

# XML Schema Documentation

---

## Table of Contents

- [Schema Document Properties](#)
- [Global Definitions](#)
  - [Complex Type: AddressStruct\\_Type](#)
  - [Complex Type: EUMoneyAmount\\_Type](#)
  - [Complex Type: Header\\_Type](#)
  - [Complex Type: I18nString\\_Type](#)
  - [Complex Type: Individual\\_Type](#)
  - [Complex Type: IndividualWithAddress\\_Type](#)
  - [Complex Type: IndividualWithBirthDate\\_Type](#)
  - [Complex Type: IndivPersBirthData\\_Type](#)
  - [Complex Type: IndivPersData\\_Type](#)
  - [Complex Type: LegalEntity\\_Type](#)
  - [Complex Type: MoneyAmount\\_Type](#)
  - [Complex Type: Name\\_Type](#)
  - [Complex Type: NameReducedFix\\_Type](#)
  - [Complex Type: NameStruct\\_Type](#)
  - [Complex Type: NVPair\\_Type](#)
  - [Complex Type: OrderedString\\_Type](#)
  - [Complex Type: Period\\_Type](#)
  - [Complex Type: Person\\_Type](#)
  - [Complex Type: ShortAddressStruct\\_Type](#)
  - [Simple Type: AddressFree\\_Type](#)
  - [Simple Type: AddressStatus\\_Type](#)
  - [Simple Type: AmountValue\\_Type](#)
  - [Simple Type: ApplicationId\\_Type](#)
  - [Simple Type: CountryIsoCodeAlpha2\\_Type](#)
  - [Simple Type: CountryIsoCodeNum3\\_Type](#)
  - [Simple Type: CurrencyIsoCode\\_Type](#)
  - [Simple Type: EMail\\_Type](#)
  - [Simple Type: EUCountryIsoCodeAlpha2\\_Type](#)
  - [Simple Type: EUCountryIsoCodeAlpha2List\\_Type](#)
  - [Simple Type: EUCountryIsoCodeNum3\\_Type](#)
  - [Simple Type: EUCountryIsoCodeNum3List\\_Type](#)
  - [Simple Type: EUCurrencyIsoCode\\_Type](#)
  - [Simple Type: EULanguageCode\\_Type](#)
  - [Simple Type: EULanguageList\\_Type](#)
  - [Simple Type: Gender\\_Type](#)
  - [Simple Type: IBAN\\_Type](#)
  - [Simple Type: MsgId\\_Type](#)
  - [Simple Type: NameFree\\_Type](#)
  - [Simple Type: nameType\\_Type](#)
  - [Simple Type: NumberLength3\\_Type](#)
  - [Simple Type: PhoneNumber\\_Type](#)
  - [Simple Type: Quarter\\_Type](#)
  - [Simple Type: VATNumber\\_Type](#)
  - [Simple Type: VATNumberOrTIN\\_Type](#)
  - [Simple Type: VatRate\\_Type](#)

[top](#)

---

## Schema Document Properties

<b>Target Namespace</b>	<a href="#">urn:ec.europa.eu:taxud:fiscalis:common:v1</a>
<b>Version</b>	1.6

<b>Language</b>	en
<b>Element and Attribute Namespaces</b>	<ul style="list-style-type: none"><li>• Global element and attribute declarations belong to this schema's target namespace.</li><li>• By default, local element declarations belong to this schema's target namespace.</li><li>• By default, local attribute declarations have no namespace.</li></ul>
<b>Schema Composition</b>	<ul style="list-style-type: none"><li>• This schema includes components from the following schema document(s):<ul style="list-style-type: none"><li>◦ isotypes_v1.xsd</li></ul></li></ul>
<b>Documentation</b>	List of Goods Description types and sub-types === HISTORY === Version 1.6 - add Turkish to the list of languages Version 1.5 - reduce size of VATNumberOrTIN_Type to 20 characters from 50 characters (20/03/2009) Version 1.4 - first release of the schema to the Member States for VAT Refund

## Declared Namespaces

Prefix	Namespace
Default namespace	<a href="urn:ec.europa.eu:taxud:fiscalis:common:v1">urn:ec.europa.eu:taxud:fiscalis:common:v1</a>
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
cm	<a href="urn:ec.europa.eu:taxud:fiscalis:common:v1">urn:ec.europa.eu:taxud:fiscalis:common:v1</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

### Schema Component Representation

```
<xs:schema xml:lang="en"
targetNamespace="urn:ec.europa.eu:taxud:fiscalis:common:v1"
elementFormDefault="qualified" version="1.6">
  <xs:include schemaLocation="isotypes_v1.xsd"/>
  ...
</xs:schema>
```

[top](#)

## Global Definitions

### Complex Type: **AddressStruct\_Type**

<i>Super-types:</i>	None
<i>Sub-types:</i>	<ul style="list-style-type: none"><li>• <a href="#">ShortAddressStruct_Type</a> (by restriction)</li></ul>

<b>Name</b>	AddressStruct_Type
<b><u>Abstract</u></b>	no
<b>Documentation</b>	Structure of the address for a party broken down into its logical parts, recommended for easy matching. The 'City' element is the only required subelement. All of the subelements are simple text - data type 'string'.

### XML Instance Representation

```

<...
  Allow any attributes from any namespace (strict validation).
>
  <Street> xs:string </Street> [0..1]
  <BuildingIdentifier> xs:string </BuildingIdentifier> [0..1]
  <SuiteIdentifier> xs:string </SuiteIdentifier> [0..1]
  <FloorIdentifier> xs:string </FloorIdentifier> [0..1]
  <DistrictName> xs:string </DistrictName> [0..1]
  <POB> xs:string </POB> [0..1]
  <PostCode> xs:string </PostCode> [0..1]
  <City> xs:string </City> [1]
  <CountrySubentity> xs:string </CountrySubentity> [0..1]
  <OtherLocalId> xs:string </OtherLocalId> [0..1]
</...>

```

### Schema Component Representation

```

<xs:complexType name="AddressStruct_Type">
  <xs:sequence>
    <xs:element name="Street" type="xs:string" minOccurs="0"/>
    <xs:element name="BuildingIdentifier" type="xs:string" minOccurs="0"/>
    <xs:element name="SuiteIdentifier" type="xs:string" minOccurs="0"/>
    <xs:element name="FloorIdentifier" type="xs:string" minOccurs="0"/>
    <xs:element name="DistrictName" type="xs:string" minOccurs="0"/>
    <xs:element name="POB" type="xs:string" minOccurs="0"/>
    <xs:element name="PostCode" type="xs:string" minOccurs="0"/>
    <xs:element name="City" type="xs:string"/>
    <xs:element name="CountrySubentity" type="xs:string" minOccurs="0"/>
    <xs:element name="OtherLocalId" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute/>
</xs:complexType>

```

[top](#)

## Complex Type: **EUMoneyAmount\_Type**

**Super-types:** [xs:decimal](#) < [AmountValue\\_Type](#) (by restriction) < [MoneyAmount\\_Type](#) (by extension) < **EUMoneyAmount\_Type** (by restriction)

**Sub-types:** None

<b>Name</b>	EUMoneyAmount_Type
<b>Abstract</b>	no
<b>Documentation</b>	An amount for the EU currencies

### XML Instance Representation

```

<...
  currency="EUCurrencyIsoCode_Type [1]">
    MoneyAmount_Type
</...>

```

### Schema Component Representation

```

<xs:complexType name="EUMoneyAmount_Type">
  <xs:simpleContent>
    <xs:restriction base="MoneyAmount_Type">
      <xs:attribute name="currency" type="EUCurrencyIsoCode_Type"
        use="required"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>

```

## Complex Type: **Header\_Type**

<i>Super-types:</i>	None
<i>Sub-types:</i>	None

<b>Name</b>	Header_Type
<b><u>Abstract</u></b>	no
<b>Documentation</b>	Header of generic tax message

### XML Instance Representation

```
<...>
  <OriginatingCountry> EUCountryIsoCodeAlpha2_Type </OriginatingCountry> [1]
  <DestinationCountries> EUCountryIsoCodeAlpha2List_Type
</DestinationCountries> [1]
  <MessageId> MsgId_Type </MessageId> [1]
  <CorrelationId> MsgId_Type </CorrelationId> [0..1]
  <Timestamp> xs:dateTime </Timestamp> [1]
  <ResponseRequired> xs:date </ResponseRequired> [0..1]
  <Language> xs:language </Language> [0..1] ?
</...>
```

### Schema Component Representation

```
<xs:complexType name="Header_Type">
  <xs:sequence>
    <xs:element name="OriginatingCountry"
      type="EUCountryIsoCodeAlpha2_Type" />
    <xs:element name="DestinationCountries"
      type="EUCountryIsoCodeAlpha2List_Type" />
    <xs:element name="MessageId" type="MsgId_Type" />
    <xs:element name="CorrelationId" type="MsgId_Type" minOccurs="0" />
    <xs:element name="Timestamp" type="xs:dateTime" />
    <xs:element name="ResponseRequired" type="xs:date" minOccurs="0" />
    <xs:element name="Language" type="xs:language" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

## Complex Type: **I18nString\_Type**

<i>Super-types:</i>	<a href="#">xs:string</a> < <b>I18nString_Type</b> (by extension)
<i>Sub-types:</i>	None

<b>Name</b>	I18nString_Type
<b><u>Abstract</u></b>	no
<b>Documentation</b>	Internationalised string

### XML Instance Representation

```
<...
  lang="xs:language [0..1]">
    xs:string
  </...>
```

```
</...>
```

### Schema Component Representation

```
<xs:complexType name="I18nString_Type">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="lang" type="xs:language" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

[top](#)

## Complex Type: **Individual\_Type**

*Super-types:* None

*Sub-types:*

- [IndividualWithAddress\\_Type](#) (by restriction)
- [IndividualWithBirthDate\\_Type](#) (by restriction)

**Name** Individual\_Type

**Abstract** no

**Documentation** An individual

### XML Instance Representation

```
<...
  indivQlf="xs:string [0..1]">
    <Name> NameFree_Type </Name> [0..1]
    <BirthDate> xs:date </BirthDate> [0..1]
    <Address> cm:AddressFree_Type </Address> [0..1]
  </...>
```

### Schema Component Representation

```
<xs:complexType name="Individual_Type">
  <xs:sequence>
    <xs:element name="Name" type="NameFree_Type" minOccurs="0"/>
    <xs:element name="BirthDate" type="xs:date" minOccurs="0"/>
    <xs:element name="Address" type="cm:AddressFree_Type" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="indivQlf" type="xs:string" use="optional"/>
</xs:complexType>
```

[top](#)

## Complex Type: **IndividualWithAddress\_Type**

*Super-types:* [Individual\\_Type](#) < IndividualWithAddress\_Type (by restriction)

*Sub-types:* None

**Name** IndividualWithAddress\_Type

**Abstract** no

### XML Instance Representation

```
<...
  indivQlf="xs:string [0..1]">
    <Name> NameFree Type </Name> [0..1]
    <Address> cm:AddressFree Type </Address> [0..1]
  </...>
```

#### Schema Component Representation

```
<xs:complexType name="IndividualWithAddress_Type">
  <xs:complexContent>
    <xs:restriction base="Individual_Type">
      <xs:sequence>
        <xs:element name="Name" type="NameFree Type" minOccurs="0"/>
        <xs:element name="BirthDate" type="xs:date" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="Address" type="cm:AddressFree Type"
          minOccurs="0"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

[top](#)

### Complex Type: **IndividualWithBirthDate\_Type**

Super-types: [Individual\\_Type](#) < IndividualWithBirthDate\_Type (by restriction)

Sub-types: None

Name IndividualWithBirthDate\_Type

**Abstract** no

#### XML Instance Representation

```
<...
  indivQlf="xs:string [0..1]">
    <Name> NameFree Type </Name> [0..1]
    <BirthDate> xs:date </BirthDate> [0..1]
  </...>
```

#### Schema Component Representation

```
<xs:complexType name="IndividualWithBirthDate_Type">
  <xs:complexContent>
    <xs:restriction base="Individual_Type">
      <xs:sequence>
        <xs:element name="Name" type="NameFree Type" minOccurs="0"/>
        <xs:element name="BirthDate" type="xs:date" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="Address" type="cm:AddressFree Type"
          minOccurs="0" maxOccurs="0"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

[top](#)

### Complex Type: **IndivPersBirthData\_Type**

**Super-types:** [IndivPersData\\_Type](#) < **IndivPersBirthData\_Type** (by restriction)

**Sub-types:** None

**Name** IndivPersBirthData\_Type  
**Abstract** no  
**Documentation** A simplified version of IndivPersData\_Type

#### XML Instance Representation

```
<...>
  <BirthDate> xs:date </BirthDate> [0..1]
  <BirthCity> xs:string </BirthCity> [0..1]
  <BirthCountryCode> xs:string </BirthCountryCode> [0..1]
</...>
```

#### Schema Component Representation

```
<xs:complexType name="IndivPersBirthData_Type">
  <xs:complexContent>
    <xs:restriction base="IndivPersData_Type">
      <xs:sequence>
        <xs:element name="Gender" type="Gender_Type" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="Nationality" type="CountryIsoCodeAlpha2_Type"
          minOccurs="0" maxOccurs="0"/>
        <xs:element name="BirthDate" type="xs:date" minOccurs="0"/>
        <xs:element name="BirthCity" type="xs:string" minOccurs="0"/>
        <xs:element name="BirthCitySubentity" type="xs:string"
          minOccurs="0" maxOccurs="0"/>
        <xs:element name="BirthCountryCode" type="xs:string"
          minOccurs="0"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

[top](#)

## Complex Type: **IndivPersData\_Type**

**Super-types:** None

**Sub-types:**

- [IndivPersBirthData\\_Type](#) (by restriction)

**Name** IndivPersData\_Type  
**Abstract** no  
**Documentation** Data (other than Name and Address) to describe and identify an Individual.

#### XML Instance Representation

```
<...>
  <Gender> Gender_Type </Gender> [0..1]
  <Nationality> CountryIsoCodeAlpha2_Type </Nationality> [0..1]
  <BirthDate> xs:date </BirthDate> [0..1]
  <BirthCity> xs:string </BirthCity> [0..1]
  <BirthCitySubentity> xs:string </BirthCitySubentity> [0..1]
</...>
```

```
<BirthCountryCode> xs:string </BirthCountryCode> [0..1]
</...>
```

### Schema Component Representation

```
<xs:complexType name="IndivPersData_Type">
  <xs:sequence>
    <xs:element name="Gender" type="Gender_Type" minOccurs="0"/>
    <xs:element name="Nationality" type="CountryIsoCodeAlpha2_Type"
      minOccurs="0"/>
    <xs:element name="BirthDate" type="xs:date" minOccurs="0"/>
    <xs:element name="BirthCity" type="xs:string" minOccurs="0"/>
    <xs:element name="BirthCitySubentity" type="xs:string" minOccurs="0"/>
    <xs:element name="BirthCountryCode" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

[top](#)

## Complex Type: **LegalEntity\_Type**

*Super-types:* None

*Sub-types:* None

**Name** LegalEntity\_Type

**Abstract** no

**Documentation** A legal entity

### XML Instance Representation

```
<...>
  <Name> NameFree_Type </Name> [1]
  <LegalStatus> xs:string </LegalStatus> [1]
  <Address> AddressStruct_Type </Address> [1]
</...>
```

### Schema Component Representation

```
<xs:complexType name="LegalEntity_Type">
  <xs:sequence>
    <xs:element name="Name" type="NameFree_Type" />
    <xs:element name="LegalStatus" type="xs:string" />
    <xs:element name="Address" type="AddressStruct_Type" />
  </xs:sequence>
</xs:complexType>
```

[top](#)

## Complex Type: **MoneyAmount\_Type**

*Super-types:* [xs:decimal](#) < [AmountValue\\_Type](#) (by restriction) < **MoneyAmount\_Type** (by extension)

*Sub-types:* None

**Name** MoneyAmount\_Type

**Abstract** no



**Documentation**

An amount: a value with a currency attribute

**XML Instance Representation**

```
<...
  currency="CurrencyIsoCode Type [1]">
    AmountValue_Type
</...>
```

**Schema Component Representation**

```
<xs:complexType name="MoneyAmount_Type">
  <xs:simpleContent>
    <xs:extension base="AmountValue_Type">
      <xs:attribute name="currency" type="CurrencyIsoCode_Type"
        use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

[top](#)**Complex Type: Name\_Type**

Super-types: None

Sub-types: None

Name Name\_Type

**Abstract** no**XML Instance Representation**

```
<...
  nameType="nameType_Type [0..1]">
    Start Choice [1]
    <NameFree> xs:string </NameFree> [1]
    <NameStruct> NameStruct_Type </NameStruct> [1]
    <NameFree> xs:string </NameFree> [0..1]
    End Choice
  </...>
```

**Schema Component Representation**

```
<xs:complexType name="Name_Type">
  <xs:choice>
    <xs:element name="NameFree" type="xs:string"/>
    <xs:sequence>
      <xs:element name="NameStruct" type="NameStruct_Type"/>
      <xs:element name="NameFree" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:choice>
  <xs:attribute name="nameType" type="nameType_Type" use="optional"/>
</xs:complexType>
```

[top](#)**Complex Type: NameReducedFix\_Type**

**Super-types:** [NameStruct\\_Type](#) < **NameReducedFix\_Type** (by restriction)

**Sub-types:** None

**Name** NameReducedFix\_Type

**Abstract** no

#### XML Instance Representation

```
<...>
  <FirstName> xs:string </FirstName> [1]
  <LastName> xs:string </LastName> [1]
  <MaidenName> xs:string </MaidenName> [0..1]
</...>
```

#### Schema Component Representation

```
<xs:complexType name="NameReducedFix_Type">
  <xs:complexContent>
    <xs:restriction base="NameStruct_Type">
      <xs:sequence>
        <xs:element name="PrecedingTitle" type="xs:string" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="Title" type="xs:string" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="FirstName" type="xs:string"/>
        <xs:element name="MiddleName" type="xs:string" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="NamePrefix" type="xs:string" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="LastName" type="xs:string"/>
        <xs:element name="GenerationIdentifier" type="xs:string"
          minOccurs="0" maxOccurs="0"/>
        <xs:element name="Suffix" type="xs:string" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="GeneralSuffix" type="xs:string" minOccurs="0"
          maxOccurs="0"/>
        <xs:element name="MaidenName" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

[top](#)

### Complex Type: **NameStruct\_Type**

**Super-types:** None

**Sub-types:**

- [NameReducedFix\\_Type](#) (by restriction)

**Name** NameStruct\_Type

**Abstract** no

#### XML Instance Representation

```
<...>
  <PrecedingTitle> xs:string </PrecedingTitle> [0..1]
  <Title> xs:string </Title> [0..*]
</...>
```

```

<FirstName> xs:string </FirstName> [1]
<MiddleName> xs:string </MiddleName> [0..*]
<NamePrefix> xs:string </NamePrefix> [0..1]
<LastName> xs:string </LastName> [1]
<GenerationIdentifier> xs:string </GenerationIdentifier> [0..*]
<Suffix> xs:string </Suffix> [0..*]
<GeneralSuffix> xs:string </GeneralSuffix> [0..1]
<MaidenName> xs:string </MaidenName> [0..1]
</...>

```

### Schema Component Representation

```

<xs:complexType name="NameStruct_Type">
  <xs:sequence>
    <xs:element name="PrecedingTitle" type="xs:string" minOccurs="0"/>
    <xs:element name="Title" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="FirstName" type="xs:string" />
    <xs:element name="MiddleName" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="NamePrefix" type="xs:string" minOccurs="0"/>
    <xs:element name="LastName" type="xs:string" />
    <xs:element name="GenerationIdentifier" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="Suffix" type="xs:string" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="GeneralSuffix" type="xs:string" minOccurs="0"/>
    <xs:element name="MaidenName" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

```

[top](#)

## Complex Type: **NVPair\_Type**

**Super-types:** [xs:string](#) < **NVPair\_Type** (by extension)

**Sub-types:** None

**Name** NVPair\_Type

**Abstract** no

**Documentation** A type that holds a name value pair. The name is stored in the 'name' attribute, and the value as a string in the element value itself

### XML Instance Representation

```

<...
  name="xs:token [1]">
    xs:string
  </...>

```

### Schema Component Representation

```

<xs:complexType name="NVPair_Type">
  <xs:simpleContent>
    <xs:extension base="xs:string"
      <xs:attribute name="name" type="xs:token" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

[top](#)

## Complex Type: **OrderedString\_Type**

*Super-types:* [xs:string](#) < **OrderedString\_Type** (by extension)

*Sub-types:* None

**Name** OrderedString\_Type

**Abstract** no

**Documentation** A string with an attribute that allows ordering in a list.

### XML Instance Representation

```
<...  
  order="xs:byte [1]">  
    xs:string  
</...>
```

### Schema Component Representation

```
<xs:complexType name="OrderedString_Type">  
  <xs:simpleContent>  
    <xs:extension base="xs:string">  
      <xs:attribute name="order" type="xs:byte" use="required"/>  
    </xs:extension>  
  </xs:simpleContent>  
</xs:complexType>
```

[top](#)

## Complex Type: **Period\_Type**

*Super-types:* None

*Sub-types:* None

**Name** Period\_Type

**Abstract** no

**Documentation** Represents a period of time specified by a start date and an end date.

### XML Instance Representation

```
<...>  
  <StartDate> xs:date </StartDate> [1]  
  <EndDate> xs:date </EndDate> [1]  
</...>
```

### Schema Component Representation

```
<xs:complexType name="Period_Type">  
  <xs:sequence>  
    <xs:element name="StartDate" type="xs:date" />  
    <xs:element name="EndDate" type="xs:date" />  
  </xs:sequence>  
</xs:complexType>
```

[top](#)

**Complex Type: Person\_Type***Super-types:* None*Sub-types:* None**Name** Person\_Type**Abstract** no**Documentation** A person**XML Instance Representation**

```

<...>
  <Name> NameStruct_Type </Name> [1]
  <IndivPersData> IndivPersData_Type </IndivPersData> [1]
  <Address> AddressStruct_Type </Address> [1]
</...>

```

**Schema Component Representation**

```

<xs:complexType name="Person_Type">
  <xs:sequence>
    <xs:element name="Name" type="NameStruct_Type"/>
    <xs:element name="IndivPersData" type="IndivPersData_Type"/>
    <xs:element name="Address" type="AddressStruct_Type"/>
  </xs:sequence>
</xs:complexType>

```

[top](#)**Complex Type: ShortAddressStruct\_Type***Super-types:* [AddressStruct\\_Type](#) < ShortAddressStruct\_Type (by restriction)*Sub-types:* None**Name** ShortAddressStruct\_Type**Abstract** no**Documentation** A shortenedform of Address\_Fix with a mandatory status attribute**XML Instance Representation**

```

<...
  status="AddressStatus_Type [1]">
    <!-- 'cm:AddressStruct_Type' super type was not found in this schema. Some
    elements and attributes may be missing. -->
    <Street> xs:string </Street> [0..1]
    <BuildingIdentifier> xs:string </BuildingIdentifier> [0..1]
    <SuiteIdentifier> xs:string </SuiteIdentifier> [0..1]
    <PostCode> xs:string </PostCode> [0..1]
    <City> xs:string </City> [1]
  </...>

```

**Schema Component Representation**

```

<xs:complexType name="ShortAddressStruct_Type">
  <xs:complexContent>
    <xs:restriction base="cm:AddressStruct_Type">
      <xs:sequence>

```

```

<xs:element name="Street" type="xs:string" minOccurs="0"/>
<xs:element name="BuildingIdentifier" type="xs:string"
minOccurs="0"/>
<xs:element name="SuiteIdentifier" type="xs:string" minOccurs="0"/>
<xs:element name="FloorIdentifier" type="xs:string" minOccurs="0"
maxOccurs="0"/>
<xs:element name="DistrictName" type="xs:string" minOccurs="0"
maxOccurs="0"/>
<xs:element name="POB" type="xs:string" minOccurs="0"
maxOccurs="0"/>
<xs:element name="PostCode" type="xs:string" minOccurs="0"/>
<xs:element name="City" type="xs:string"/>
<xs:element name="CountrySubentity" type="xs:string" minOccurs="0"
maxOccurs="0"/>
<-- <xs:element name="Country" type="CountryIsoCodeAlpha2_Type"
minOccurs="0"/> -->
</xs:sequence>
<xs:attribute name="status" type="AddressStatus_Type"
use="required"/>
</xs:restriction>
</xs:complexContent>
</xs:complexType>

```

[top](#)

## Simple Type: AddressFree\_Type

**Super-types:** [xs:string](#) < **AddressFree\_Type** (by restriction)

**Sub-types:** None

**Name** AddressFree\_Type

**Content**

- Base XSD Type: string

**Documentation** An unstructured address

### Schema Component Representation

```

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

```

[top](#)

## Simple Type: AddressStatus\_Type

**Super-types:** [xs:string](#) < **AddressStatus\_Type** (by restriction)

**Sub-types:** None

**Name** AddressStatus\_Type

**Content**

- Base XSD Type: string
- *value* comes from list: {'known'|'assumed'}

### Schema Component Representation

```
<xs:simpleType name="AddressStatus_Type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="known"/>
    <xs:enumeration value="assumed"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **AmountValue\_Type**

**Super-types:** [xs:decimal](#) < **AmountValue\_Type** (by restriction)

**Sub-types:**

- [MoneyAmount\\_Type](#) (by extension)

**Name** AmountValue\_Type

**Content**

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

**Documentation** A decimal amount, restricted to two decimal places

### Schema Component Representation

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

[top](#)

## Simple Type: **ApplicationId\_Type**

**Super-types:** [xs:token](#) < **ApplicationId\_Type** (by restriction)

**Sub-types:** None

**Name** ApplicationId\_Type

**Content**

- Base XSD Type: token
- *value* comes from list: {'ecommerce'}

**Documentation** Application Identifier

### Schema Component Representation

```
<xs:simpleType name="ApplicationId_Type">
  <xs:restriction base="xs:token">
    <xs:enumeration value="ecommerce"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **CountryIsoCodeAlpha2\_Type**

**Super-types:** [xs:NMTOKEN](#) < **CountryIsoCodeAlpha2\_Type** (by restriction)

**Sub-types:** None

**Name** CountryIsoCodeAlpha2\_Type

**Content**

- Base XSD Type: NMTOKEN
- *pattern* = [A-Z]{2}

**Documentation** ISO 3166 alpha 2 country code

### Schema Component Representation

```
<xs:simpleType name="CountryIsoCodeAlpha2_Type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="[A-Z]{2}" />
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **CountryIsoCodeNum3\_Type**

**Super-types:** [xs:integer](#) < **CountryIsoCodeNum3\_Type** (by restriction)

**Sub-types:**

- [EUCountryIsoCodeNum3\\_Type](#) (by restriction)

**Name** CountryIsoCodeNum3\_Type

**Content**

- Base XSD Type: integer
- *total no. of digits* = 3

**Documentation** ISO 3166 numeric country code

### Schema Component Representation

```
<xs:simpleType name="CountryIsoCodeNum3_Type">
  <xs:restriction base="xs:integer">
    <xs:totalDigits value="3" />
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **CurrencyIsoCode\_Type**

**Super-types:** [xs:NMTOKEN](#) < **CurrencyIsoCode\_Type** (by restriction)

**Sub-types:**

- [EUCurrencyIsoCode\\_Type](#) (by restriction)



Name	CurrencyIsoCode_Type
Content	<ul style="list-style-type: none"><li>• Base XSD Type: NMTOKEN</li><li>• <i>pattern</i> = [A-Z]{3}</li></ul>
Documentation	ISO 4217 currency code

#### Schema Component Representation

```
<xs:simpleType name="CurrencyIsoCode_Type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="[A-Z]{3}" />
  </xs:restriction>
</xs:simpleType>
```

[top](#)

### Simple Type: EMail\_Type

Super-types: [xs:token](#) < **EMail\_Type** (by restriction)

Sub-types: None

Name	EMail_Type
Content	<ul style="list-style-type: none"><li>• Base XSD Type: token</li><li>• <i>pattern</i> = ([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5})</li></ul>
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: EUCountryIsoCodeAlpha2\_Type

Super-types: [MSCountryCode\\_Type](#) < **EUCountryIsoCodeAlpha2\_Type** (by restriction)

Sub-types: None

Name	EUCountryIsoCodeAlpha2_Type
Content	<ul style="list-style-type: none"><li>• 'MSCountryCode_Type' super type was not found in this schema. Its facets could not be printed out.</li></ul>

**Documentation**

The set of ISO 3166 alpha 2 country code values for the EU member states, with the exception of the Hellenic Republic which is represented by a non-standard code.

**Schema Component Representation**

```
<xs:simpleType name="EUCountryIsoCodeAlpha2_Type">
  <xs:restriction base="MSCountryCode_Type"/>
</xs:simpleType>
```

[top](#)**Simple Type: EUCountryIsoCodeAlpha2List\_Type**

<i>Super-types:</i>	None
<i>Sub-types:</i>	None

**Name**

EUCountryIsoCodeAlpha2List\_Type

**Content**

- List of: [EUCountryIsoCodeAlpha2\\_Type](#)

**Documentation**

List of EU country codes

**Schema Component Representation**

```
<xs:simpleType name="EUCountryIsoCodeAlpha2List_Type">
  <xs:list itemType="EUCountryIsoCodeAlpha2_Type"/>
</xs:simpleType>
```

[top](#)**Simple Type: EUCountryIsoCodeNum3\_Type**

<i>Super-types:</i>	<a href="#">xs:integer</a> < <a href="#">CountryIsoCodeNum3_Type</a> (by restriction) < <b>EUCountryIsoCodeNum3_Type</b> (by restriction)
<i>Sub-types:</i>	None

**Name**

EUCountryIsoCodeNum3\_Type

**Content**

- Base XSD Type: integer
- *total no. of digits* = 3
- *value* comes from list:  
{'040'|'056'|'100'|'196'|'203'|'208'|'233'|'246'|'250'|'276'|'300'|'348'|'372'|'380'|'428'|'440'|'4

**Documentation**

The set of ISO 3166 numeric country code values for the EU member states

**Schema Component Representation**

```
<xs:simpleType name="EUCountryIsoCodeNum3_Type">
  <xs:restriction base="CountryIsoCodeNum3_Type">
    <xs:enumeration value="040"/>
    <xs:enumeration value="056"/>
    <xs:enumeration value="100"/>
    <xs:enumeration value="196"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:enumeration value="203"/>
<xs:enumeration value="208"/>
<xs:enumeration value="233"/>
<xs:enumeration value="246"/>
<xs:enumeration value="250"/>
<xs:enumeration value="276"/>
<xs:enumeration value="300"/>
<xs:enumeration value="348"/>
<xs:enumeration value="372"/>
<xs:enumeration value="380"/>
<xs:enumeration value="428"/>
<xs:enumeration value="440"/>
<xs:enumeration value="442"/>
<xs:enumeration value="470"/>
<xs:enumeration value="528"/>
<xs:enumeration value="616"/>
<xs:enumeration value="620"/>
<xs:enumeration value="642"/>
<xs:enumeration value="703"/>
<xs:enumeration value="705"/>
<xs:enumeration value="724"/>
<xs:enumeration value="752"/>
<xs:enumeration value="826"/>
</xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **EUCountryIsoCodeNum3List\_Type**

*Super-types:* None

*Sub-types:* None

**Name** EUCountryIsoCodeNum3List\_Type

**Content**

- List of: [EUCountryIsoCodeNum3\\_Type](#)

**Documentation** List of EU country code numbers

### Schema Component Representation

```
<xs:simpleType name="EUCountryIsoCodeNum3List_Type">
  <xs:list itemType="EUCountryIsoCodeNum3_Type"/>
</xs:simpleType>
```

[top](#)

## Simple Type: **EUCurrencyIsoCode\_Type**

*Super-types:* [xs:NMTOKEN](#) < [CurrencyIsoCode\\_Type](#) (by restriction) < **EUCurrencyIsoCode\_Type** (by restriction)

*Sub-types:* None

**Name** EUCurrencyIsoCode\_Type

**Content**

- Base XSD Type: NMTOKEN

- *pattern* = [A-Z]{3}
- *value* comes from list:  
{'BGN'|'CYP'|'CZK'|'DKK'|'EEK'|'EUR'|'GBP'|'HUF'|'LTL'|'LVL'|'MTL'|'PLN'|'ROL'|'SEK'|'SK

**Documentation** The set of currency code values for the EU member states. Currencies that have been legal in recent years are included, even if replaced by the Euro.

#### Schema Component Representation

```
<xs:simpleType name="EUCurrencyIsoCode_Type">
  <xs:restriction base="CurrencyIsoCode_Type">
    <xs:enumeration value="BGN"/>
    <xs:enumeration value="CYP"/>
    <xs:enumeration value="CZK"/>
    <xs:enumeration value="DKK"/>
    <xs:enumeration value="EEK"/>
    <xs:enumeration value="EUR"/>
    <xs:enumeration value="GBP"/>
    <xs:enumeration value="HUF"/>
    <xs:enumeration value="LTL"/>
    <xs:enumeration value="LVL"/>
    <xs:enumeration value="MTL"/>
    <xs:enumeration value="PLN"/>
    <xs:enumeration value="ROL"/>
    <xs:enumeration value="SEK"/>
    <xs:enumeration value="SKK"/>
    <xs:enumeration value="SIT"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)

### Simple Type: **EULanguageCode\_Type**

**Super-types:** [xs:string](#) < **EULanguageCode\_Type** (by restriction)

**Sub-types:** None

**Name** EULanguageCode\_Type

**Content**

- Base XSD Type: string
- *value* comes from list:  
{'bg'|'cs'|'da'|'de'|'el'|'en'|'es'|'et'|'fi'|'fr'|'ga'|'hu'|'it'|'lt'|'lv'|'mt'|'nl'|'pl'|'pt'|'ro'|'sk'|'sl'|'sv'|'tr'}

**Documentation** The list of official languages of the EU.

#### Schema Component Representation

```
<xs:simpleType name="EULanguageCode_Type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="bg"/>
    <xs:enumeration value="cs"/>
    <xs:enumeration value="da"/>
    <xs:enumeration value="de"/>
    <xs:enumeration value="el"/>
    <xs:enumeration value="en"/>
    <xs:enumeration value="es"/>
    <xs:enumeration value="et"/>
    <xs:enumeration value="fi"/>
    <xs:enumeration value="fr"/>
```

```
<xs:enumeration value="ga"/>
<xs:enumeration value="hu"/>
<xs:enumeration value="it"/>
<xs:enumeration value="lt"/>
<xs:enumeration value="lv"/>
<xs:enumeration value="mt"/>
<xs:enumeration value="nl"/>
<xs:enumeration value="pl"/>
<xs:enumeration value="pt"/>
<xs:enumeration value="ro"/>
<xs:enumeration value="sk"/>
<xs:enumeration value="sl"/>
<xs:enumeration value="sv"/>
<xs:enumeration value="tr"/>
</xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **EULanguageList\_Type**

<i>Super-types:</i>	None
<i>Sub-types:</i>	None

<b>Name</b>	EULanguageList_Type
<b>Content</b>	<ul style="list-style-type: none"><li>List of: <a href="#">EULanguageCode_Type</a></li></ul>
<b>Documentation</b>	List of EU Languages

### Schema Component Representation

```
<xs:simpleType name="EULanguageList_Type">
  <xs:list itemType="EULanguageCode_Type"/>
</xs:simpleType>
```

[top](#)

## Simple Type: **Gender\_Type**

<i>Super-types:</i>	<a href="#">xs:token</a> < <b>Gender_Type</b> (by restriction)
<i>Sub-types:</i>	None

<b>Name</b>	Gender_Type
<b>Content</b>	<ul style="list-style-type: none"><li>Base XSD Type: token</li><li><i>value</i> comes from list: {'M' 'F'}</li></ul>
<b>Documentation</b>	This element can be used to indicate Gender for individuals.

### Schema Component Representation

```
<xs:simpleType name="Gender_Type">
  <xs:restriction base="xs:token">
    <xs:enumeration value="M"/>
    <xs:enumeration value="F"/>
  </xs:restriction>
</xs:simpleType>
```

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

[top](#)

## Simple Type: **IBAN\_Type**

*Super-types:* [xs:string](#) < **IBAN\_Type** (by restriction)

*Sub-types:* None

**Name** IBAN\_Type

**Content**

- Base XSD Type: string
- *pattern* = [A-Z]{2}[0-9]{2}[0-9,A-Z]{10,30}

**Documentation**

The International Bank Account Number has to be given here for the account into which the payment in question has been made. Depending on the transmission type this element is optional. Its structure is: Country code, 2 letters/Check digits, 2 digits/Basic Bank Account Number (BBAN), 10 to 30 alphanumeric characters

### Schema Component Representation

```
<xs:simpleType name="IBAN_Type">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Z]{2}[0-9]{2}[0-9,A-Z]{10,30}" />
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **MsgId\_Type**

*Super-types:* [xs:string](#) < **MsgId\_Type** (by restriction)

*Sub-types:* None

**Name** MsgId\_Type

**Content**

- Base XSD Type: string
- *pattern* = [a-zA-Z0-9\-\\_:@\.]\*
- *length* <= 64

**Documentation**

Message Identifier

### Schema Component Representation

```
<xs:simpleType name="MsgId_Type">
  <xs:restriction base="xs:string">
    <xs:maxLength value="64" />
    <xs:pattern value="[a-zA-Z0-9\-\_:@\.]*" />
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **NameFree\_Type**

*Super-types:* [xs:string](#) < **NameFree\_Type** (by restriction)

*Sub-types:* None

**Name** NameFree\_Type

**Content**

- Base XSD Type: string

**Documentation** An unstructured name

### Schema Component Representation

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

[top](#)

## Simple Type: **nameType\_Type**

*Super-types:* [xs:string](#) < **nameType\_Type** (by restriction)

*Sub-types:* None

**Name** nameType\_Type

**Content**

- Base XSD Type: string
- *value* comes from list:  
{'indiv'|'alias'|'nick'|'aka'|'dba'|'legal'|'atbirth'}

### Schema Component Representation

```
<xs:simpleType name="nameType_Type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="indiv"/>
    <xs:enumeration value="alias"/>
    <xs:enumeration value="nick"/>
    <xs:enumeration value="aka"/>
    <xs:enumeration value="dba"/>
    <xs:enumeration value="legal"/>
    <xs:enumeration value="atbirth"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **NumberLength3\_Type**

*Super-types:* [xs:integer](#) < **NumberLength3\_Type** (by restriction)

*Sub-types:* None

Name	NumberLength3_Type
Content	<ul style="list-style-type: none"><li>• Base XSD Type: integer</li><li>• <math>0 \leq \text{value} \leq 999</math></li></ul>
Documentation	A number with 3 digits

#### Schema Component Representation

```
<xs:simpleType name="NumberLength3_Type">
  <xs:restriction base="xs:integer">
    <xs:maxInclusive value="999"/>
    <xs:minInclusive value="0"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)

---

### Simple Type: **PhoneNumber\_Type**

Super-types:	<a href="#">xs:string</a> < <b>PhoneNumber_Type</b> (by restriction)
Sub-types:	None

Name	PhoneNumber_Type
Content	<ul style="list-style-type: none"><li>• Base XSD Type: string</li><li>• <math>\text{pattern} = (\backslash+)?[0-9]\{1,20\}</math></li></ul>
Documentation	A telephone number

#### Schema Component Representation

```
<xs:simpleType name="PhoneNumber_Type">
  <xs:restriction base="xs:string">
    <xs:pattern value="(\backslash+)?[0-9]\{1,20\}" />
  </xs:restriction>
</xs:simpleType>
```

[top](#)

---

### Simple Type: **Quarter\_Type**

Super-types:	<a href="#">xs:integer</a> < <b>Quarter_Type</b> (by restriction)
Sub-types:	None

Name	Quarter_Type
Content	<ul style="list-style-type: none"><li>• Base XSD Type: integer</li><li>• <math>1 \leq \text{value} \leq 4</math></li></ul>
Documentation	A quarter (Q1, Q2, Q3, Q4)

#### Schema Component Representation



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

[top](#)

## Simple Type: **VATNumber\_Type**

*Super-types:* [xs:string](#) < **VATNumber\_Type** (by restriction)

*Sub-types:* None

**Name** VATNumber\_Type

**Content**

- Base XSD Type: string
- *pattern* = [A-Za-z0-9\+\\*]{1,12}

### Schema Component Representation

```
<xs:simpleType name="VATNumber_Type">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Za-z0-9\+\*]{1,12}" />
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **VATNumberOrTIN\_Type**

*Super-types:* [xs:string](#) < **VATNumberOrTIN\_Type** (by restriction)

*Sub-types:* None

**Name** VATNumberOrTIN\_Type

**Content**

- Base XSD Type: string
- *length* <= 20

**Documentation** A tax identifier, VAT or other

### Schema Component Representation

```
<xs:simpleType name="VATNumberOrTIN_Type">
  <xs:restriction base="xs:string">
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)

## Simple Type: **VatRate\_Type**

**Super-types:** [xs:decimal](#) < **VatRate\_Type** (by restriction)

**Sub-types:** None

**Name** VatRate\_Type

**Content**

- Base XSD Type: decimal
- $0.00 \leq \text{value} \leq 100.00$
- *no. of fraction digits* = 2

**Documentation** VAT rate

### Schema Component Representation

```
<xs:simpleType name="VatRate_Type">
  <xs:restriction base="xs:decimal">
    <xs:minInclusive value="0.00"/>
    <xs:maxInclusive value="100.00"/>
    <xs:fractionDigits value="2"/>
  </xs:restriction>
</xs:simpleType>
```

[top](#)

---

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