs:token < **DeclarantReference_Type** (by restriction)
Sub-types:	None

Name DeclarantReference_Type

Content

- Base XSD Type: token
- *length* <= 14

Schema Component Representation

```
<xs:simpleType name="DeclarantReference_Type">
  <xs:restriction base="xs:token">
    <xs:maxLength value="14"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: EMail_Type

Super-types:	xs:token < **EMail_Type** (by restriction)
Sub-types:	None

Name	EMail_Type
Content	• Base XSD Type: token • *pattern* = ([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})
Documentation	An email address

Schema Component Representation

```
<xs:simpleType name="EMail_Type">
  <xs:restriction base="xs:token">
    <xs:pattern value="([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-
    zA-Z]{2,5})"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: EUVATNumber

Super-types:	xs:token < **EUVATNumber** (by restriction)
Sub-types:	None

Name	EUVATNumber
Content	• Base XSD Type: token • *length* <= 12

Schema Component Representation

```
<xs:simpleType name="EUVATNumber">
  <xs:restriction base="xs:token">
    <xs:maxLength value="12"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: FileTypeCode

Super-types:	xs:string < **FileTypeCode** (by restriction)
Sub-types:	None

Name	FileTypeCode
Content	- Base XSD Type: string - *value* comes from list: {'application/pdf'|'image/jpeg'|'image/tiff'}

Schema Component Representation

```
<xs:simpleType name="FileTypeCode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="application/pdf"/>
    <xs:enumeration value="image/jpeg"/>
    <xs:enumeration value="image/tiff"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: IntervatConsignmentReference_Type

Super-types:	xs:positiveInteger < **IntervatConsignmentReference_Type** (by restriction)
Sub-types:	None

Name	IntervatConsignmentReference_Type
Content	- Base XSD Type: positiveInteger

Schema Component Representation

```
<xs:simpleType name="IntervatConsignmentReference_Type">
  <xs:restriction base="xs:positiveInteger"/>
</xs:simpleType>
```

top

Simple Type: IntervatDeclarationReference_Type

Super-types:	xs:string < **IntervatDeclarationReference_Type** (by restriction)
Sub-types:	None

Name	IntervatDeclarationReference_Type
Content	- Base XSD Type: string - *pattern* = [0-9]+-[0-9]{10}-[0-9]{6}

Schema Component Representation

```
<xs:simpleType name="IntervatDeclarationReference_Type">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]+-[0-9]{10}-[0-9]{6}"/>
  </xs:restriction>
```

```
</xs:simpleType>
```

top

Simple Type: IntervatOrSTIRINTDeclReference_Type

Super-types:	xs:string < **IntervatOrSTIRINTDeclReference_Type** (by restriction)
Sub-types:	None

Name	IntervatOrSTIRINTDeclReference_Type
Content	- Base XSD Type: string - *pattern* = ([0-9]+-[0-9]{10}-[0-9]{6})|(BE[0-9]{16})

Schema Component Representation

```
<xs:simpleType name="IntervatOrSTIRINTDeclReference_Type">
  <xs:restriction base="xs:string">
    <xs:pattern value="([0-9]+-[0-9]{10}-[0-9]{6})|(BE[0-9]{16})"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: Month_Type

Super-types:	xs:integer < **Month_Type** (by restriction)
Sub-types:	None

Name	Month_Type
Content	- Base XSD Type: integer - 1 <= *value* <= 12

Schema Component Representation

```
<xs:simpleType name="Month_Type">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="12"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: Name_Type

Super-types:	xs:string < **Name_Type** (by restriction)

Sub-types:	None

Name Name_Type

Content

- Base XSD Type: string
- *length* <= 200

Schema Component Representation

```
<xs:simpleType name="Name_Type">
  <xs:restriction base="xs:string">
    <xs:maxLength value="200"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: PositiveAmount_Type

Super-types:	xs:decimal < **PositiveAmount_Type** (by restriction)
Sub-types:	None

Name PositiveAmount_Type

Content

- Base XSD Type: decimal
- 0 <= *value* <= 99999999999.99
- *no. of fraction digits* = 2

Schema Component Representation

```
<xs:simpleType name="PositiveAmount_Type">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="2"/>
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="99999999999.99"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: PostCode_Type

Super-types:	xs:string < **PostCode_Type** (by restriction)
Sub-types:	None

Name PostCode_Type

Content

- Base XSD Type: string

- *length* <= 15

Schema Component Representation

```
<xs:simpleType name="PostCode_Type">
   <xs:restriction base="xs:string">
      <xs:maxLength value="15"/>
   </xs:restriction>
</xs:simpleType>
```

top

Simple Type: Quarter_Type

Super-types:	xs:integer < **Quarter_Type** (by restriction)
Sub-types:	None

Name Quarter_Type

Content

- Base XSD Type: integer
- 1 <= *value* <= 4

Schema Component Representation

```
<xs:simpleType name="Quarter_Type">
   <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="4"/>
   </xs:restriction>
</xs:simpleType>
```

top

Simple Type: RepresentativeIDType_Type

Super-types:	xs:string < **RepresentativeIDType_Type** (by restriction)
Sub-types:	None

Name RepresentativeIDType_Type

Content

- Base XSD Type: string
- *value* comes from list: {'NVAT'|'TIN'|'other'}

Schema Component Representation

```
<xs:simpleType name="RepresentativeIDType_Type">
   <xs:restriction base="xs:string">
      <xs:enumeration value="NVAT"/>
      <xs:enumeration value="TIN"/>
      <xs:enumeration value="other"/>
   </xs:restriction>
```

```
</xs:simpleType>
```

top

Simple Type: RepresentativeReference_Type

Super-types:	xs:token < **RepresentativeReference_Type** (by restriction)
Sub-types:	None

Name	RepresentativeReference_Type
Content	• Base XSD Type: token • *length* <= 14

Schema Component Representation

```
<xs:simpleType name="RepresentativeReference_Type">
  <xs:restriction base="xs:token">
    <xs:maxLength value="14"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: RestrictedDate_Type

Super-types:	xs:date < **RestrictedDate_Type** (by restriction)
Sub-types:	None

Name	RestrictedDate_Type
Content	• Base XSD Type: date • *pattern* = \d{4}-\d{2}-\d{2}

Schema Component Representation

```
<xs:simpleType name="RestrictedDate_Type">
  <xs:restriction base="xs:date">
    <xs:pattern value="\d{4}-\d{2}-\d{2}"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: RestrictedDateTime_Type

Super-types:	xs:dateTime < **RestrictedDateTime_Type** (by restriction)
Sub-types:	None

Name	RestrictedDateTime_Type
Content	- Base XSD Type: dateTime - *pattern* = \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}

Schema Component Representation

```
<xs:simpleType name="RestrictedDateTime_Type">
  <xs:restriction base="xs:dateTime">
    <xs:pattern value="\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: SignedAmount_Type

Super-types:	xs:decimal < **SignedAmount_Type** (by restriction)
Sub-types:	None

Name	SignedAmount_Type
Content	- Base XSD Type: decimal - -99999999999.99 <= *value* <= 99999999999.99 - *no. of fraction digits* = 2

Schema Component Representation

```
<xs:simpleType name="SignedAmount_Type">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="2"/>
    <xs:minInclusive value="-99999999999.99"/>
    <xs:maxInclusive value="99999999999.99"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: STIRINTReference_Type

Super-types:	xs:string < **STIRINTReference_Type** (by restriction)
Sub-types:	None

Name	STIRINTReference_Type
Content	- Base XSD Type: string - *pattern* = BE[0-9]{16}

Schema Component Representation

```
<xs:simpleType name="STIRINTReference_Type">
   <xs:restriction base="xs:string">
      <xs:pattern value="BE[0-9]{16}"/>
   </xs:restriction>
</xs:simpleType>
```

top

Simple Type: Street_Type

Super-types:	xs:string < **Street_Type** (by restriction)
Sub-types:	None

Name Street_Type

Content

- Base XSD Type: string
- *length* <= 200

Schema Component Representation

```
<xs:simpleType name="Street_Type">
   <xs:restriction base="xs:string">
      <xs:maxLength value="200"/>
   </xs:restriction>
</xs:simpleType>
```

top

Simple Type: UnlimitedSignedAmount_Type

Super-types:	xs:decimal < **UnlimitedSignedAmount_Type** (by restriction)
Sub-types:	None

Name UnlimitedSignedAmount_Type

Content

- Base XSD Type: decimal
- *no. of fraction digits* = 2

Schema Component Representation

```
<xs:simpleType name="UnlimitedSignedAmount_Type">
   <xs:restriction base="xs:decimal">
      <xs:fractionDigits value="2"/>
   </xs:restriction>
</xs:simpleType>
```

top

Simple Type: Year_Type

Super-types:	xs:unsignedLong < **Year_Type** (by restriction)
Sub-types:	None

Name	Year_Type
Content	- Base XSD Type: unsignedLong - *total no. of digits* = 4

Schema Component Representation

```
<xs:simpleType name="Year_Type">
  <xs:restriction base="xs:unsignedLong">
    <xs:totalDigits value="4"/>
  </xs:restriction>
</xs:simpleType>
```

top

Simple Type: YesNo_Type

Super-types:	xs:string < **YesNo_Type** (by restriction)
Sub-types:	None

Name	YesNo_Type
Content	- Base XSD Type: string - *value* comes from list: {'YES'|'NO'}

Schema Component Representation

```
<xs:simpleType name="YesNo_Type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="YES"/>
    <xs:enumeration value="NO"/>
  </xs:restriction>
</xs:simpleType>
```

top

Generated by xs3p (old link) . Last modified: 12/09/2011 11:12:01