



HANDBOOK/IMPLEMENTATION GUIDELINE FOR XML INTRASTAT DECLARATION FILE



**NATIONAL INSTITUTE OF STATISTICS
ROMANIA**

National Institute of Statistics
16 Blvd Libertatii, sector 5, Bucharest
Phone: 318.18.58; 317.77.20;
317.77.21; 317.77.22;
317.77.23

E-mail: intrastat@insse.ro
<http://www.intrastat.ro>

© INS 2007

Reproducing the content of this publication, completely or partly, in original or modified, as well as its storage in a retrieval system, or transmitted, in any form and by any means are forbidden without the written permission of the National Institute of Statistics.

Using the content of this publication with explanatory or justifying title, in articles, studies, books is allowed only clearly and precisely indicating the source.

Table of Contents

PREFACE	4
ABBREVIATIONS.....	4
1. XML SCHEMA DEFINITION LANGUAGE OF DECLARATION: INTRASTAT.XSD.....	5
2. XML DIAGRAM	16
3. XML SCHEMA DOCUMENTATION	23
LEGEND.....	71

Preface

Creation and submission of Intrastat declarations is possible through the following ways:

- Using the Intrastat offline application software. The software is available, free of charge, from INS or it can be downloaded from the website www.intrastat.ro.
- Using the online Intrastat application. This service is available from on the website www.intrastat.ro.
- Producing the electronic declaration files by modifying appropriately existing software (e.g. ERP) used by companies.

The present handbook deals the above mentioned third way of creating and submitting declarations and **its purpose is to provide a Guide for Message Implementation Guideline of the Intrastat declaration in XML.**

Such information is valuable to the **IT departments of Providers of Statistical Information (PSIs)** who would like to use their own software to produce the Intrastat declarations in XML. This of course means that such software should be modified appropriately in order to be able to produce the declaration in the required file format.

The specifications of XML take into account the following standards:

- Extensible Markup Language (XML) 1.0 (Second edition), Recommendations W3C, 6 October 2000 (<http://www.w3.org/XML>);
- XML Schema Part 1: Structures, Recommendations W3C, 2 May 2001 (<http://www.w3.org/XML/Schema>);
- XML Schema Part 2: Datatypes, Recommendations W3C, 2 May 2001 (<http://www.w3.org/XML/Schema>).

Abbreviations

For the purposes of the present document, the following symbols apply:

INS National Institute of Statistics of Romania

PSI Provider of Statistical Information - Party responsible for providing statistical information in the Intrastat system

TDP Third Declaring Party – Third Declaring Party submitting the Intrastat declarations on behalf of a PSI

XML eXtended Markup Language

Note: The information contained in a xml declaration file is produced by a company's own software in not encrypted. To encrypt the information contained in the xml declaration file use the Intrastat offline application software. Using the Intrastat offline application software requires only importing the xml file and then save it. When the xml file is saved the information contained will be encrypted. This procedure also validates the contents of the xml file.

1. XML schema definition language of Declaration: intrastat.xsd

The XML schema definition language for a Declaration is described below.

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

```
<!--
```

```
Document : intrastat4.xsd
```

```
Created on : July 17, 2014, 5:14 PM
```

```
Author : intrarom
```

```
Description:
```

```
Purpose of XML Schema document follows.
```

```
-->
```

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.intrastat.ro/xml/InsSchema"
  xmlns="http://www.intrastat.ro/xml/InsSchema"
  elementFormDefault="qualified">
```

```
<!--ROOT ELEMENTS -->
```

```
<!--The Nill Arrival Declaration Root Element -->
```

```
<xsd:element name="InsNillArrival" type="InsNillArrivalType">
```

```
<xsd:annotation>
```

```
<xsd:documentation>
```

```
Root element for the nill declaration for arrivals (imports).
```

```
</xsd:documentation>
```

```
</xsd:annotation>
```

```
</xsd:element>
```

```
<!--The Nill Dispatch Declaration Root Element -->
```

```
<xsd:element name="InsNillDispatch" type="InsNillDispatchType">
```

```
<xsd:annotation>
```

```
<xsd:documentation>
```

```
Root element for the nill declaration for dispatches (exports).
```

```
</xsd:documentation>
```

```
</xsd:annotation>
```

```
</xsd:element>
```

```

<!--The New Arrival Declaration Root Element -->
<xsd:element name="InsNewArrival" type="InsNewArrivalType">
  <xsd:annotation>
    <xsd:documentation>
      Root element for a declaration of arrivals (imports).
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<!--The New Dispatch Declaration Root Element -->
<xsd:element name="InsNewDispatch" type="InsNewDispatchType">
  <xsd:annotation>
    <xsd:documentation>
      Root element for a declaration of dispatches (exports).
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<!--The Revised Arrival Declaration Root Element -->
<xsd:element name="InsRevisedArrival" type="InsRevisedArrivalType">
  <xsd:annotation>
    <xsd:documentation>
      Root element for a revised declaration of arrivals (imports).
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<!--The Revised Dispatch Declaration Root Element -->
<xsd:element name="InsRevisedDispatch" type="InsRevisedDispatchType">
  <xsd:annotation>
    <xsd:documentation>
      Root element for a revised declaration of dispatches (exports).
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<!-- SIMPLE TYPES -->

<!--Positive longs -->
<xsd:simpleType name="PositiveLongType">
  <xsd:restriction base="xsd:long">
    <xsd:minExclusive value="0"/>
  </xsd:restriction>
</xsd:simpleType>

<!--Positive ints-->
<xsd:simpleType name="PositiveIntType">
  <xsd:restriction base="xsd:int">

```

```

    <xsd:minExclusive value="0"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<!--VAT Number-->
<xsd:simpleType name="VatNumberType" >
  <xsd:annotation>
    <xsd:documentation>
      The 10-digit string corresponding to the VAT number of the firm
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-9]{10}"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<!--CN8 Code -->
<xsd:simpleType name="Cn8CodificationType">
  <xsd:annotation>
    <xsd:documentation>
      The 8-digit CN8 commodity/item code. See the corresponding CN8
      nomenclature.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-9]{8}"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<!--Country Code -->
<xsd:simpleType name="CountryType">
  <xsd:annotation>
    <xsd:documentation>
      The code number for the country. See the corresponding country
      nomenclature.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:token">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="2"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<!-- COMPLEX TYPES -->

```

```

<!--The codifications version Information Type-->
<xsd:complexType name="InsCodeVersionsType">

```

```
<xsd:annotation>
  <xsd:documentation>
```

Information about the nomenclatures used in the declaration and their version.

```
    </xsd:documentation>
  </xsd:annotation>
</xsd:sequence>
  <xsd:element name="CountryVer" type="xsd:token" />
  <xsd:element name="EuCountryVer" type="xsd:token" />
  <xsd:element name="CnVer" type="xsd:token" />
  <xsd:element name="ModeOfTransportVer" type="xsd:token" />
  <xsd:element name="DeliveryTermsVer" type="xsd:token" />
  <xsd:element name="NatureOfTransactionAVer" type="xsd:token" />
  <xsd:element name="NatureOfTransactionBVer" type="xsd:token" />
  <xsd:element name="CountyVer" type="xsd:token"/>
  <xsd:element name="LocalityVer" type="xsd:token"/>
  <xsd:element name="UnitVer" type="xsd:token"/>
</xsd:sequence>
</xsd:complexType>
```

```
<!--The Contact person information Type -->
<xsd:complexType name="ContactPersonType">
  <xsd:annotation>
    <xsd:documentation>
```

Information about the contact person responsible for filling up the declaration.

```
    </xsd:documentation>
  </xsd:annotation>
</xsd:sequence>
  <xsd:element name="LastName" type="xsd:token"/>
  <xsd:element name="FirstName" type="xsd:token"/>
  <xsd:element name="Email" minOccurs="0" type="xsd:token"/>
  <xsd:element name="Phone" type="xsd:token"/>
  <xsd:element name="Fax" minOccurs="0" type="xsd:token"/>
  <xsd:element name="Position" minOccurs="0" type="xsd:token"/>
</xsd:sequence>
</xsd:complexType>
```

```
<!--The Address Information Type-->
<xsd:complexType name="AddressType">
  <xsd:annotation>
    <xsd:documentation>
```

Information about the address. LocalityCode and CountyCode are strings that take as values the corresponding codes from the related nomenclatures.

```
    </xsd:documentation>
  </xsd:annotation>
</xsd:sequence>
```

```

<xsd:element name="Street" type="xsd:token" />
<xsd:element name="StreetNumber" type="xsd:token" minOccurs="0"/>
<xsd:element name="Block" type="xsd:token" minOccurs="0"/>
<xsd:element name="Stairs" type="xsd:token" minOccurs="0"/>
<xsd:element name="Apartment" type="xsd:token" minOccurs="0"/>
<xsd:element name="LocalityCode" type="xsd:token" />
<xsd:element name="CountyCode" type="xsd:token"/>
<xsd:element name="PostalCode" type="xsd:token" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>

```

<!--The Third Declaring Party Information Type-->

```

<xsd:complexType name="DTPTYPE">
  <xsd:annotation>
    <xsd:documentation>
      Identification info for a Third Party Declarant (DTP).
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="VatNr" type="VatNumberType"/>
    <xsd:element name="FirmName" type="xsd:token"/>
    <xsd:element name="DTPAddress" type="AddressType"/>
  </xsd:sequence>
</xsd:complexType>

```

<!--The Declaration Header Type -->

```

<xsd:complexType name="InsDeclarationHeaderType" >
  <xsd:annotation>
    <xsd:documentation>
      Information that makes up the declaration header:
      - VAT number:
      - Name of the firm
      - Reference period
      - Date of creation
      - Application Reference (this is not to be completed by the declarant)
      - DTP details
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="VatNr" type="VatNumberType" />
    <xsd:element name="FirmName" type="xsd:token"/>
    <xsd:element name="RefPeriod" type="xsd:gYearMonth" />
    <xsd:element name="CreateDt" type="xsd:dateTime" />
    <xsd:element name="ApplicationRef" type="xsd:token" minOccurs="0"/>
    <xsd:element name="ContactPerson" type="ContactPersonType"/>
    <xsd:element name="DTPDetails" type="DTPTYPE"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

```

<!--Supplementary Unit Details Type-->
<xsd:complexType name="InsSupplUnitsInfoType">
  <xsd:sequence>
    <xsd:element name="SupplUnitCode" type="xsd:token">
      <xsd:annotation>
        <xsd:documentation>
          The Supplementary Units code taken from the related nomenclature.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="QtyInSupplUnits" type="PositiveLongType" />
  </xsd:sequence>
</xsd:complexType>

```

```

<!--The Abstract Declaration Item Type-->
<xsd:complexType name="InsDeclarationItemType" abstract="true">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        Information that makes up a declaration item:
        - CN8 commodity/item code from the related nomenclature
        - Invoice Value
        - Statistical Value
        - Net Mass (in Kg)
        - Nature of Transaction code from the related nomenclature
        - Terms of Delivery code from the related nomenclature
        - Mode of Transport code from the related nomenclature
        - Supplementary Units information
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="Cn8Code" type="Cn8CodificationType" />
    <xsd:element name="InvoiceValue" type="PositiveLongType"

```

> <xsd:annotation>

 <xsd:documentation>

 Invoiced amount is the value of the commodity indicated on the invoice, which might contain transport and insurance costs according to the delivery terms but not taxes or levies. Should be given in RON, without decimals. For further information see Handbook for Data Providers chapter 6.3.5 Invoice value stated in RON.

```

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="StatisticalValue" minOccurs="0"
type="PositiveLongType" >

```

```

          <xsd:annotation>
            <xsd:documentation>
              The statistical value is the value of a product at the time of border crossing. Should be given in RON, without decimals. For further information see

```

Handbook for Data Providers chapter 6.3.6 Statistical value.

```

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="NetMass" type="PositiveLongType" >
  <xsd:annotation>
    <xsd:documentation>

```

The net weight is the weight in kilograms without packaging of any kind. The net weight is entered without decimals. Product items weighing less than 1 kg are entered with the figure 1. For certain CN product numbers, net weight in kg does not have to be stated. For certain products there are also supplementary units that must be declared. For further information see Handbook for Data Providers chapter 6.3.3 Net mass in whole kg.

```

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="NatureOfTransactionACode" type="xsd:token" >
  <xsd:annotation>
    <xsd:documentation>

```

The Nature of Transaction (A) code taken from the related nomenclature.

```

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="NatureOfTransactionBCode" minOccurs="0"
type="xsd:token" >
  <xsd:annotation>
    <xsd:documentation>

```

The Nature of Transaction (B) code taken from the related nomenclature.

```

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="DeliveryTermsCode" type="xsd:token" >
  <xsd:annotation>
    <xsd:documentation>

```

The Terms of Delivery code taken from the related nomenclature.

```

    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:element name="ModeOfTransportCode" type="xsd:token" >
  <xsd:annotation>
    <xsd:documentation>
    The Mode of Transport code taken from the related nomenclature.
  </xsd:documentation>
  </xsd:annotation>
</xsd:element>

```

```

    <xsd:element name="CountryOfOrigin" type="CountryType" />
    <xsd:element name="InsSupplUnitsInfo" type="InsSupplUnitsInfoType"
minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="OrderNr" type="PositiveIntType"/>
</xsd:complexType>

<!--The Arrival Declaration Item Type -->
<xsd:complexType name="InsArrivalItemType">
  <xsd:annotation>
    <xsd:documentation>
      The declaration item of a declaration for arrivals.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationItemType">
      <xsd:sequence>
        <xsd:element name="CountryOfConsignment" type="CountryType"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!--The Dispatch Declaration Item Type -->
<xsd:complexType name="InsDispatchItemType">
  <xsd:annotation>
    <xsd:documentation>
      The declaration item of a declaration for dispatches.
      - Partner CUI_Number
      - Partner CUI Country
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationItemType">
      <xsd:sequence>
        <xsd:element name="CountryOfDestination" type="CountryType"/>
        <xsd:element name="PartnerCountryCode" type="CountryType">
          </xsd:element>
        <xsd:element name="PartnerVatNr"
type="xsd:string"></xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

<!--The Abstract Declaration Type -->
<xsd:complexType name="InsDeclarationType" abstract="true" >
  <xsd:annotation>
    <xsd:documentation>
      The abstract definition of a declaration.
      Attribute "SchemaVersion" is a string constant and must be set always to
"1.0".
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="InsCodeVersions" type="InsCodeVersionsType"/>
    <xsd:element name="InsDeclarationHeader"
type="InsDeclarationHeaderType"/>
  </xsd:sequence>
  <xsd:attribute name="SchemaVersion" type="xsd:string" use="required"
fixed="1.0"/>
</xsd:complexType>

<!--The Nill Arrival Declaration Type -->
<xsd:complexType name="InsNillArrivalType">
  <xsd:annotation>
    <xsd:documentation>
      The nill declaration for arrivals
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType"/>
  </xsd:complexContent>
</xsd:complexType>

<!--The Nill Dispatch Declaration Type -->
<xsd:complexType name="InsNillDispatchType">
  <xsd:annotation>
    <xsd:documentation>
      The nill declaration for dispatches
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType"/>
  </xsd:complexContent>
</xsd:complexType>

<!--The New Arrival Declaration Type -->
<xsd:complexType name="InsNewArrivalType">
  <xsd:annotation>
    <xsd:documentation>
      The declaration for arrivals

```

```

    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType">
      <xsd:sequence>
        <xsd:element name="InsArrivalItem" type="InsArrivalItemType"
          minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!--The Revised Arrival Declaration Type -->
<xsd:complexType name="InsRevisedArrivalType">
  <xsd:annotation>
    <xsd:documentation>
      The revised declaration for arrivals
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType">
      <xsd:sequence>
        <xsd:element name="InsArrivalItem" type="InsArrivalItemType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!--The New Dispatch Declaration Type -->
<xsd:complexType name="InsNewDispatchType">
  <xsd:annotation>
    <xsd:documentation>
      The declaration for dispatches
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType">
      <xsd:sequence>
        <xsd:element name="InsDispatchItem" type="InsDispatchItemType"
          minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

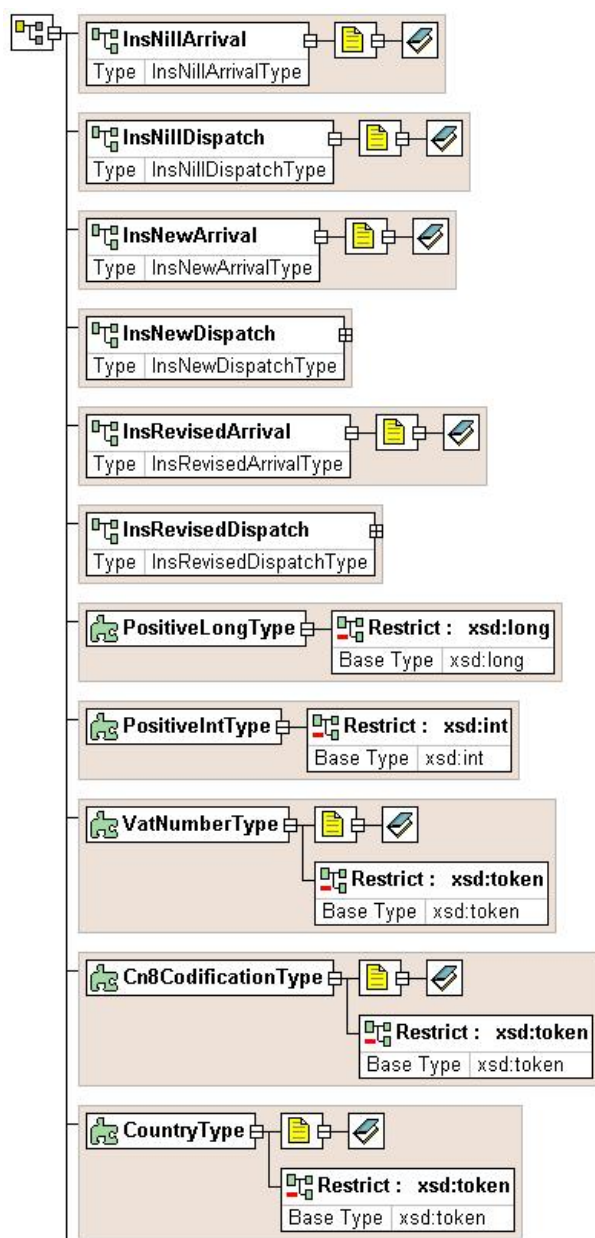
<!--The Revised Dispatch Declaration Type -->
<xsd:complexType name="InsRevisedDispatchType">

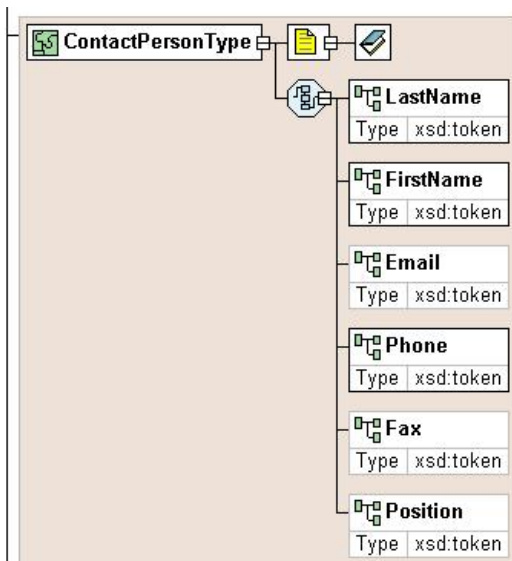
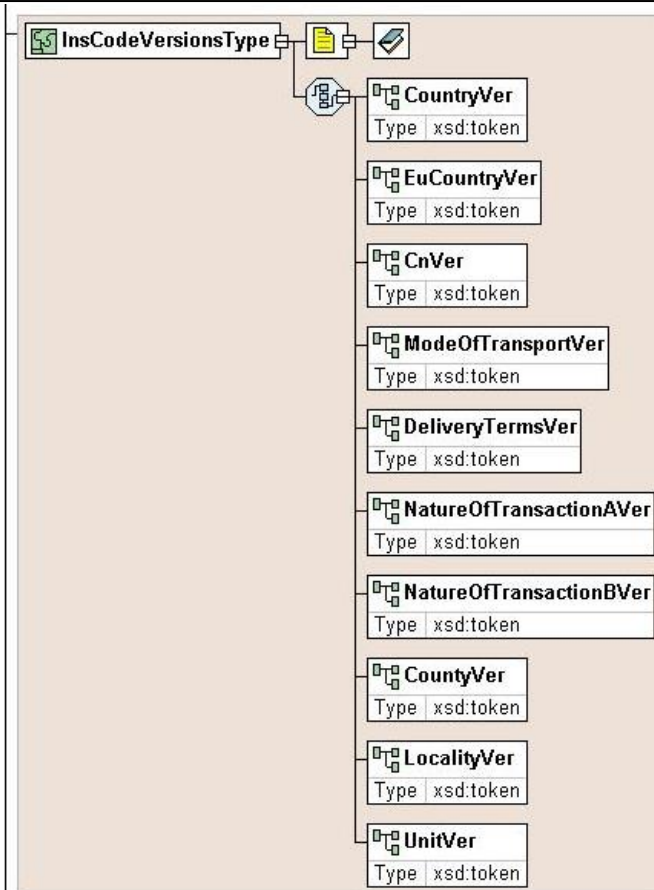
```

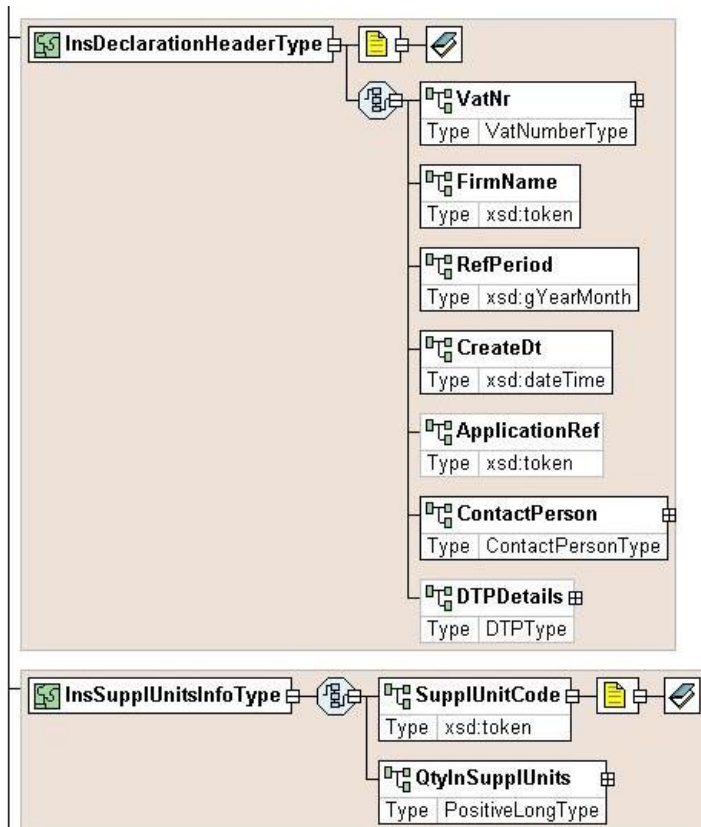
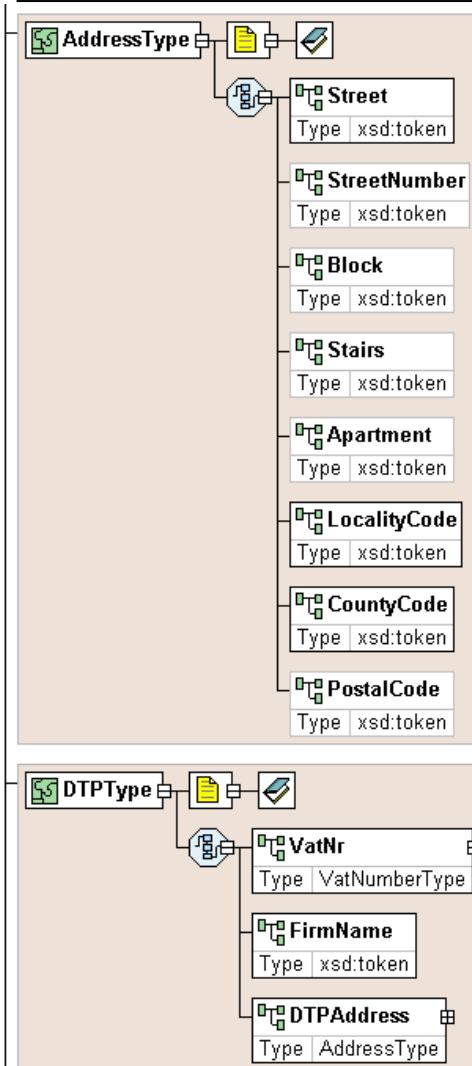
```
<xsd:annotation>
  <xsd:documentation>
    The revised declaration for dispatches
  </xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
  <xsd:extension base="InsDeclarationType">
    <xsd:sequence>
      <xsd:element name="InsDispatchItem" type="InsDispatchItemType"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

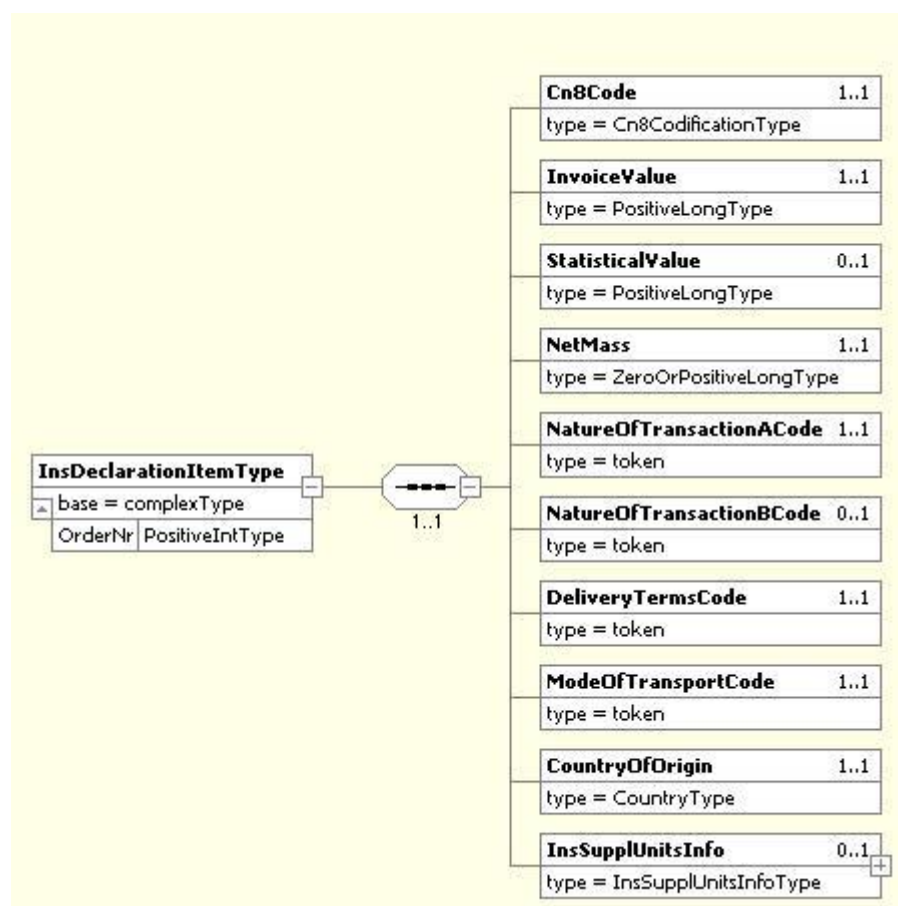
</xsd:schema>
```

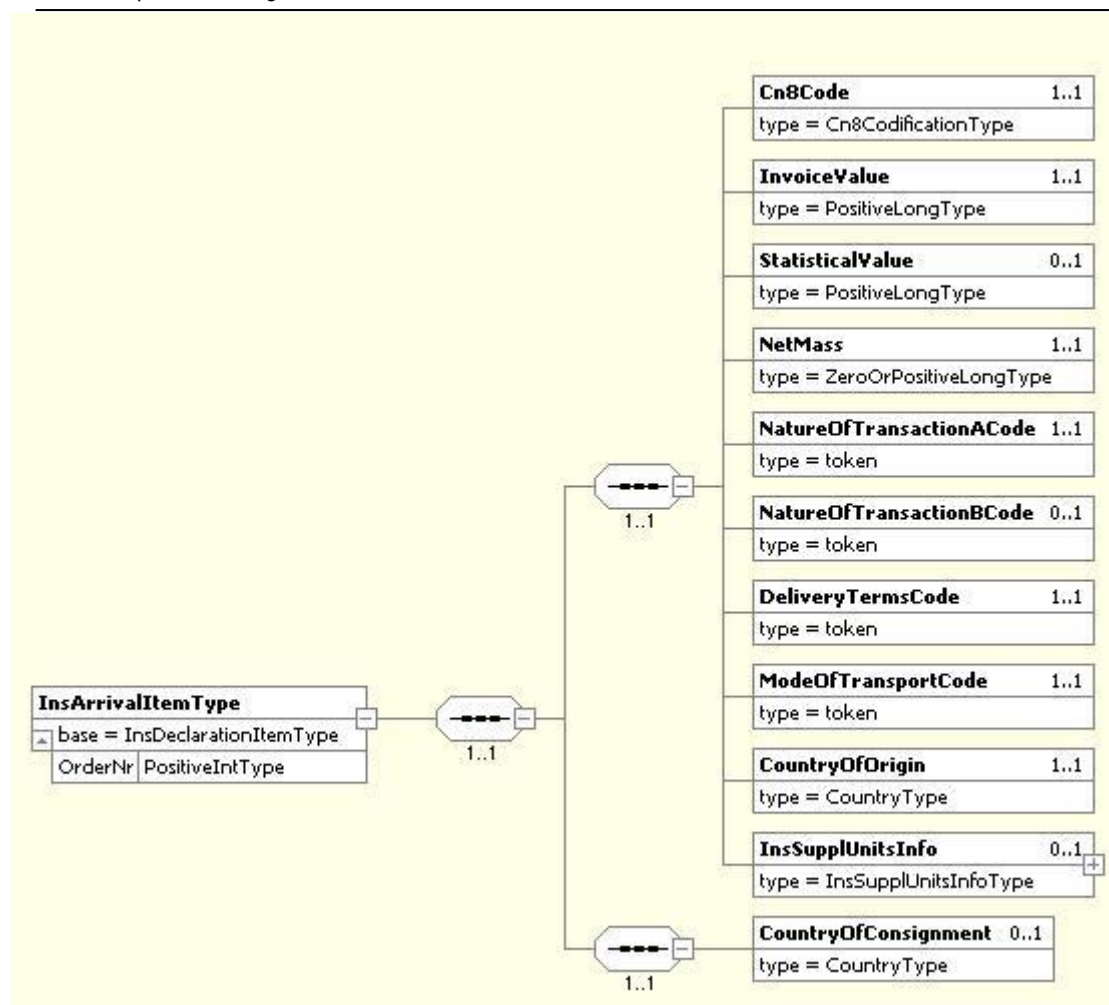
2. XML Diagram

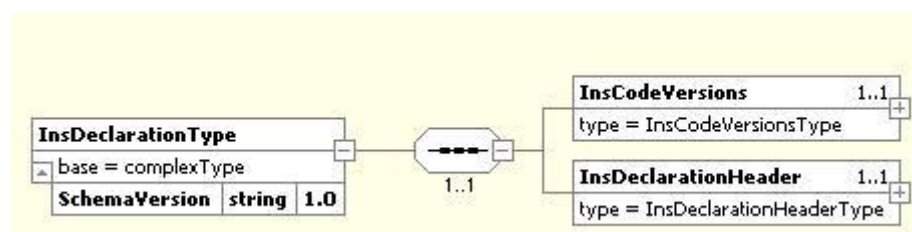
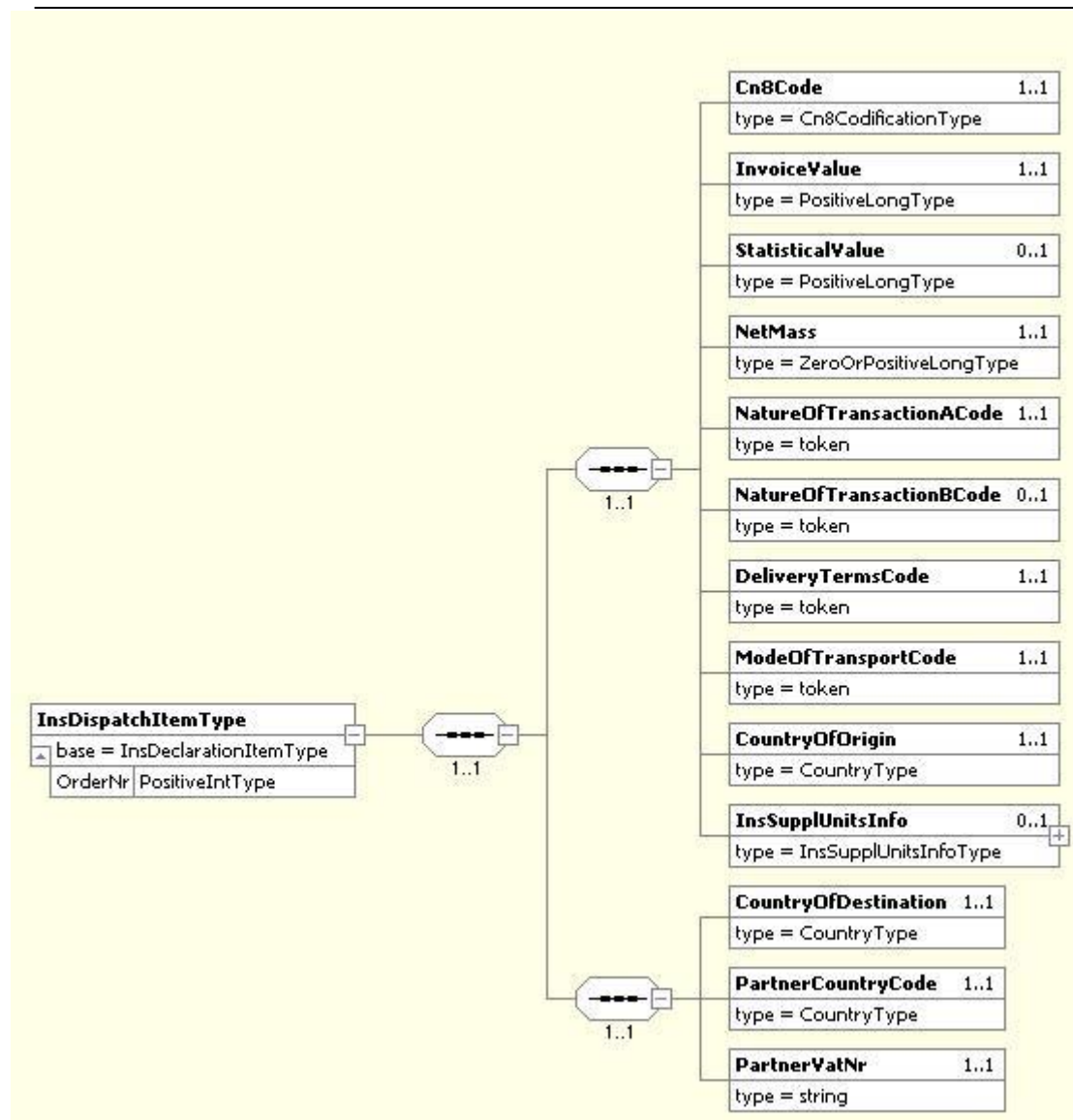


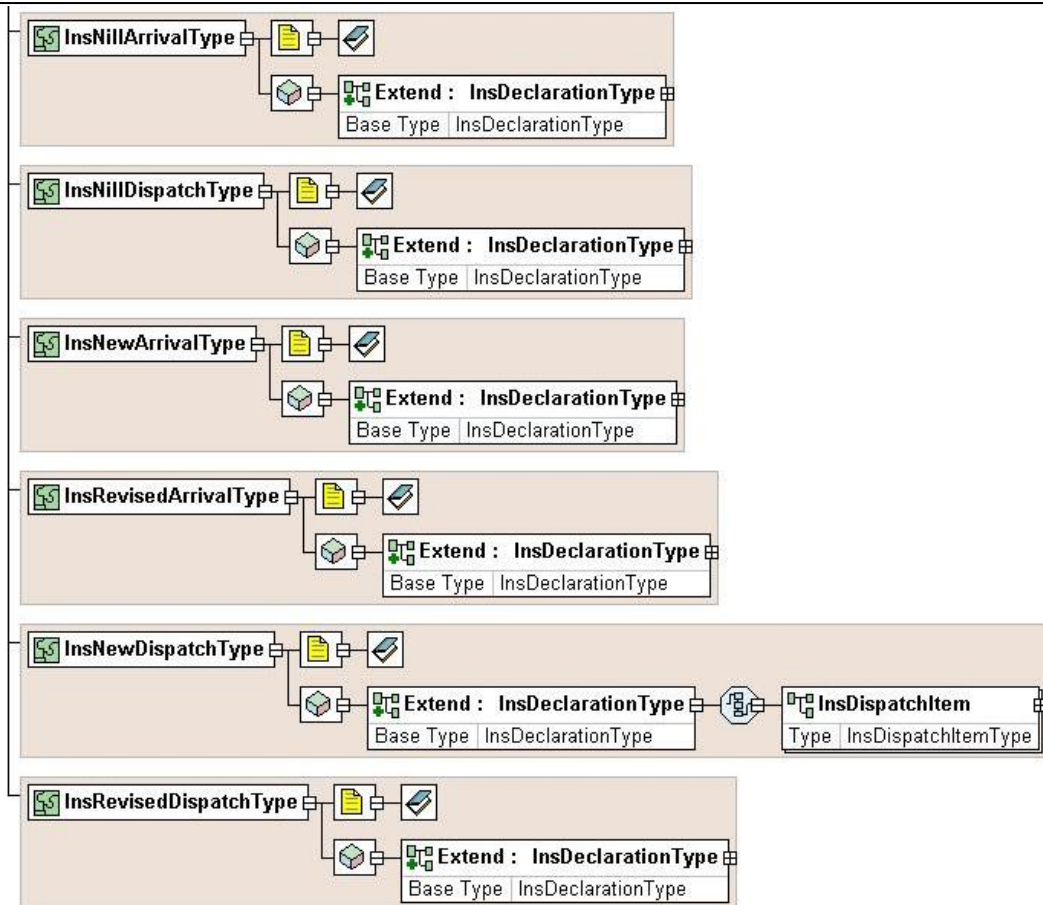












3. XML Schema Documentation

Schema Document Properties

<u>Target Namespace</u>	
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.

Declared Namespaces

Prefix	Namespace
Default namespace	
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema

Schema Component Representation

```
<xsd:schema targetNamespace="http://www.intrastat.ro/xml/InsSchema" elementFormDefault="qualified">
```

...

```
</xsd:schema>
```

Global Declarations

Element: **InsNewArrival**

Name	InsNewArrival
Type	InsNewArrivalType
<u>Nilable</u>	no
<u>Abstract</u>	no
Documentation	Root element for a declaration of arrivals (imports).
Diagram	

XML Instance Representation

```
<InsNewArrival
  SchemaVersion="1.0 [1]">
  <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
```

```
<InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
```

```
<InsArrivalItem> InsArrivalItemType </InsArrivalItem> [1..*]
```

```
</InsNewArrival>
```

Schema Component Representation

```
<xsd:element name="InsNewArrival" type="InsNewArrivalType" />
```

Element: **InsNewDispatch**

Name	InsNewDispatch
Type	InsNewDispatchType
<u>Nilable</u>	no
<u>Abstract</u>	no
Documentation	Root element for a declaration of dispatches (exports).
Diagram	 <pre> classDiagram class InsNewDispatch { } class InsNewDispatchType { } InsNewDispatch -- InsNewDispatchType : Type </pre>

XML Instance Representation

```

<InsNewDispatch
  SchemaVersion="1.0 [1]">
  <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
  <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
  <InsDispatchItem> InsDispatchItemType </InsDispatchItem> [1..*]
</InsNewDispatch>

```

Schema Component Representation

```
<xsd:element name="InsNewDispatch" type="InsNewDispatchType" />
```

Element: **InsNillArrival**

Name	InsNillArrival
Type	InsNillArrivalType
<u>Nillable</u>	no
<u>Abstract</u>	no
Documentation	Root element for the nill declaration for arrivals (imports).

Diagram**XML Instance Representation**

```

<InsNillArrival
  SchemaVersion="1.0 [1]">
  <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
  <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
</InsNillArrival>
  
```

Schema Component Representation

```

<xsd:element name="InsNillArrival" type="InsNillArrivalType" />
  
```

Element: [InsNillDispatch](#)

Name	InsNillDispatch
Type	InsNillDispatchType
<u>Nilable</u>	no
<u>Abstract</u>	no

Documentation	Root element for the nill declaration for dispatches (exports).
Diagram	

XML Instance Representation

```

<InsNillDispatch
  SchemaVersion="1.0 [1]">
  <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
  <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
</InsNillDispatch>

```

Schema Component Representation

```
<xsd:element name="InsNillDispatch" type="InsNillDispatchType" />
```

Element: **InsRevisedArrival**

Name	InsRevisedArrival
Type	InsRevisedArrivalType
<u>Nillable</u>	no

<u>Abstract</u>	no
Documentation	Root element for a revised declaration of arrivals (imports).
Diagram	 <pre> classDiagram class InsRevisedArrival { Type InsRevisedArrivalType } InsRevisedArrival --> InsRevisedArrivalType </pre>

XML Instance Representation

```

<InsRevisedArrival
  SchemaVersion="1.0 [1]">
  <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
  <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
  <InsArrivalItem> InsArrivalItemType </InsArrivalItem> [0..*]
</InsRevisedArrival>
  
```

Schema Component Representation

```
<xsd:element name="InsRevisedArrival" type="InsRevisedArrivalType"/>
```

Element: **InsRevisedDispatch**

Name	InsRevisedDispatch
-------------	--------------------

Type	InsRevisedDispatchType
<u>Nilable</u>	no
<u>Abstract</u>	no
Documentation	Root element for a revised declaration of dispatches (imports).
Diagram	

XML Instance Representation

```

<InsRevisedDispatch
  SchemaVersion="1.0 [1]">
  <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
  <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
  <InsDispatchItem> InsDispatchItemType </InsDispatchItem> [0..*]
</InsRevisedDispatch>

```

Schema Component Representation

```

<xsd:element name="InsRevisedDispatch" type="InsRevisedDispatchType" />

```

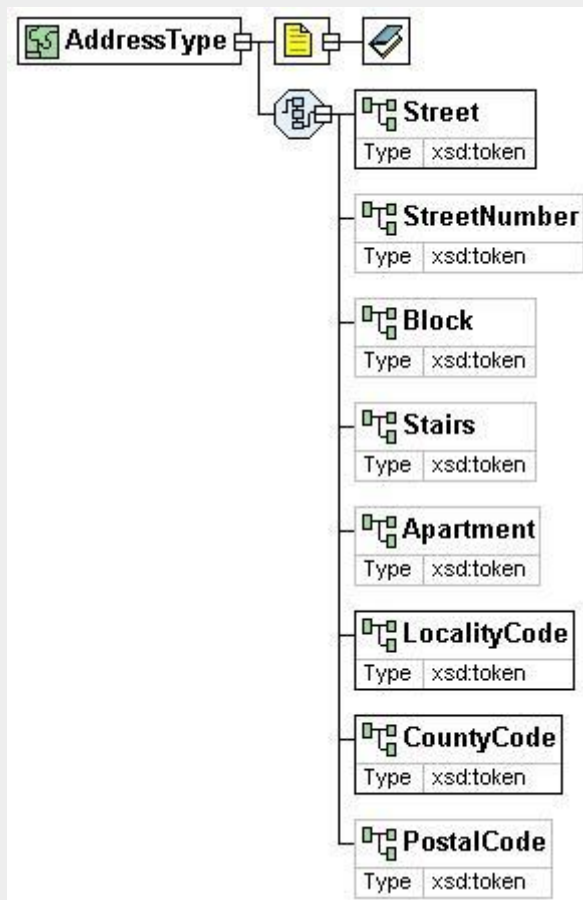
Global Definitions

Complex Type: **AddressType**

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

Name	AddressType
<u>Abstract</u>	no
Documentation	Information about the address. LocalityCode and CountyCode are stings that take as values the corresponding codes from the related nomenclatures.

Diagram



XML Instance Representation

<...>

<Street> [xsd:token](#) </Street> [1]

<StreetNumber> [xsd:token](#) </StreetNumber> [1]

<Block> [xsd:token](#) </Block> [0..1]

```
<Stairs> xsd:token </Stairs> [0..1]
<Apartment> xsd:token </Apartment> [0..1]
<LocalityCode> xsd:token </LocalityCode> [1]
<CountyCode> xsd:token </CountyCode> [1]
<PostalCode> xsd:token </PostalCode> [0..1]
</...>
```

Schema Component Representation

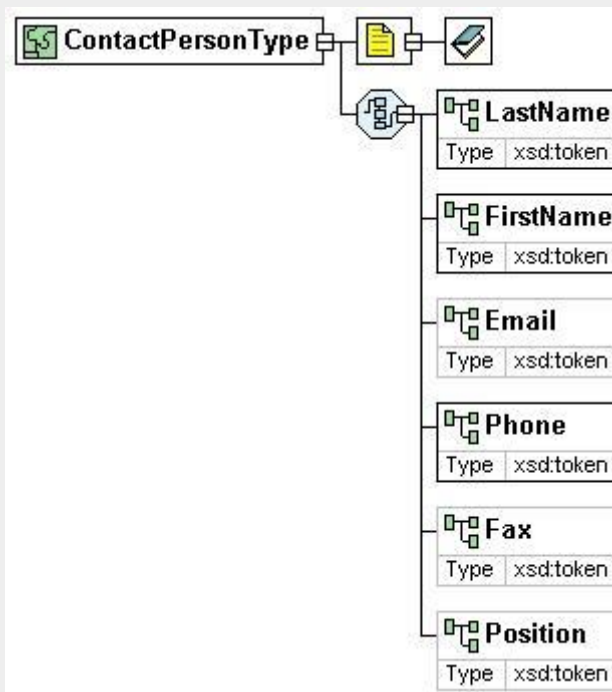
```
<xsd:complexType name="AddressType">
  <xsd:sequence>
    <xsd:element name="Street" type="xsd:token"/>
    <xsd:element name="StreetNumber" type="xsd:token" minOccurs="0"/>
    <xsd:element name="Block" type="xsd:token" minOccurs="0"/>
    <xsd:element name="Stairs" type="xsd:token" minOccurs="0"/>
    <xsd:element name="Apartment" type="xsd:token" minOccurs="0"/>
    <xsd:element name="LocalityCode" type="xsd:token"/>
    <xsd:element name="CountyCode" type="xsd:token"/>
    <xsd:element name="PostalCode" type="xsd:token" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Complex Type: **ContactPersonType**

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

Name	ContactPersonType
<u>Abstract</u>	no
Documentation	Information about the contact person responsible for filling up the declaration.

Diagram



XML Instance Representation

<...>

<LastName> [xsd:token](#) </LastName> [1]

<FirstName> [xsd:token](#) </FirstName> [1]

<Email> [xsd:token](#) </Email> [0..1]

<Phone> [xsd:token](#) </Phone> [1]

<Fax> [xsd:token](#) </Fax> [0..1]

<Position> [xsd:token](#) </Position> [0..1]

</...>

Schema Component Representation

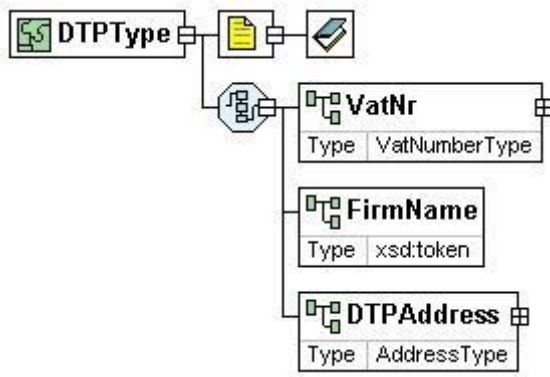
```
<xsd:complexType name="ContactPersonType">
  <xsd:sequence>
    <xsd:element name="LastName" type="xsd:token"/>
    <xsd:element name="FirstName" type="xsd:token"/>
    <xsd:element name="Email" type="xsd:token" minOccurs="0"/>
    <xsd:element name="Phone" type="xsd:token"/>
    <xsd:element name="Fax" type="xsd:token" minOccurs="0"/>
    <xsd:element name="Position" type="xsd:token" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Complex Type: **DTPTType**

Super-types: None

Sub-types: None

Name	DTPTType
-------------	----------

<u>Abstract</u>	no
Documentation	Identification info for a Third Party Declarant (DTP).
Diagram	

XML Instance Representation

```

<...>
  <VatNr> VatNumberType </VatNr> [1]
  <FirmName> xsd:token </FirmName> [1]
  <DTPAddress> AddressType </DTPAddress> [1]
</...>

```

Schema Component Representation

```

<xsd:complexType name="DTPType">
  <xsd:sequence>

```

```

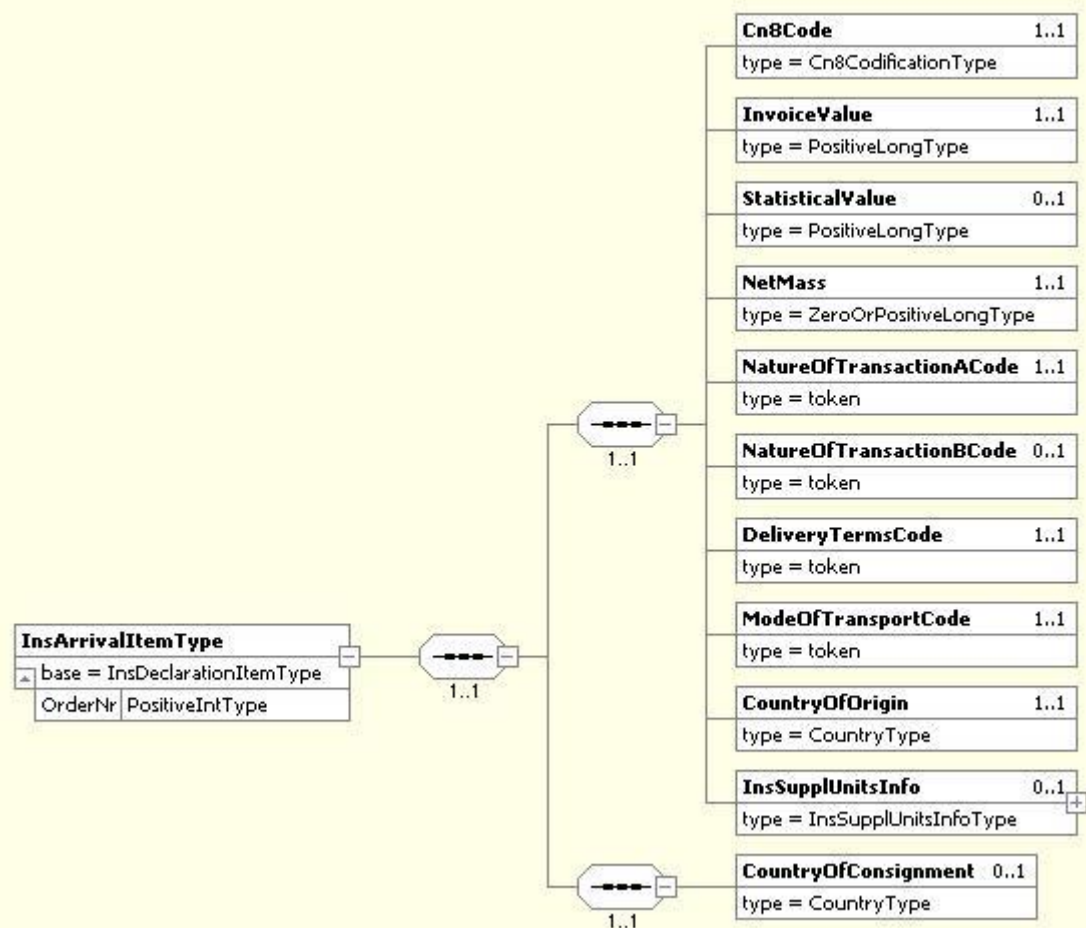
    <xsd:element name="VatNr" type="VatNumberType"/>
    <xsd:element name="FirmName" type="xsd:token"/>
    <xsd:element name="DTPAddress" type="AddressType"/>
  </xsd:sequence>
</xsd:complexType>

```

Complex Type: **InsArrivalItemType**

<i>Super-types:</i>	InsDeclarationItemType < InsArrivalItemType (by extension)
<i>Sub-types:</i>	None

Name	InsArrivalItemType
<u>Abstract</u>	no
Documentation	The declaration item of a declaration for arrivals.
Diagram	



XML Instance Representation

```

<...
  OrderNr="PositiveIntType [0..1]">
    <Cn8Code> Cn8CodificationType </Cn8Code> [1]
  
```

```
<InvoiceValue> PositiveLongType </InvoiceValue> [1]
<StatisticalValue> PositiveLongType </StatisticalValue> [0..1]
<NetMass> PositiveLongType </NetMass> [1]
<NatureOfTransactionACode> xsd:token </NatureOfTransactionACode> [1]
<NatureOfTransactionBCode> xsd:token </NatureOfTransactionBCode> [0..1]
<DeliveryTermsCode> xsd:token </DeliveryTermsCode> [1]
<ModeOfTransportCode> xsd:token </ModeOfTransportCode> [1]
<InsSupplUnitsInfo> InsSupplUnitsInfoType </InsSupplUnitsInfo> [0..1]
<CountryOfOrigin> CountryType </CountryOfOrigin> [1]
<CountryOfConsignment> CountryType </CountryOfConsignment> [0..1]
</...>
```

Schema Component Representation

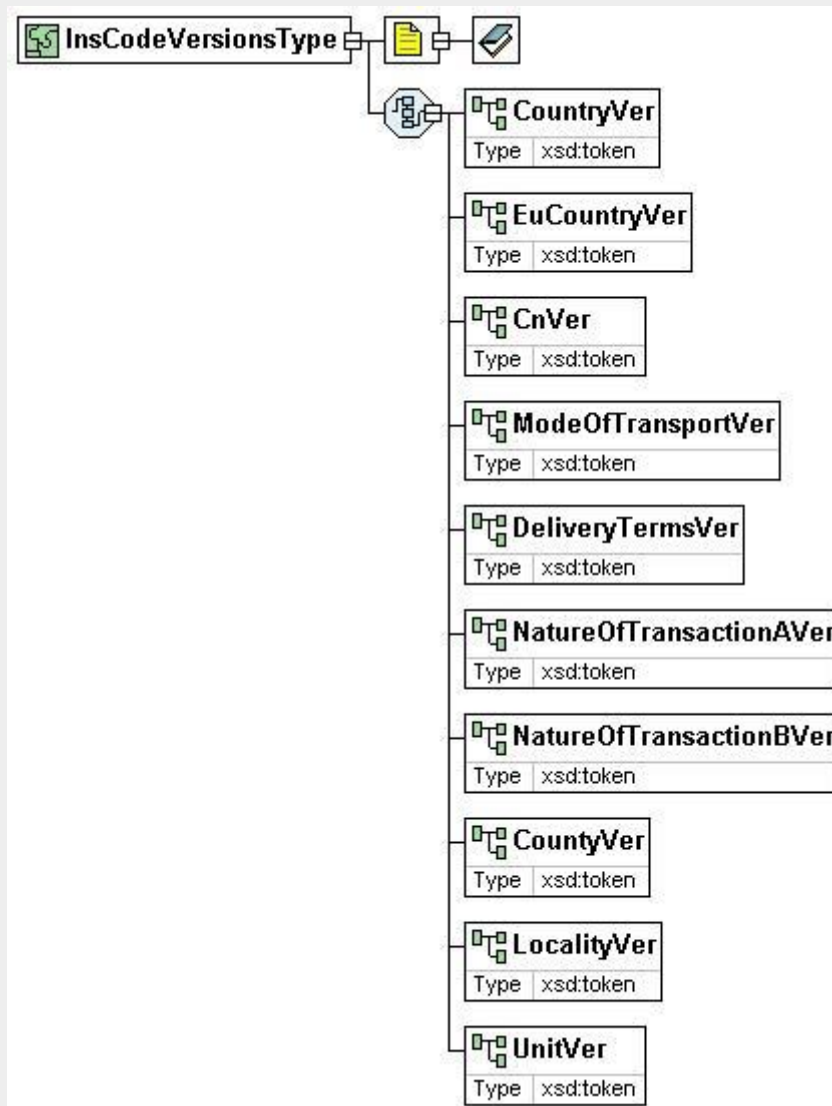
```
<xsd:complexType name="InsArrivalItemType">
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationItemType">
      <xsd:sequence>
        <xsd:element name="CountryOfConsignment" type="CountryType" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Complex Type: [InsCodeVersionsType](#)

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

Name	InsCodeVersionsType
<u>Abstract</u>	no
Documentation	Information about the nomenclatures used in the declaration and their version.

Diagram



XML Instance Representation

```
<...>
  <CountryVer> xsd:token </CountryVer> [1]
  <EuCountryVer> xsd:token </EuCountryVer> [1]
  <CnVer> xsd:token </CnVer> [1]
  <ModeOfTransportVer> xsd:token </ModeOfTransportVer> [1]
  <DeliveryTermsVer> xsd:token </DeliveryTermsVer> [1]
  <NatureOfTransactionAVer> xsd:token </NatureOfTransactionAVer> [1]
  <NatureOfTransactionBVer> xsd:token </NatureOfTransactionBVer> [1]
  <CountyVer> xsd:token </CountyVer> [1]
  <LocalityVer> xsd:token </LocalityVer> [1]
  <UnitVer> xsd:token </UnitVer> [1]
</...>
```

Schema Component Representation

```
<xsd:complexType name="InsCodeVersionsType">
  <xsd:sequence>
    <xsd:element name="CountryVer" type="xsd:token"/>
    <xsd:element name="EuCountryVer" type="xsd:token"/>
    <xsd:element name="CnVer" type="xsd:token"/>
    <xsd:element name="ModeOfTransportVer" type="xsd:token"/>
    <xsd:element name="DeliveryTermsVer" type="xsd:token"/>
    <xsd:element name="NatureOfTransactionAVer" type="xsd:token"/>
    <xsd:element name="NatureOfTransactionBVer" type="xsd:token"/>
```

```

    <xsd:element name="CountyVer" type="xsd:token"/>
    <xsd:element name="LocalityVer" type="xsd:token"/>
    <xsd:element name="UnitVer" type="xsd:token"/>
  </xsd:sequence>
</xsd:complexType>

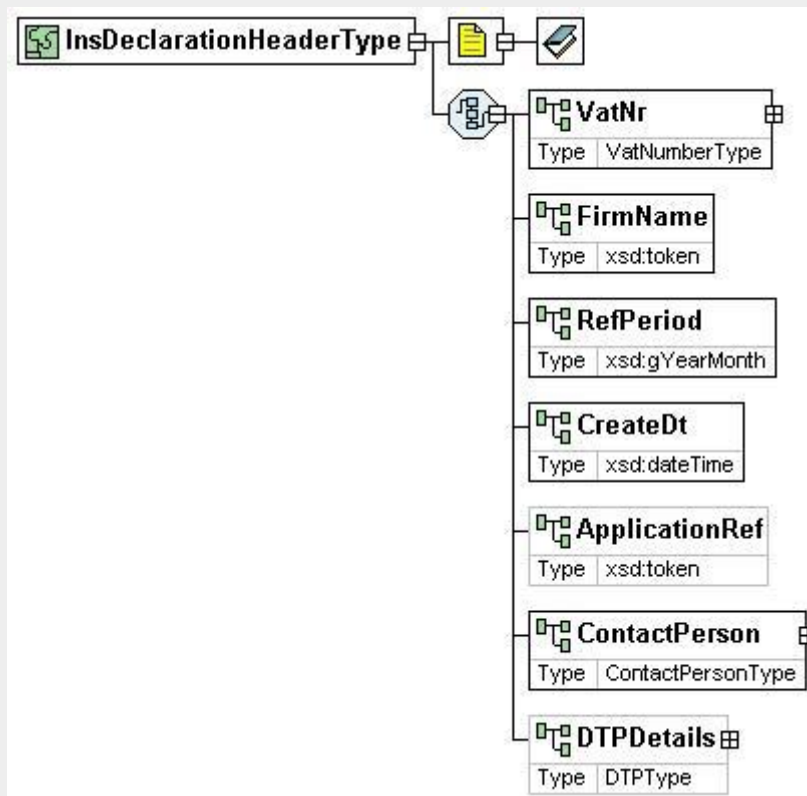
```

Complex Type: **InsDeclarationHeaderType**

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

Name	InsDeclarationHeaderType
<u>Abstract</u>	no
Documentation	Information that makes up the declaration header: - VAT number: - Name of the firm - Reference period - Date of creation - Application Reference (this is not to be completed by the declarant) - DTP details

Diagram



XML Instance Representation

<...>

```

<VatNr> VatNumberType </VatNr> [1]
<FirmName> xsd:token </FirmName> [1]
<RefPeriod> xsd:gYearMonth </RefPeriod> [1]
<CreateDt> xsd:dateTime </CreateDt> [1]
<ApplicationRef> xsd:token </ApplicationRef> [0..1]
  
```

```
<ContactPerson> ContactPersonType </ContactPerson> [1]
<DTPDetails> DTPTType </DTPDetails> [0..1]
</...>
```

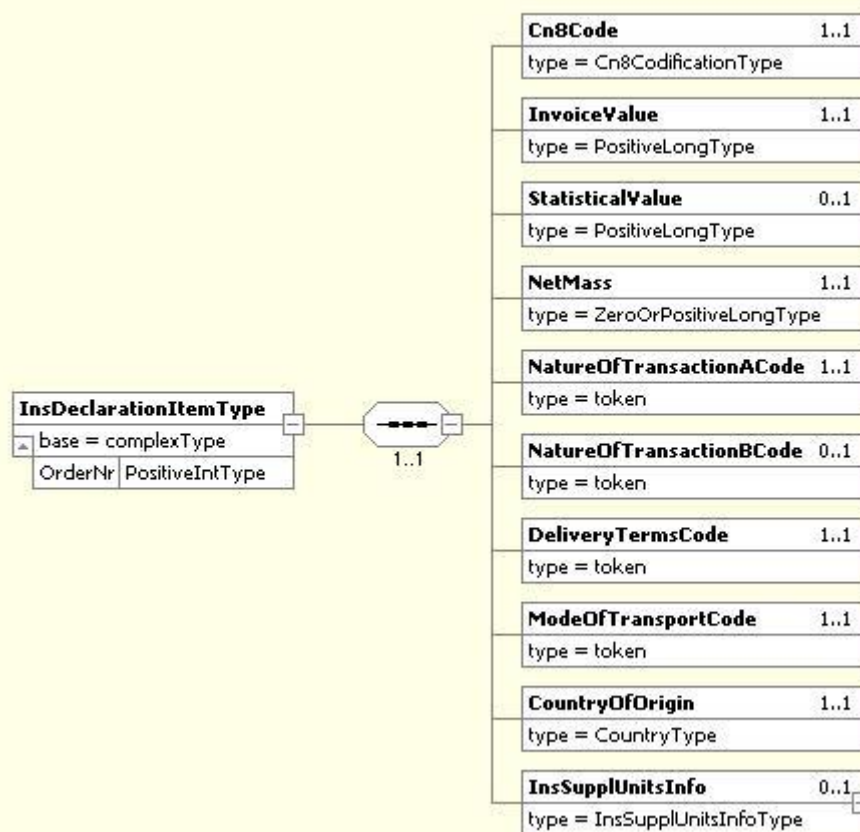
Schema Component Representation

```
<xsd:complexType name="InsDeclarationHeaderType">
  <xsd:sequence>
    <xsd:element name="VatNr" type="VatNumberType"/>
    <xsd:element name="FirmName" type="xsd:token"/>
    <xsd:element name="RefPeriod" type="xsd:gYearMonth"/>
    <xsd:element name="CreateDt" type="xsd:dateTime"/>
    <xsd:element name="ApplicationRef" type="xsd:token" minOccurs="0"/>
    <xsd:element name="ContactPerson" type="ContactPersonType"/>
    <xsd:element name="DTPDetails" type="DTPTType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Complex Type: *InsDeclarationItemType*

<i>Super-types:</i>	None
<i>Sub-types:</i>	• InsArrivalItemType (by extension) • InsDispatchItemType (by extension)

Name	InsDeclarationItemType
<u>Abstract</u>	yes
Diagram	



XML Instance Representation

```
<...
  OrderNr="PositiveIntType [0..1]">
    <Cn8Code> Cn8CodificationType </Cn8Code> [1]
    <InvoiceValue> PositiveLongType </InvoiceValue> [1]
```

```

<StatisticalValue> PositiveLongType </StatisticalValue> [0..1]
<NetMass> PositiveLongType </NetMass> [1]
<NatureOfTransactionACode> xsd:token </NatureOfTransactionACode> [1]
<NatureOfTransactionBCode> xsd:token </NatureOfTransactionBCode> [0..1]
<DeliveryTermsCode> xsd:token </DeliveryTermsCode> [1]
<ModeOfTransportCode> xsd:token </ModeOfTransportCode> [1]
<CountryOfOrigin> CountryType </CountryOfOrigin> [1]
<InsSupplUnitsInfo> InsSupplUnitsInfoType </InsSupplUnitsInfo> [0..1]
</...>

```

Schema Component Representation

```

<xsd:complexType name="InsDeclarationItemType" abstract="true">
  <xsd:sequence>
    <xsd:element name="Cn8Code" type="Cn8CodificationType"/>
    <xsd:element name="InvoiceValue" type="PositiveLongType"/>
    <xsd:element name="StatisticalValue" type="PositiveLongType" minOccurs="0"/>
    <xsd:element name="NetMass" type="PositiveLongType"/>
    <xsd:element name="NatureOfTransactionACode" type="xsd:token"/>
    <xsd:element name="NatureOfTransactionBCode" type="xsd:token" minOccurs="0"/>
    <xsd:element name="DeliveryTermsCode" type="xsd:token"/>
    <xsd:element name="ModeOfTransportCode" type="xsd:token"/>
    <xsd:element name="CountryOfOrigin" type="CountryType"/>
    <xsd:element name="InsSupplUnitsInfo" type="InsSupplUnitsInfoType" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

```

```

</xsd:sequence>
  <xsd:attribute name="OrderNr" type="PositiveIntType"/>
</xsd:complexType>

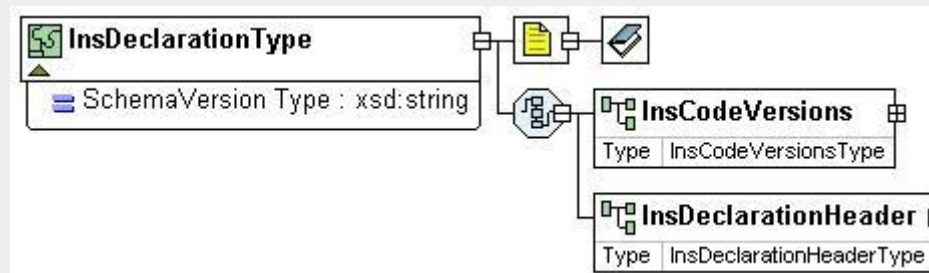
```

Complex Type: **InsDeclarationType**

<i>Super-types:</i>	None
<i>Sub-types:</i>	• InsNilArrivalType (by extension) • InsNilDispatchType (by extension) • InsNewArrivalType (by extension) • InsRevisedArrivalType (by extension) • InsNewDispatchType (by extension) • InsRevisedDispatchType (by extension)

Name	InsDeclarationType
<u>Abstract</u>	yes
Documentation	The abstract definition of a declaration. Attribute "SchemaVersion" is a string constant and must be set always to "1.0".

Diagram



XML Instance Representation

```

<...
  SchemaVersion="1.0 [1]">
    <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
    <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
  </...>
  
```

Schema Component Representation

```

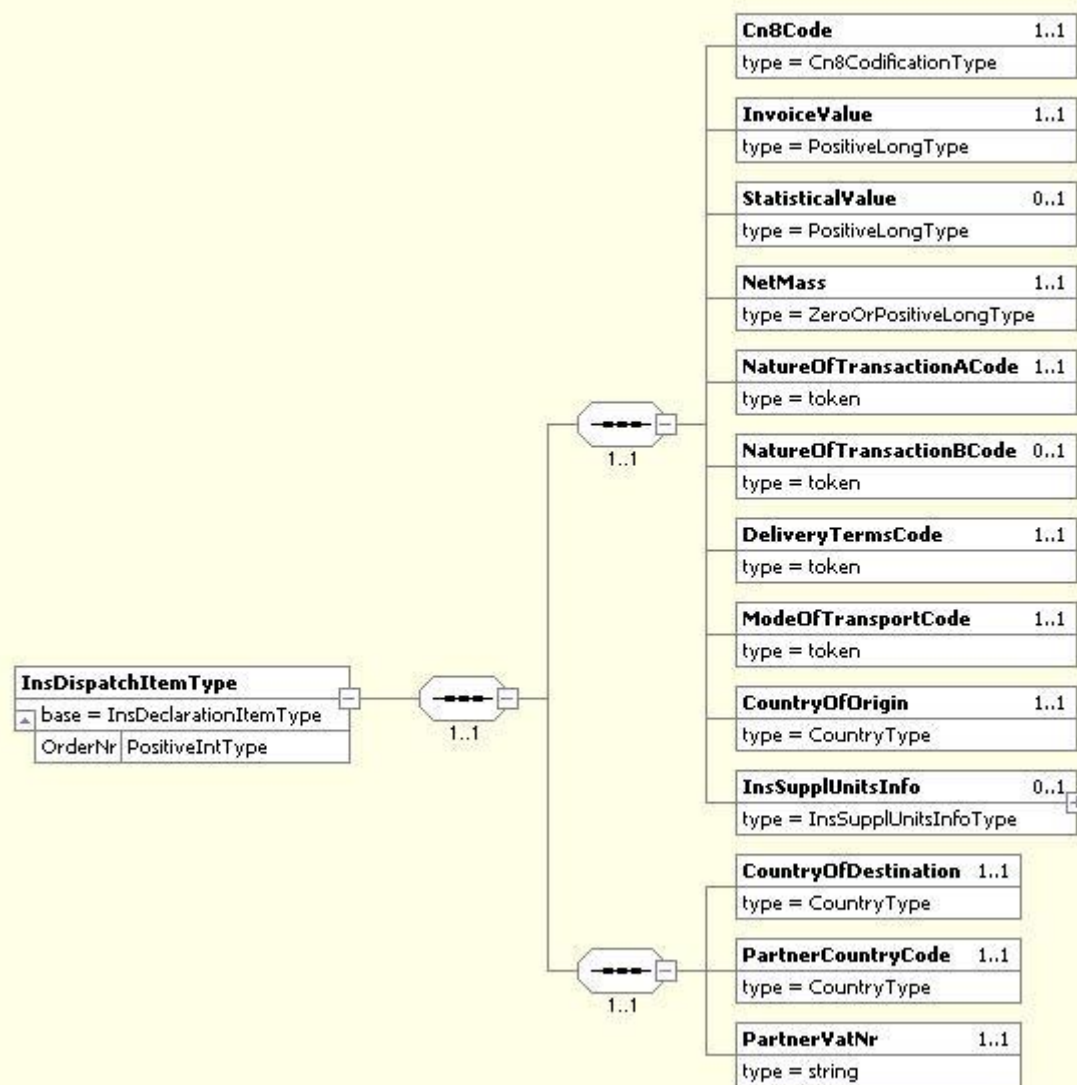
<xsd:complexType name="InsDeclarationType" abstract="true">
  <xsd:sequence>
    <xsd:element name="InsCodeVersions" type="InsCodeVersionsType"/>
    <xsd:element name="InsDeclarationHeader" type="InsDeclarationHeaderType"/>
  </xsd:sequence>
  <xsd:attribute name="SchemaVersion" type="xsd:string" use="required" fixed="1.0"/>
</xsd:complexType>
  
```

Complex Type: *InsDispatchItemType*

Super-types: [InsDeclarationItemType](#) < **InsDispatchItemType** (by extension)

Sub-types: None

Name	InsDispatchItemType
<u>Abstract</u>	No
Documentation	The declaration item of a declaration for dispatches.
Diagram	



XML Instance Representation

```
<...
  OrderNr="PositiveIntType [0..1]">
    <Cn8Code> Cn8CodificationType </Cn8Code> [1]
    <InvoiceValue> PositiveLongType </InvoiceValue> [1]
    <StatisticalValue> PositiveLongType </StatisticalValue> [0..1]
    <NetMass> PositiveLongType </NetMass> [1]
    <NatureOfTransactionACode> xsd:token </NatureOfTransactionACode> [1]
    <NatureOfTransactionBCode> xsd:token </NatureOfTransactionBCode> [0..1]
    <DeliveryTermsCode> xsd:token </DeliveryTermsCode> [1]
    <ModeOfTransportCode> xsd:token </ModeOfTransportCode> [1]
    <InsSupplUnitsInfo> InsSupplUnitsInfoType </InsSupplUnitsInfo> [0..1]
    <CountryOfDestination> CountryType </CountryOfDestination> [1]
    <PartnerCountryCode> CountryType </PartnerCountryCode> [1]
    <PartnerVatNr> xsd:string </PartnerVatNr> [1]
  </...>
```

Schema Component Representation

```
<xsd:complexType name="InsDispatchItemType">
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationItemType">
      <xsd:sequence>
        <xsd:element name="CountryOfDestination" type="CountryType"/>
        <xsd:element name="PartnerCountryCode" type="CountryType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

        <xsd:element name="PartnerVatNr" type="xsd:string"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

Complex Type: InsNewArrivalType

<i>Super-types:</i>	InsDeclarationType < InsNewArrivalType (by extension)
<i>Sub-types:</i>	None

Name	InsNewArrivalType
<u>Abstract</u>	no
Documentation	The declaration for arrivals

Diagram



XML Instance Representation

```
<...
  SchemaVersion="1.0 [1]">
    <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
    <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
    <InsArrivalItem> InsArrivalItemType </InsArrivalItem> [1..*]
  </...>
```

Schema Component Representation

```
<xsd:complexType name="InsNewArrivalType">
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType">
      <xsd:sequence>
        <xsd:element name="InsArrivalItem" type="InsArrivalItemType" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Complex Type: **InsNewDispatchType**

Super-types: [InsDeclarationType](#) < **InsNewDispatchType** (by extension)

Sub-types: None

Name	InsNewDispatchType
<u>Abstract</u>	no
Documentation	The declaration for dispatches
Diagram	

XML Instance Representation

```

<...
  SchemaVersion="1.0 [1]">
    <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
    <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
    <InsDispatchItem> InsDispatchItemType </InsDispatchItem> [1..*]
```

</...>

Schema Component Representation

```

<xsd:complexType name="InsNewDispatchType">
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType">
      <xsd:sequence>
        <xsd:element name="InsDispatchItem" type="InsDispatchItemType" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

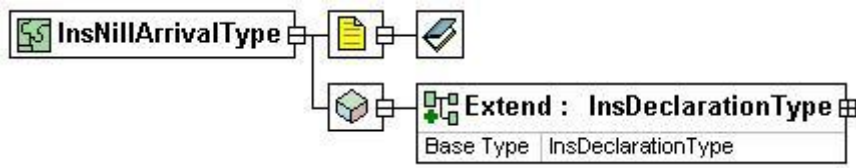
```

Complex Type: InsNillArrivalType

Super-types: [InsDeclarationType](#) < **InsNillArrivalType** (by extension)

Sub-types: None

Name	InsNillArrivalType
------	--------------------

<u>Abstract</u>	no
Documentation	The null declaration for arrivals
Diagram	

XML Instance Representation

```

<...
  SchemaVersion="1.0 [1]">
    <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
    <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
  </...>

```

Schema Component Representation

```

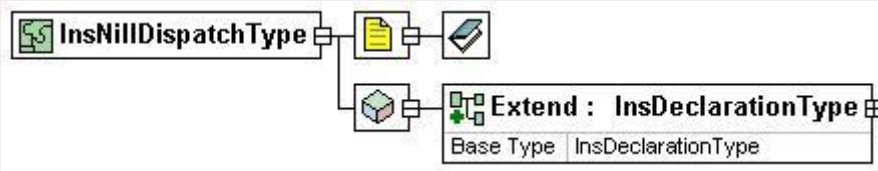
<xsd:complexType name="InsNullArrivalType">
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType"/>
  </xsd:complexContent>
</xsd:complexType>

```

Complex Type: **InsNilDispatchType**

Super-types: [InsDeclarationType](#) < **InsNilDispatchType** (by extension)

Sub-types: None

Name	InsNilDispatchType
<u>Abstract</u>	no
Documentation	The nil declaration for dispatches
Diagram	

XML Instance Representation

```
<...
  SchemaVersion="1.0 [1]">
    <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
    <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
</...>
```

Schema Component Representation

```
<xsd:complexType name="InsNilIDispatchType">
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType"/>
  </xsd:complexContent>
</xsd:complexType>
```

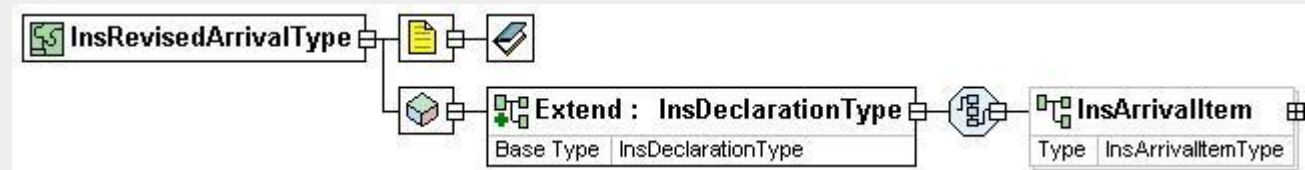
Complex Type: **InsRevisedArrivalType**

Super-types: [InsDeclarationType](#) < **InsRevisedArrivalType** (by extension)

Sub-types: None

Name	InsRevisedArrivalType
<u>Abstract</u>	no
Documentation	The revised declaration for arrivals

Diagram



XML Instance Representation

```

<...
  SchemaVersion="1.0 [1]">
    <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
    <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
    <InsArrivalItem> InsArrivalItemType </InsArrivalItem> [0..*]
  </...>

```

Schema Component Representation

```

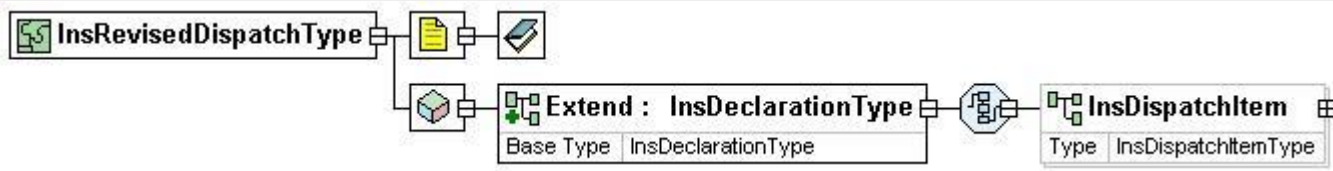
<xsd:complexType name="InsRevisedArrivalType">
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType">
      <xsd:sequence>
        <xsd:element name="InsArrivalItem" type="InsArrivalItemType" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

Complex Type: **InsRevisedDispatchType**

Super-types: [InsDeclarationType](#) < **InsRevisedDispatchType** (by extension)

Sub-types: None

Name	InsRevisedDispatchType
<u>Abstract</u>	no
Documentation	The revised declaration for dispatches
Diagram	

XML Instance Representation

```

<...
  SchemaVersion="1.0 [1]">
    <InsCodeVersions> InsCodeVersionsType </InsCodeVersions> [1]
    <InsDeclarationHeader> InsDeclarationHeaderType </InsDeclarationHeader> [1]
    <InsDispatchItem> InsDispatchItemType </InsDispatchItem> [0..*]
  
```

</...>

Schema Component Representation

```

<xsd:complexType name="InsRevisedDispatchType">
  <xsd:complexContent>
    <xsd:extension base="InsDeclarationType">
      <xsd:sequence>
        <xsd:element name="InsDispatchItem" type="InsDispatchItemType" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

Complex Type: [InsSupplUnitsInfoType](#)

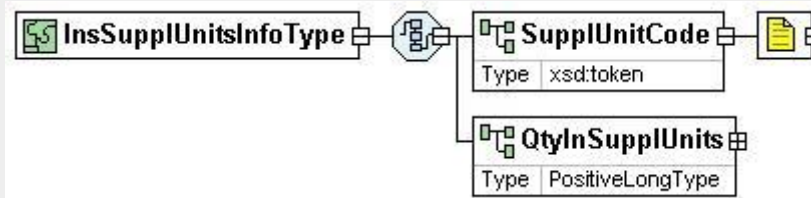
Super-types: None

Sub-types: None

Name	InsSupplUnitsInfoType
------	-----------------------

Abstract

no

Diagram**XML Instance Representation**

```

<...>
  <SupplUnitCode> xsd:token </SupplUnitCode> [1]
  <QtyInSupplUnits> PositiveLongType </QtyInSupplUnits> [1]
</...>

```

Schema Component Representation

```

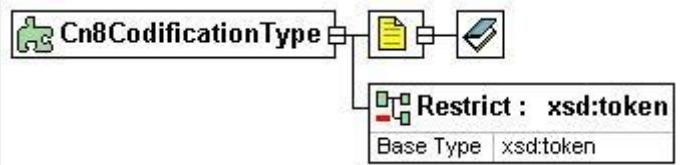
<xsd:complexType name="InsSupplUnitsInfoType">
  <xsd:sequence>
    <xsd:element name="SupplUnitCode" type="xsd:token"/>
    <xsd:element name="QtyInSupplUnits" type="PositiveLongType"/>
  </xsd:sequence>
</xsd:complexType>

```

Simple Type: [Cn8CodificationType](#)

Super-types: [xsd:token](#) < **Cn8CodificationType** (by restriction)

Sub-types: None

Name	Cn8CodificationType
Content	- Base XSD Type: token <i>pattern</i> = [0-9]{8}
Documentation	The 8-digit CN8 commodity/item code. See the corresponding CN8 nomenclature.
Diagram	

Schema Component Representation

```

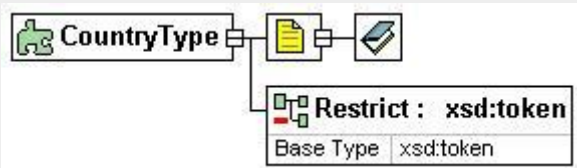
<xsd:simpleType name="Cn8CodificationType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-9]{8}"/>
  </xsd:restriction>
</xsd:simpleType>

```

Simple Type: **CountryType**

Super-types: [xsd:token](#) < **CountryType** (by restriction)

Sub-types: None

Name	CountryType
Content	- Base XSD Type: token <i>length</i> >= 1
Documentation	The code number for the country. See the corresponding country nomenclature.
Diagram	

Schema Component Representation

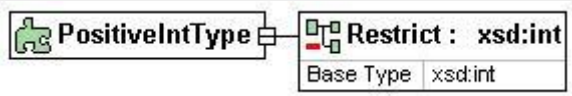
```
<xsd:simpleType name="CountryType">
  <xsd:restriction base="xsd:token">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="2"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
</xsd:simpleType>
```

Simple Type: **PositiveIntType**

Super-types: [xsd:int](#) < **PositiveIntType** (by restriction)

Sub-types: None

Name	PositiveIntType
Content	- Base XSD Type: int $value > 0$
Diagram	

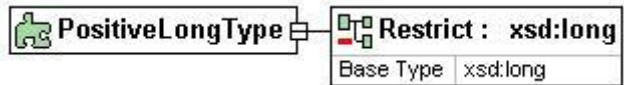
Schema Component Representation

```
<xsd:simpleType name="PositiveIntType">
  <xsd:restriction base="xsd:int">
    <xsd:minExclusive value="0"/>
  </xsd:restriction>
</xsd:simpleType>
```

Simple Type: **PositiveLongType**

Super-types: [xsd:long](#) < **PositiveLongType** (by restriction)

Sub-types: None

Name	PositiveLongType
Content	- Base XSD Type: long <i>value</i> > 0
Diagram	

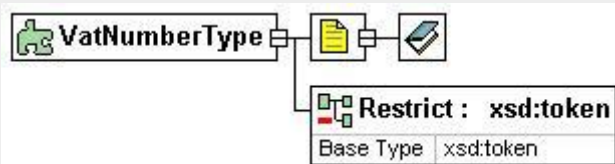
Schema Component Representation

```
<xsd:simpleType name="PositiveLongType">
  <xsd:restriction base="xsd:long">
    <xsd:minExclusive value="0"/>
  </xsd:restriction>
</xsd:simpleType>
```

Simple Type: **VatNumberType**

Super-types: [xsd:token](#) < **VatNumberType** (by restriction)

Sub-types: None

Name	VatNumberType
Content	- Base XSD Type: token <i>pattern</i> = [0-9]{10}
Documentation	The 10-digit string corresponding to the VAT number of the firm
Diagram	 <pre> classDiagram class VatNumberType class Restrict["Restrict : xsd:token"] Restrict --> VatNumberType Restrict -- Base Type xsd:token </pre>

Schema Component Representation

```

<xsd:simpleType name="VatNumberType">
  <xsd:restriction base="xsd:token">
    <xsd:pattern value="[0-9]{10}"/>
  </xsd:restriction>
</xsd:simpleType>

```

Legend

Clarifications on how to use the XML Instance Representation:

<... country="Australia" >

<unitNo> string </unitNo> [0..1]

<houseNo> string </houseNo> [1]

<street> string </street> [1]

<state> AusStates </state> [1]

<postcode> string <<pattern = [1-9][0-9]{3}>> </postcode> [1]

</...>

The XML Instance Representation above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- Attribute "SchemaVersion" has a fixed value "1.0"
- Otherwise, the type of the element/attribute is displayed.
- If the element/attribute's type is in the schema, a link is provided to it.
- For local simple type definitions, the constraints are displayed in angle brackets, e.g. <<pattern = [1-9][0-9]{3}>>.