

Schema documentation for DR-GW-SDS.xsd

november 5, 2024

Table of Contents

Namespace: "DR-GW-Interface/DR-GW-SDS"	2
Schema(s)	2
Main schema DR-GW-SDS.xsd	2
Element(s)	2
Element SDS_Send	2
Element SDS_Send / sds	3
Element SDS_SendReport	3
Element SDS_SendReport / target	4
Element SDS_SendReport / msgRef	5
Element SDS_SendReport / deliveryStatus	5
Namespace: "DR-GW-Interface/DR-GW-SDS.CommonTypes"	6
Schema(s)	6
Imported schema DR-GW-SDS.CommonTypes.xsd	6
Element(s)	6
Element typeSDS / protocol	6
Element typeSDS / sdsType	6
Element typeSDS / msgRef	6
Element typeSDS / report	7
Element typeSDS / sdsdata	7
Element typeSDSData / data	7
Element typeSDSData / hexdata	8
Element typeSDSData / hexdatalength	8
Element typeSDS / source	8
Element typeSDS / target	9
Element typeSDS / forward	9
Element typeSDS / validity	10
Element typeSDS / tstamp	10
Element typeSDS / encryption	11
Element typeSDS / e2eegroup	11
Element typeSDSValidity / value	11
Complex Type(s)	11
Complex Type typeSDS	11
Complex Type typeSDSData	12
Complex Type typeSDSValidity	13
Simple Type(s)	13
Simple Type typeSDSType	13
Simple Type typeReport	14
Namespace: "DR-GW-Interface/CommonTypes"	15
Schema(s)	15
Imported schema CommonTypes.xsd	15
Element(s)	15
Element ct:typeRequest / ct:requestId	15
Element ct:typeAddress / ct:subscriber	15
Element ct:typeSubscriberAddress / ct:ssi	15
Element ct:typeSubscriberAddress / ct:tsi	16
Element ct:typeTSI / ct:mnc	16
Element ct:typeTSI / ct:mcc	16
Element ct:typeTSI / ct:ssi	17
Element ct:typeAddress / ct:alias	17
Element ct:typeAddress / ct:msisdn	17
Element ct:typeAddress / ct:fssn	17
Element ct:typeAddress / ct:external	18
Element ct:typeExternal / ct:gatewayNumber	18
Element ct:typeExternal / ct:number	18
Element ct:typeAddress / ct:opta	18
Element ct:typeAddress / ct:cell	19
Element ct:typeResult / ct:responseCode	19
Element ct:typeResult / ct:sourceSystem	19
Element ct:typeResult / ct:result	19
Element ct:typeResponse / ct:requestId	20
Element ct:typeResponse / ct:result	20
Element ct:typeEvent / ct:requestId	20

Element ct:typeEvent / ct:result	21
Complex Type(s)	21
Complex Type ct:typeRequest	21
Complex Type ct:typeAddress	21
Complex Type ct:typeSubscriberAddress	22
Complex Type ct:typeTSI	22
Complex Type ct:typeExternal	23
Complex Type ct:typeResult	23
Complex Type ct:typeResponse	24
Complex Type ct:typeEvent	24
Complex Type ct:typeEmpty	24
Simple Type(s)	25
Simple Type ct:typeDialString	25
Simple Type ct:typeOPTA	25
Simple Type ct:typeResponseCode	25
Simple Type ct:typeSourceSystem	26
Simple Type ct:typeAddressingStyle	26

Namespace: "DR-GW-Interface/DR-GW-SDS"

Schema(s)

Main schema DR-GW-SDS.xsd

Namespace	DR-GW-Interface/DR-GW-SDS
Annotations	Version 1.1.1
Properties	attribute form default: unqualified
	element form default: qualified

Element(s)

Element SDS_Send

Namespace	DR-GW-Interface/DR-GW-SDS
Annotations	
Diagram	<pre> classDiagram class SDS_Send { <<Extension of 'ct:typeRequest'>> } class ct.typeRequest { <<extension base>> attribute requestId attribute sds } SDS_Send < -- ct.typeRequest </pre>
Type	extension of ct:typeRequest
Type hierarchy	<ul style="list-style-type: none"> • ct:typeRequest
Properties	content: complex
Model	ct:requestId , sds
Children	ct:requestId, sds
Instance	<pre> <SDS_Send xmlns="DR-GW-Interface/DR-GW-SDS" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{1,1}</ct:requestId> <sds>{1,1}</sds> </SDS_Send> </pre>
Source	<pre> <x:element name="SDS_Send"> <x:annotation> <x:documentation></x:documentation> </x:annotation> <x:complexType> <x:complexContent> <x:extension base="ct:typeRequest"> <x:sequence> <x:element name="sds" type="ct:typeSDS"/> </x:sequence> </x:extension> </x:complexContent> </x:complexType> </x:element> </pre>

```

</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>

```

Element SDS_Send / sds

Namespace	DR-GW-Interface/DR-GW-SDS
Diagram	<pre> classDiagram class sds { <<Type ctS:typeSDS>> } class ctS:typeSDS { protocol sdsType msgRef report sdsdata source target forward validity tstamp encryption e2eegroup } sds "1" --> "1" ctS:typeSDS : Type </pre>
Type	typeSDS
Properties	content: complex
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, tstamp, validity
Instance	<pre> <sds xmlns="DR-GW-Interface/DR-GW-SDS" xmlns:ctS="DR-GW-Interface/DR-GW-SDS.CommonTypes"> <ctS:protocol>{0,1}</ctS:protocol> <ctS:sdsType>{1,1}</ctS:sdsType> <ctS:msgRef>{0,1}</ctS:msgRef> <ctS:report>{0,1}</ctS:report> <ctS:sdsdata>{1,1}</ctS:sdsdata> <ctS:source>{0,1}</ctS:source> <ctS:target>{1,1}</ctS:target> <ctS:forward>{0,1}</ctS:forward> <ctS:validity>{0,1}</ctS:validity> <ctS:tstamp>{0,1}</ctS:tstamp> <ctS:encryption>{0,1}</ctS:encryption> <ctS:e2eegroup>{0,1}</ctS:e2eegroup> </sds> </pre>
Source	<pre> <xs:element name="sds" type="ctS:typeSDS"/> </pre>

Element SDS_SendReport

Namespace	DR-GW-Interface/DR-GW-SDS
Annotations	

Diagram	<pre> classDiagram ct:typeRequest < -- SDS_SendReport SDS_SendReport { requestId : ct:typeAddress target : ct:typeAddress msgRef : xs:unsignedByte deliveryStatus : xs:unsignedByte } </pre>
Type	extension of ct:typeRequest
Type hierarchy	<ul style="list-style-type: none"> • ct:typeRequest
Properties	content: complex
Model	ct:requestId , target , msgRef , deliveryStatus
Children	ct:requestId, deliveryStatus, msgRef, target
Instance	<pre> <SDS_SendReport xmlns="DR-GW-Interface/DR-GW-SDS" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{1,1}</ct:requestId> <target>{1,1}</target> <msgRef>{1,1}</msgRef> <deliveryStatus>{1,1}</deliveryStatus> </SDS_SendReport> </pre>
Source	<pre> <xs:element name="SDS_SendReport"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="ct:typeRequest"> <xs:sequence> <xs:element name="target" type="ct:typeAddress"/> <xs:element name="msgRef" type="xs:unsignedByte"/> <xs:element name="deliveryStatus" type="xs:unsignedByte"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element> </pre>

Element SDS_SendReport / target

Namespace	DR-GW-Interface/DR-GW-SDS
-----------	---------------------------

Diagram	<p>ct:typeAddress</p> <ul style="list-style-type: none"> subscriber alias msisdn fssn <ul style="list-style-type: none"> Fleet specific short number external opta cell <p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>
Type	ct:typeAddress
Properties	content: complex
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Instance	<pre><target xmlns="DR-GW-Interface/DR-GW-SDS" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:subscriber>{0,1}</ct:subscriber> <ct:alias>{0,1}</ct:alias> <ct:msisdn>{0,1}</ct:msisdn> <ct:fssn>{0,1}</ct:fssn> <ct:external>{0,1}</ct:external> <ct:opta>{0,1}</ct:opta> <ct:cell>{0,1}</ct:cell> </target></pre>
Source	<code><xss:element name="target" type="ct:typeAddress" /></code>

Element SDS_SendReport / msgRef

Namespace	DR-GW-Interface/DR-GW-SDS
Diagram	<p>msgRef</p> <p>Type xs:unsignedByte</p> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>
Type	xs:unsignedByte
Properties	content: simple
Source	<code><xss:element name="msgRef" type="xs:unsignedByte" /></code>

Element SDS_SendReport / deliveryStatus

Namespace	DR-GW-Interface/DR-GW-SDS
Diagram	<p>deliveryStatus</p> <p>Type xs:unsignedByte</p> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>

Type	xs:unsignedByte
Properties	content: simple
Source	<xs:element name="deliveryStatus" type="xs:unsignedByte" />

Namespace: "DR-GW-Interface/DR-GW-SDS.CommonTypes"

Schema(s)

Imported schema DR-GW-SDS.CommonTypes.xsd

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Annotations	Version 1.1.1
Properties	attribute form default: unqualified
	element form default: qualified

Element(s)

Element typeSDS / protocol

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<p>protocol</p> <p>xs:unsignedByte</p> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0
Source	<xs:element name="protocol" type="xs:unsignedByte" minOccurs="0" />

Element typeSDS / sdsType

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes																		
Diagram	<p>sdsType</p> <p>typeSDSType</p>																		
Type	typeSDSType																		
Properties	content: simple																		
Facets	<table> <tr> <td>enumeration</td> <td>0</td> <td>SDS1.</td> </tr> <tr> <td>enumeration</td> <td>1</td> <td>SDS2.</td> </tr> <tr> <td>enumeration</td> <td>2</td> <td>SDS3.</td> </tr> <tr> <td>enumeration</td> <td>3</td> <td>SDS4.</td> </tr> <tr> <td>enumeration</td> <td>4</td> <td>SDS-TL.</td> </tr> <tr> <td>enumeration</td> <td>5</td> <td>Status.</td> </tr> </table>	enumeration	0	SDS1.	enumeration	1	SDS2.	enumeration	2	SDS3.	enumeration	3	SDS4.	enumeration	4	SDS-TL.	enumeration	5	Status.
enumeration	0	SDS1.																	
enumeration	1	SDS2.																	
enumeration	2	SDS3.																	
enumeration	3	SDS4.																	
enumeration	4	SDS-TL.																	
enumeration	5	Status.																	
Source	<xs:element name="sdsType" type="typeSDSType" />																		

Element typeSDS / msgRef

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<p>msgRef</p> <p>xs:unsignedByte</p> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>

Type	xs:unsignedByte
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="msgRef" type="xs:unsignedByte" minOccurs="0" />

Element typesSDS / report

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<pre> classDiagram class report class typeReport report < -- typeReport </pre>
Type	typeReport
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>default: none</p>
Facets	<p>enumeration none</p> <p>enumeration delivery</p> <p>enumeration consume</p> <p>enumeration both</p>
Source	<xs:element name="report" type="typeReport" default="none" minOccurs="0" />

Element typesSDS / sdsdata

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<pre> classDiagram class sdsdata class typeSDSDData typeSDSDData < -- sdsdata class data class hexdata class hexdatalength typeSDSDData <--> data : typeSDSDData <--> hexdata : typeSDSDData <--> hexdatalength : </pre> <p>2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will...</p>
Type	typeSDSDData
Properties	content: complex
Model	data{0,1} , hexdata{0,1} , hexdatalength{0,1}
Children	data, hexdata, hexdatalength
Instance	<sdsdata xmlns="DR-GW-Interface/DR-GW-SDS.CommonTypes"> <data>{0,1}</data> <hexdata>{0,1}</hexdata> <hexdatalength>{0,1}</hexdatalength> </sdsdata>
Source	<xs:element name="sdsdata" type="typeSDSDData" />

Element typeSDSDData / data

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<pre> classDiagram class data class xsString data < -- xsString </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>

Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="data" type="xs:string" minOccurs="0" />

Element typesSDSData / hexdata

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<pre> graph LR hexdata[hexdata] --> hexBinary[xs:hexBinary] style hexdata fill:#f0f0f0 style hexBinary fill:#e0e0e0 </pre>
Type	xs:hexBinary
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="hexdata" type="xs:hexBinary" minOccurs="0" />

Element typesSDSData / hexdatalength

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<pre> graph LR hexdatalength[hexdatalength] --> integer[xs:integer] style hexdatalength fill:#f0f0f0 style integer fill:#e0e0e0 </pre>
Type	xs:integer
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="hexdatalength" type="xs:integer" minOccurs="0" />

Element typesSDS / source

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<pre> graph LR source[source] --> typeAddress[ct:typeAddress] style source fill:#f0f0f0 style typeAddress fill:#e0e0e0 </pre> <p>The diagram shows the 'source' element pointing to the 'ct:typeAddress' complex type. The 'ct:typeAddress' type is defined as follows:</p> <ul style="list-style-type: none"> It has a sequence of zero or more elements: 'subscriber', 'alias', 'msisdn', 'fssn', 'external', 'opta', and 'cell'. Each of these elements has a plus sign (+) next to it, indicating they are optional. The 'fssn' element has a callout box stating: "Fleet specific short number". A large callout box at the bottom states: "Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA)."

Type	ct:typeAddress
Properties	<p>content: complex</p> <p>minOccurs: 0</p>
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Instance	<pre><source xmlns="DR-GW-Interface/DR-GW-SDS.CommonTypes" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:subscriber>{0,1}</ct:subscriber> <ct:alias>{0,1}</ct:alias> <ct:msisdn>{0,1}</ct:msisdn> <ct:fssn>{0,1}</ct:fssn> <ct:external>{0,1}</ct:external> <ct:opta>{0,1}</ct:opta> <ct:cell>{0,1}</ct:cell> </source></pre>
Source	<code><xss:element name="source" type="ct:typeAddress" minOccurs="0" /></code>

Element typesDS / target

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<p>The diagram illustrates the structure of the <code>ct:typeAddress</code> element. It is a composite element containing seven children: <code>subscriber</code>, <code>alias</code>, <code>msisdn</code>, <code>fssn</code>, <code>external</code>, <code>opta</code>, and <code>cell</code>. A separate <code>target</code> element is shown connected to the <code>fssn</code> child. A callout box below the tree indicates: "Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA)."/></p>
Type	ct:typeAddress
Properties	content: complex
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Instance	<pre><target xmlns="DR-GW-Interface/DR-GW-SDS.CommonTypes" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:subscriber>{0,1}</ct:subscriber> <ct:alias>{0,1}</ct:alias> <ct:msisdn>{0,1}</ct:msisdn> <ct:fssn>{0,1}</ct:fssn> <ct:external>{0,1}</ct:external> <ct:opta>{0,1}</ct:opta> <ct:cell>{0,1}</ct:cell> </target></pre>
Source	<code><xss:element name="target" type="ct:typeAddress" /></code>

Element typesDS / forward

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
-----------	---------------------------------------

Diagram	<p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>				
Type	ct:typeAddress				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}				
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber				
Instance	<pre><forward xmlns="DR-GW-Interface/DR-GW-SDS.CommonTypes" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:subscriber>{0,1}</ct:subscriber> <ct:alias>{0,1}</ct:alias> <ct:msisdn>{0,1}</ct:msisdn> <ct:fssn>{0,1}</ct:fssn> <ct:external>{0,1}</ct:external> <ct:opta>{0,1}</ct:opta> <ct:cell>{0,1}</ct:cell> </forward></pre>				
Source	<code><xss:element name="forward" type="ct:typeAddress" minOccurs="0" /></code>				

Element typeSDS / validity

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes				
Diagram	<p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>				
Type	xs:unsignedByte				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">simple</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="validity" type="xs:unsignedByte" minOccurs="0" /></code>				

Element typeSDS / tstamp

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<p>Built-in primitive type. The dateTime datatype represents a specific instant of time.</p>

Type	xs:dateTime
Properties	content: simple minOccurs: 0
Source	<xs:element name="tstamp" type="xs:dateTime" minOccurs="0" />

Element typeSDS / encryption

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<p>The diagram shows a class named 'encryption' with a multiplicity of 0..1. It has a directed association labeled with a minus sign (-) pointing to a class named 'xs:boolean'. A callout box indicates that 'xs:boolean' is a built-in primitive type that defines boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: true
Source	<xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0" />

Element typeSDS / e2eegroup

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<p>The diagram shows a class named 'e2eegroup' with a multiplicity of 0..1. It has a directed association labeled with a minus sign (-) pointing to a class named 'ct:typeSubscriberAddress'. This class has two associations: one to 'ssi' (multiplicity 0..1) and one to 'tsi' (multiplicity 0..1). Both 'ssi' and 'tsi' have plus signs (+) next to them, indicating they are required.</p>
Type	ct:typeSubscriberAddress
Properties	content: complex minOccurs: 0
Model	ct:ssi ct:tsi
Children	ct:ssi, ct:tsi
Instance	<e2eegroup xmlns="DR-GW-Interface/DR-GW-SDS.CommonTypes" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:ssi>{1,1}</ct:ssi> <ct:tsi>{1,1}</ct:tsi> </e2eegroup>
Source	<xs:element name="e2eegroup" type="ct:typeSubscriberAddress" minOccurs="0" />

Element typeSDSValidity / value

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Diagram	<p>The diagram shows a class named 'value' with a multiplicity of 0..1. It has a directed association labeled with a minus sign (-) pointing to a class named 'xs:unsignedLong'. A callout box indicates that 'xs:unsignedLong' is a built-in derived type that is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="value" type="xs:unsignedLong" />

Complex Type(s)

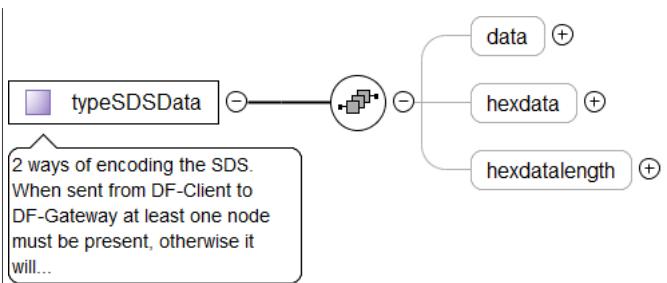
Complex Type typeSDS

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
-----------	---------------------------------------

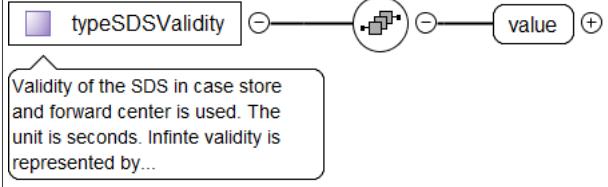
Annotations	
Diagram	<pre> classDiagram class typeSDS { protocol sdsType msgRef report sdsdata source target forward validity tstamp encryption e2eegroup } </pre>
Used by	Element SDS_Send/sds
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, tstamp, validity
Source	<pre> <xs:complexType name="typeSDS"> <xs:annotation> <xs:documentation></xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="protocol" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="sdsType" type="typeSDSType"/> <xs:element name="msgRef" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="report" type="typeReport" default="none" minOccurs="0"/> <xs:element name="sdsdata" type="typeSDSData"/> <xs:element name="source" type="ct:typeAddress" minOccurs="0"/> <xs:element name="target" type="ct:typeAddress"/> <xs:element name="forward" type="ct:typeAddress" minOccurs="0"/> <xs:element name="validity" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/> <xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/> <xs:element name="e2eegroup" type="ct:typeSubscriberAddress" minOccurs="0"/> </xs:sequence> </xs:complexType> </pre>

Complex Type typeSDSData

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Annotations	<p>2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will be discarded as not valid.</p> <p>When sent from DF-Gateway to DF-Client both nodes must be present, as it is unclear if the DF-Client supports the encoding inside raw "hexdata", so the readable decoded content must be present to.</p> <p>The default charset used within the "data" node is ISO-8859-15.</p>

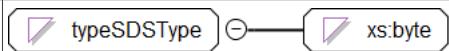
Diagram	
	2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will...
Used by	Element typeSDS/sdsdata
Model	data{0,1} , hexdata{0,1} , hexdatalength{0,1}
Children	data, hexdata, hexdatalength
Source	<pre><xs:complexType name="typeSDSData"> <xs:annotation> <xs:documentation>2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will be discarded as not valid. When sent from DF-Gateway to DF-Client both nodes must be present, as it is unclear if the DF-Client supports the encoding inside raw "hexdata", so the readable decoded content must be present to. The default charset used within the "data" node is ISO-8859-15.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="data" type="xs:string" minOccurs="0"/> <xs:element name="hexdata" type="xs:hexBinary" minOccurs="0"/> <xs:element name="hexdatalength" type="xs:integer" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeSDSValidity

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Annotations	Validity of the SDS in case store and forward center is used. The unit is seconds. Infinte validity is represented by 0xFFFFFFFF
Diagram	
	Validity of the SDS in case store and forward center is used. The unit is seconds. Infinte validity is represented by...
Model	value
Children	value
Source	<pre><xs:complexType name="typeSDSValidity"> <xs:annotation> <xs:documentation>Validity of the SDS in case store and forward center is used. The unit is seconds. Infinte validity is represented by 0xFFFFFFFF</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="value" type="xs:unsignedLong"/> </xs:sequence> </xs:complexType></pre>

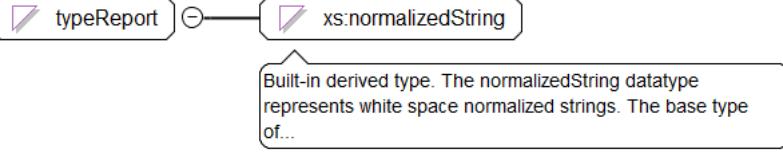
Simple Type(s)

Simple Type typeSDSType

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes
Annotations	
Diagram	
	Built-in derived type. The byte datatype is derived from short by setting the value of maxInclusive to be 127 and...
Type	restriction of xs:byte

Facets	enumeration	0	SDS1.
	enumeration	1	SDS2.
	enumeration	2	SDS3.
	enumeration	3	SDS4.
	enumeration	4	SDS-TL.
	enumeration	5	Status.
Used by	Element	typeSDS/sdsType	
Source	<pre><xs:simpleType name="typeSDSType"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:restriction base="xs:byte"> <xs:enumeration value="0"> <xs:annotation> <xs:documentation>SDS1.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="1"> <xs:annotation> <xs:documentation>SDS2.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="2"> <xs:annotation> <xs:documentation>SDS3.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="3"> <xs:annotation> <xs:documentation>SDS4.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="4"> <xs:annotation> <xs:documentation>SDS-TL.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="5"> <xs:annotation> <xs:documentation>Status.</xs:documentation> </xs:annotation> </xs:enumeration> </xs:restriction> </xs:simpleType></pre>		

Simple Type typeReport

Namespace	DR-GW-Interface/DR-GW-SDS.CommonTypes								
Annotations									
Diagram	 <p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>								
Type	restriction of xs:normalizedString								
Facets	<table border="1"> <tr> <td>enumeration</td> <td>none</td> </tr> <tr> <td>enumeration</td> <td>delivery</td> </tr> <tr> <td>enumeration</td> <td>consume</td> </tr> <tr> <td>enumeration</td> <td>both</td> </tr> </table>	enumeration	none	enumeration	delivery	enumeration	consume	enumeration	both
enumeration	none								
enumeration	delivery								
enumeration	consume								
enumeration	both								
Used by	Element typeSDS/report								
Source	<pre><xs:simpleType name="typeReport"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="none"/> <xs:enumeration value="delivery"/></pre>								

```

<xs:enumeration value="consume" />
<xs:enumeration value="both" />
</xs:restriction>
</xs:simpleType>

```

Namespace: "DR-GW-Interface/CommonTypes"

Schema(s)

Imported schema CommonTypes.xsd

Namespace	DR-GW-Interface/CommonTypes
Annotations	Version 1.1.1
Properties	attribute form default: unqualified element form default: qualified

Element(s)

Element ct:typeRequest / ct:requestId

Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows a class named 'requestId' with a multiplicity of 0..1. It has a directed association labeled 'xs:unsignedLong' pointing to a box representing the datatype. A callout box states: 'Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...'.</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="requestId" type="xs:unsignedLong"/>

Element ct:typeAddress / ct:subscriber

Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows a class named 'ct:subscriber' with a multiplicity of 0..1. It has two directed associations: one labeled 'ct:typeSubscriberAddress' with multiplicity 0..1, and another labeled 'ct:ssi ct:tsi' with multiplicity 0..1. The 'ct:typeSubscriberAddress' association points to a box labeled 'ct:typeSubscriberAddress' which contains 'ct:ssi' and 'ct:tsi'.</p>
Type	ct:typeSubscriberAddress
Properties	content: complex minOccurs: 0
Model	ct:ssi ct:tsi
Children	ct:ssi, ct:tsi
Instance	<ct:subscriber xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:ssi>{1,1}</ct:ssi> <ct:tsi>{1,1}</ct:tsi> </ct:subscriber>
Source	<xs:element name="subscriber" type="ct:typeSubscriberAddress" minOccurs="0"/>

Element ct:typeSubscriberAddress / ct:ssi

Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows a class named 'ssi' with a multiplicity of 0..1. It has a directed association labeled 'xs:unsignedLong' pointing to a box representing the datatype. A callout box states: 'Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...'.</p>

Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="ssi" type="xs:unsignedLong" />

Element ct:typeSubscriberAddress / ct:tsi

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> classDiagram class ct:typeTSI { tsi mnc mcc ssi } tsi --> mnc tsi --> mcc tsi --> ssi </pre> <p>Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).</p>
Type	ct:typeTSI
Properties	content: complex
Model	ct:mnc , ct:mcc , ct:ssi
Children	ct:mcc, ct:mnc, ct:ssi
Instance	<ct:tsi xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:mnc>{1,1}</ct:mnc> <ct:mcc>{1,1}</ct:mcc> <ct:ssi>{1,1}</ct:ssi> </ct:tsi>
Source	<xs:element name="tsi" type="ct:typeTSI" />

Element ct:typeTSI / ct:mnc

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> classDiagram class mnc { <<xs:unsignedShort>> } </pre> <p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>
Type	xs:unsignedShort
Properties	content: simple
Source	<xs:element name="mnc" type="xs:unsignedShort" />

Element ct:typeTSI / ct:mcc

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> classDiagram class mcc { <<xs:unsignedShort>> } </pre> <p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>
Type	xs:unsignedShort
Properties	content: simple
Source	<xs:element name="mcc" type="xs:unsignedShort" />

Element ct:typeTSI / ct:ssi

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	<code><xs:element name="ssi" type="xs:unsignedLong"/></code>

Element ct:typeAddress / ct:alias

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	<code><xs:element name="alias" type="xs:normalizedString" minOccurs="0"/></code>

Element ct:typeAddress / ct:msisdn

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeDialString
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	<code><xs:element name="msisdn" type="ct:typeDialString" minOccurs="0"/></code>

Element ct:typeAddress / ct:fssn

Namespace	DR-GW-Interface/CommonTypes
Annotations	Fleet specific short number
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	<code><xs:element name="fssn" type="xs:unsignedLong" minOccurs="0"></code> <code><xs:annotation></code> <code><xs:documentation>Fleet specific short number</xs:documentation></code> <code></xs:annotation></code>

<pre></xs:element></pre>

Element ct:typeAddress / ct:external

Namespace	DR-GW-Interface/CommonTypes				
Diagram	<p>The diagram shows a class named 'ct:typeExternal' with two associations. One association points to 'gatewayNumber' with multiplicity '+'. Another association points to 'number' with multiplicity '+'. A note below the diagram states: 'External number consisting of Gateway number + DialString'.</p>				
Type	ct:typeExternal				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	ct:gatewayNumber , ct:number				
Children	ct:gatewayNumber, ct:number				
Instance	<pre><ct:external xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:gatewayNumber>{1,1}</ct:gatewayNumber> <ct:number>{1,1}</ct:number> </ct:external></pre>				
Source	<pre><xs:element name="external" type="ct:typeExternal" minOccurs="0"/></pre>				

Element ct:typeExternal / ct:gatewayNumber

Namespace	DR-GW-Interface/CommonTypes		
Diagram	<p>The diagram shows an association between 'gatewayNumber' and 'xs:unsignedLong'. A note below the diagram states: 'Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...'.</p>		
Type	xs:unsignedLong		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Source	<pre><xs:element name="gatewayNumber" type="xs:unsignedLong"/></pre>		

Element ct:typeExternal / ct:number

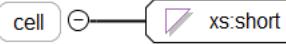
Namespace	DR-GW-Interface/CommonTypes		
Diagram	<p>The diagram shows an association between 'number' and 'ct:typeDialString'. A note below the diagram states: 'Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.'</p>		
Type	ct:typeDialString		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Facets	<table border="1"> <tr> <td>maxLength</td> <td>24</td> </tr> </table>	maxLength	24
maxLength	24		
Source	<pre><xs:element name="number" type="ct:typeDialString"/></pre>		

Element ct:typeAddress / ct:opta

Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows an association between 'opta' and 'ct:typeOPTA'. A note below the diagram states: 'OPTA string. Maximum length is 24 characters.'</p>

Type	ct:typeOPTA
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	<xs:element name="opta" type="ct:typeOPTA" minOccurs="0" />

Element ct:typeAddress / ct:cell

Namespace	DR-GW-Interface/CommonTypes
Diagram	 <p>Built-in derived type. The short datatype is derived from int by setting the value of maxInclusive to be 32767 and...</p>
Type	xs:short
Properties	content: simple minOccurs: 0
Source	<xs:element name="cell" type="xs:short" minOccurs="0" />

Element ct:typeResult / ct:resultCode

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeResponseCode
Properties	content: simple
Facets	enumeration success enumeration final_response_pending enumeration error enumeration not_authorized_error enumeration temporary_failure enumeration subscription_failed
Source	<xs:element name="resultCode" type="ct:typeResponseCode" />

Element ct:typeResult / ct:sourceSystem

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeSourceSystem
Properties	content: simple minOccurs: 0
Facets	enumeration DR-GW enumeration TCS-API enumeration TETRA
Source	<xs:element name="sourceSystem" type="ct:typeSourceSystem" minOccurs="0" />

Element ct:typeResult / ct:result

Namespace	DR-GW-Interface/CommonTypes
-----------	-----------------------------

Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	<xs:element name="result" type="xs:unsignedLong" minOccurs="0" />

Element ct:typeResponse / ct:requestId

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="requestId" type="xs:unsignedLong" />

Element ct:typeResponse / ct:result

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeResult
Properties	content: complex
Model	ct:responseCode , ct:sourceSystem {0,1} , ct:result {0,1}
Children	ct:responseCode, ct:result, ct:sourceSystem
Instance	<ct:result xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:responseCode>{1,1}</ct:responseCode> <ct:sourceSystem>{0,1}</ct:sourceSystem> <ct:result>{0,1}</ct:result> </ct:result>
Source	<xs:element name="result" type="ct:typeResult" />

Element ct:typeEvent / ct:requestId

Namespace	DR-GW-Interface/CommonTypes
Diagram	

Type	xs:unsignedLong
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="requestId" type="xs:unsignedLong" minOccurs="0" />

Element ct:typeEvent / ct:result

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> classDiagram class ct:typeResult { <<Common result values used in every response and optional specific subsystem result codes.>> } ct:typeResult "0..1" *-- "0..1" result ct:typeResult "0..1" *-- "0..1" responseCode ct:typeResult "0..1" *-- "0..1" sourceSystem </pre>
Type	ct:typeResult
Properties	<p>content: complex</p> <p>minOccurs: 0</p>
Model	ct:responseCode , ct:sourceSystem{0,1} , ct:result{0,1}
Children	ct:responseCode, ct:result, ct:sourceSystem
Instance	<ct:result xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:responseCode>{1,1}</ct:responseCode> <ct:sourceSystem>{0,1}</ct:sourceSystem> <ct:result>{0,1}</ct:result> </ct:result>
Source	<xs:element name="result" type="ct:typeResult" minOccurs="0" />

Complex Type(s)

Complex Type ct:typeRequest

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> classDiagram class typeRequest { <<Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).>> } typeRequest "0..1" *-- "0..1" requestId </pre>
Used by	Elements SDS_Send, SDS_SendReport
Model	ct:requestId
Children	ct:requestId
Source	<xs:complexType name="typeRequest"> <xs:sequence> <xs:element name="requestId" type="xs:unsignedLong" /> </xs:sequence> </xs:complexType>

Complex Type ct:typeAddress

Namespace	DR-GW-Interface/CommonTypes
Annotations	Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).

Diagram	<p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>
Used by	Elements SDS_SendReport/target, typeSDS/forward, typeSDS/source, typeSDS/target
Model	ct:subscriber{0,1}, ct:alias{0,1}, ct:msisdn{0,1}, ct:fssn{0,1}, ct:external{0,1}, ct:opta{0,1}, ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Source	<pre> <xs:complexType name="typeAddress"> <xs:annotation> <xs:documentation>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="subscriber" type="ct:typeSubscriberAddress" minOccurs="0"/> <xs:element name="alias" type="xs:normalizedString" minOccurs="0"/> <xs:element name="msisdn" type="ct:typeDialString" minOccurs="0"/> <xs:element name="fssn" type="xs:unsignedLong" minOccurs="0"> <xs:annotation> <xs:documentation>Fleet specific short number</xs:documentation> </xs:annotation> </xs:element> <xs:element name="external" type="ct:typeExternal" minOccurs="0"/> <xs:element name="opta" type="ct:typeOPTA" minOccurs="0"/> <xs:element name="cell" type="xs:short" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>

Complex Type ct:typeSubscriberAddress

Namespace	DR-GW-Interface/CommonTypes
Annotations	
Diagram	
Used by	Elements ct:typeAddress/ct:subscriber, typeSDS/e2eegroup
Model	ct:ssi ct:tsi
Children	ct:ssi, ct:tsi
Source	<pre> <xs:complexType name="typeSubscriberAddress"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:choice> <xs:element name="ssi" type="xs:unsignedLong"/> <xs:element name="tsi" type="ct:typeTSI"/> </xs:choice> </xs:complexType></pre>

Complex Type ct:typeTSI

Namespace	DR-GW-Interface/CommonTypes
Annotations	Basic type for TETRA subscriber identity containing

	Network code(MNC) and Country code(MCC).
Diagram	<p>Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).</p>
Used by	Element ct:typeSubscriberAddress/ct:tsi
Model	ct:mnc , ct:mcc , ct:ssi
Children	ct:mcc, ct:mnc, ct:ssi
Source	<pre><xs:complexType name="typeTSI"> <xs:annotation> <xs:documentation>Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="mnc" type="xs:unsignedShort"/> <xs:element name="mcc" type="xs:unsignedShort"/> <xs:element name="ssi" type="xs:unsignedLong"/> </xs:sequence> </xs:complexType></pre>

Complex Type ct:typeExternal

Namespace	DR-GW-Interface/CommonTypes
Annotations	External number consisting of Gateway number + DialString
Diagram	<p>External number consisting of Gateway number + DialString</p>
Used by	Element ct:typeAddress/ct:external
Model	ct:gatewayNumber , ct:number
Children	ct:gatewayNumber, ct:number
Source	<pre><xs:complexType name="typeExternal"> <xs:annotation> <xs:documentation>External number consisting of Gateway number + DialString</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="gatewayNumber" type="xs:unsignedLong"/> <xs:element name="number" type="ct:typeDialString"/> </xs:sequence> </xs:complexType></pre>

Complex Type ct:typeResult

Namespace	DR-GW-Interface/CommonTypes
Annotations	Common result values used in every response and optional specific subsystem result codes.
Diagram	<p>Common result values used in every response and optional specific subsystem result codes.</p>

Used by	Elements ct:typeEvent/ct:result, ct:typeResponse/ct:result
Model	ct:responseCode , ct:sourceSystem {0,1} , ct:result {0,1}
Children	ct:responseCode, ct:result, ct:sourceSystem
Source	<pre><xs:complexType name="typeResult"> <xs:annotation> <xs:documentation>Common result values used in every response and optional specific subsystem result codes.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="responseCode" type="ct:typeResponseCode" /> <xs:element name="sourceSystem" type="ct:typeSourceSystem" minOccurs="0" /> <xs:element name="result" type="xs:unsignedLong" minOccurs="0" /> </xs:sequence> </xs:complexType></pre>

Complex Type ct:typeResponse

Namespace	DR-GW-Interface/CommonTypes
Annotations	Response contains result of execution of any method.
Diagram	<pre> graph LR typeResponse["typeResponse"] -- "0..1" --> requestId["requestId"] requestId -- "+" --> result["result"] result -- "+" --> requestId </pre>
Model	ct:requestId , ct:result
Children	ct:requestId, ct:result
Source	<pre><xs:complexType name="typeResponse"> <xs:annotation> <xs:documentation>Response contains result of execution of any method.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="requestId" type="xs:unsignedLong" /> <xs:element name="result" type="ct:typeResult" /> </xs:sequence> </xs:complexType></pre>

Complex Type ct:typeEvent

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> graph LR typeEvent["typeEvent"] -- "0..1" --> requestId["requestId"] requestId -- "+" --> result["result"] result -- "+" --> requestId </pre>
Model	ct:requestId{0,1} , ct:result{0,1}
Children	ct:requestId, ct:result
Source	<pre><xs:complexType name="typeEvent"> <xs:sequence> <xs:element name="requestId" type="xs:unsignedLong" minOccurs="0" /> <xs:element name="result" type="ct:typeResult" minOccurs="0" /> </xs:sequence> </xs:complexType></pre>

Complex Type ct:typeEmpty

Namespace	DR-GW-Interface/CommonTypes
Annotations	Explicit type specification for elements that shall be empty.
Diagram	<pre> graph LR typeEmpty["typeEmpty"] </pre>
Source	<pre><xs:complexType name="typeEmpty"></pre>

```

<xs:annotation>
  <xs:documentation>Explicit type specification for elements that shall be empty.</xs:documentation>
</xs:annotation>
</xs:complexType>

```

Simple Type(s)

Simple Type ct:typeDialString

Namespace	DR-GW-Interface/CommonTypes	
Annotations	Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.	
Diagram	<pre> classDiagram typeDialString "1" -- "0..1" xs:normalizedString typeDialString "1..24" *--> xs:normalizedString </pre> <p>typeDialString is a restriction of xs:normalizedString. The restriction has a max length of 24 characters. xs:normalizedString is a built-in derived type representing white space normalized strings. The base type of typeDialString is xs:normalizedString.</p>	
Type	restriction of xs:normalizedString	
Facets	maxLength	24
Used by	Elements ct:typeAddress/ct:msisdn, ct:typeExternal/ct:number	
Source	<pre> <xs:simpleType name="typeDialString"> <xs:annotation> <xs:documentation>Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:maxLength value="24"/> </xs:restriction> </xs:simpleType> </pre>	

Simple Type ct:typeOPTA

Namespace	DR-GW-Interface/CommonTypes	
Annotations	OPTA string. Maximum length is 24 characters.	
Diagram	<pre> classDiagram typeOPTA "1" -- "0..1" xs:normalizedString typeOPTA "1..24" *--> xs:normalizedString </pre> <p>typeOPTA is a restriction of xs:normalizedString. The restriction has a max length of 24 characters. xs:normalizedString is a built-in derived type representing white space normalized strings. The base type of typeOPTA is xs:normalizedString.</p>	
Type	restriction of xs:normalizedString	
Facets	maxLength	24
Used by	Element ct:typeAddress/ct:opta	
Source	<pre> <xs:simpleType name="typeOPTA"> <xs:annotation> <xs:documentation>OPTA string. Maximum length is 24 characters.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:maxLength value="24"/> </xs:restriction> </xs:simpleType> </pre>	

Simple Type ct:typeResponseCode

Namespace	DR-GW-Interface/CommonTypes	
Diagram	<pre> classDiagram typeResponseCode "1" -- "0..1" xs:normalizedString typeResponseCode "1..24" *--> xs:normalizedString </pre> <p>typeResponseCode is a restriction of xs:normalizedString. The restriction has a max length of 24 characters. xs:normalizedString is a built-in derived type representing white space normalized strings. The base type of typeResponseCode is xs:normalizedString.</p>	

Type	restriction of xs:normalizedString	
Facets	enumeration	success
	enumeration	final_response_pending
	enumeration	error
	enumeration	not_authorized_error
	enumeration	temporary_failure
	enumeration	subscription_failed
Used by	Element	ct:typeResult/ct:responseCode
Source	<pre><xs:simpleType name="typeResponseCode"> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="success"/> <xs:enumeration value="final_response_pending"/> <xs:enumeration value="error"/> <xs:enumeration value="not_authorized_error"/> <xs:enumeration value="temporary_failure"/> <xs:enumeration value="subscription_failed"/> </xs:restriction> </xs:simpleType></pre>	

Simple Type ct:typeSourceSystem

Namespace	DR-GW-Interface/CommonTypes	
Diagram	A UML class diagram showing a class named "typeSourceSystem" with a hollow diamond symbol indicating inheritance, connected to a class named "xs:normalizedString". A callout box points to "xs:normalizedString" with the text: "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of..."	
Type	restriction of xs:normalizedString	
Facets	enumeration	DR-GW
	enumeration	TCS-API
	enumeration	TETRA
Used by	Element	ct:typeResult/ct:sourceSystem
Source	<pre><xs:simpleType name="typeSourceSystem"> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="DR-GW"/> <xs:enumeration value="TCS-API"/> <xs:enumeration value="TETRA"/> </xs:restriction> </xs:simpleType></pre>	

Simple Type ct:typeAddressingStyle

Namespace	DR-GW-Interface/CommonTypes	
Annotations	Describes the IP addressing style. Unicast or multicast.	
Diagram	A UML class diagram showing a class named "typeAddressingStyle" with a hollow diamond symbol indicating inheritance, connected to a class named "xs:normalizedString". Two callout boxes point to "xs:normalizedString": one with the text "Describes the IP addressing style. Unicast or multicast." and another with the text "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of..."	
Type	restriction of xs:normalizedString	
Facets	enumeration	ucast
	enumeration	mcast
Source	<pre><xs:simpleType name="typeAddressingStyle"> <xs:annotation> <xs:documentation>Describes the IP addressing style. Unicast or multicast.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="ucast"/> <xs:enumeration value="mcast"/> </xs:restriction> </xs:simpleType></pre>	

</xs:restriction> </xs:simpleType>
