

Schema documentation for CommonTypes.xsd

november 5, 2024

Table of Contents

Namespace: "DR-GW-Interface/CommonTypes"	1
Schema(s)	1
Main schema CommonTypes.xsd	1
Element(s)	2
Element ct:typeResult / ct:responseCode	2
Element ct:typeResult / ct:sourceSystem	2
Element ct:typeResult / ct:result	2
Element ct:typeTSI / ct:mnc	2
Element ct:typeTSI / ct:mcc	3
Element ct:typeTSI / ct:ssi	3
Element ct:typeExternal / ct:gatewayNumber	3
Element ct:typeExternal / ct:number	3
Element ct:typeSubscriberAddress / ct:ssi	4
Element ct:typeSubscriberAddress / ct:tsi	4
Element ct:typeAddress / ct:subscriber	4
Element ct:typeAddress / ct:alias	5
Element ct:typeAddress / ct:msisdn	5
Element ct:typeAddress / ct:fssn	5
Element ct:typeAddress / ct:external	6
Element ct:typeAddress / ct:opta	6
Element ct:typeAddress / ct:cell	6
Element ct:typeRequest / ct:requestId	7
Element ct:typeResponse / ct:requestId	7
Element ct:typeResponse / ct:result	7
Element ct:typeEvent / ct:requestId	7
Element ct:typeEvent / ct:result	8
Complex Type(s)	8
Complex Type ct:typeResult	8
Complex Type ct:typeTSI	9
Complex Type ct:typeExternal	9
Complex Type ct:typeSubscriberAddress	10
Complex Type ct:typeAddress	10
Complex Type ct:typeRequest	11
Complex Type ct:typeResponse	11
Complex Type ct:typeEvent	11
Complex Type ct:typeEmpty	12
Simple Type(s)	12
Simple Type ct:typeResponseCode	12
Simple Type ct:typeSourceSystem	12
Simple Type ct:typeDialString	13
Simple Type ct:typeOPTA	13
Simple Type ct:typeAddressingStyle	13

Namespace: "DR-GW-Interface/CommonTypes"

Schema(s)

Main 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:typeResult / ct:responseCode

Namespace	DR-GW-Interface/CommonTypes												
Diagram	<pre> classDiagram class responseCode class ct:typeResponseCode responseCode "1..1" o-- "1..1" ct:typeResponseCode ct:typeResponseCode < -- "1..1" Type </pre>												
Type	ct:typeResponseCode												
Properties	content: simple												
Facets	<table border="1"> <tr><td>enumeration</td><td>success</td></tr> <tr><td>enumeration</td><td>final_response_pending</td></tr> <tr><td>enumeration</td><td>error</td></tr> <tr><td>enumeration</td><td>not_authorized_error</td></tr> <tr><td>enumeration</td><td>temporary_failure</td></tr> <tr><td>enumeration</td><td>subscription_failed</td></tr> </table>	enumeration	success	enumeration	final_response_pending	enumeration	error	enumeration	not_authorized_error	enumeration	temporary_failure	enumeration	subscription_failed
enumeration	success												
enumeration	final_response_pending												
enumeration	error												
enumeration	not_authorized_error												
enumeration	temporary_failure												
enumeration	subscription_failed												
Source	<code><x:element name="responseCode" type="ct:typeResponseCode" /></code>												

Element ct:typeResult / ct:sourceSystem

Namespace	DR-GW-Interface/CommonTypes						
Diagram	<pre> classDiagram class sourceSystem class ct:typeSourceSystem sourceSystem "1..1" o-- "1..1" ct:typeSourceSystem ct:typeSourceSystem < -- "1..1" Type </pre>						
Type	ct:typeSourceSystem						
Properties	<table border="1"> <tr><td>content:</td><td>simple</td></tr> <tr><td>minOccurs:</td><td>0</td></tr> </table>	content:	simple	minOccurs:	0		
content:	simple						
minOccurs:	0						
Facets	<table border="1"> <tr><td>enumeration</td><td>DR-GW</td></tr> <tr><td>enumeration</td><td>TCS-API</td></tr> <tr><td>enumeration</td><td>TETRA</td></tr> </table>	enumeration	DR-GW	enumeration	TCS-API	enumeration	TETRA
enumeration	DR-GW						
enumeration	TCS-API						
enumeration	TETRA						
Source	<code><x:element name="sourceSystem" type="ct:typeSourceSystem" minOccurs="0" /></code>						

Element ct:typeResult / ct:result

Namespace	DR-GW-Interface/CommonTypes				
Diagram	<pre> classDiagram class result class xs:unsignedLong result "1..1" o-- "1..1" xs:unsignedLong xs:unsignedLong < -- "1..1" Type </pre> <p>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> <tr><td>minOccurs:</td><td>0</td></tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><x:element name="result" type="xs:unsignedLong" minOccurs="0" /></code>				

Element ct:typeTSI / ct:mnc

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> classDiagram class mnc class xs:unsignedShort mnc "1..1" o-- "1..1" xs:unsignedShort xs:unsignedShort < -- "1..1" Type </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	
Type	xs:unsignedShort
Properties	content: simple
Source	<xs:element name="mnc" type="xs:unsignedShort" />

Element ct:typeTSI / ct:ssi

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

Element ct:typeExternal / ct:gatewayNumber

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

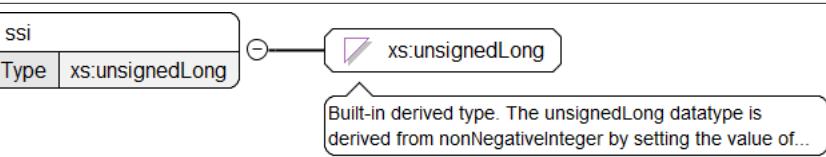
Element ct:typeExternal / ct:number

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeDialString
Properties	content: simple
Facets	maxLength 24

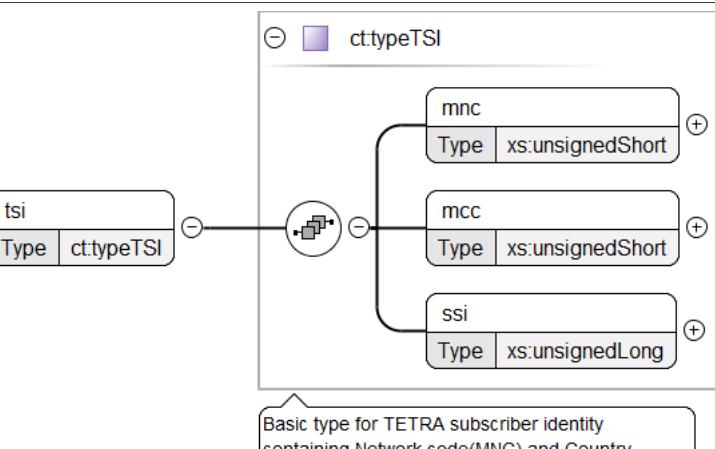
Source

```
<xss:element name="number" type="ct:typeDialString" />
```

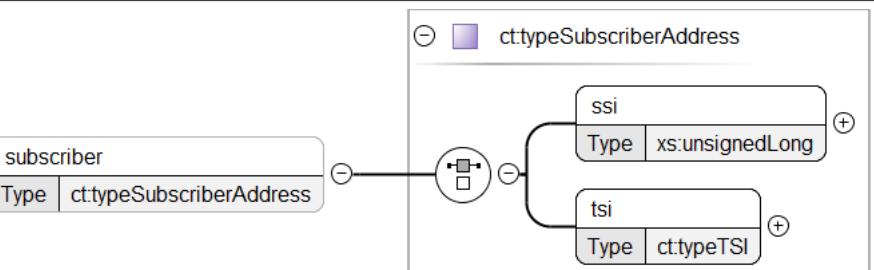
Element ct:typeSubscriberAddress / ct:ssi

Namespace	DR-GW-Interface/CommonTypes
Diagram	 <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<pre><xss:element name="ssi" type="xs:unsignedLong" /></pre>

Element ct:typeSubscriberAddress / ct:tsi

Namespace	DR-GW-Interface/CommonTypes
Diagram	 <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	<pre><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></pre>
Source	<pre><xss:element name="tsi" type="ct:typeTSI" /></pre>

Element ct:typeAddress / ct:subscriber

Namespace	DR-GW-Interface/CommonTypes
Diagram	

Type	ct:typeSubscriberAddress
Properties	content: complex minOccurs: 0
Model	ct:ssi ct:tsi
Children	ct:ssi, ct:tsi
Instance	<pre><ct:subscriber xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:ssi>{1,1}</ct:ssi> <ct:tsi>{1,1}</ct:tsi> </ct:subscriber></pre>
Source	<code><xss:element name="subscriber" type="ct:typeSubscriberAddress" minOccurs="0"/></code>

Element ct:typeAddress / ct:alias

Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows a class named 'alias' with a multiplicity of 0..1. It has a directed association labeled '-' pointing to a class named 'xs:normalizedString'. A callout box provides the following information: 'Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...'.</p>
Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	<code><xss:element name="alias" type="xs:normalizedString" minOccurs="0"/></code>

Element ct:typeAddress / ct:msisdn

Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows a class named 'msisdn' with a multiplicity of 0..1. It has a directed association labeled '-' pointing to a class named 'ct:typeDialString'. A callout box provides the following information: 'Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.'</p>
Type	ct:typeDialString
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	<code><xss: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	<p>The diagram shows a class named 'fssn' with a multiplicity of 0..1. It has a directed association labeled '-' pointing to a class named 'xs:unsignedLong'. A callout box provides the following information: 'Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...'. Another callout box labeled 'Fleet specific short number' is also present.</p>
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	<code><xss:element name="fssn" type="xs:unsignedLong" minOccurs="0"></code> <code><xss:annotation></code> <code><xss:documentation>Fleet specific short number</xss:documentation></code>

```

</xs:annotation>
</xs:element>

```

Element ct:typeAddress / ct:external

Namespace	DR-GW-Interface/CommonTypes				
Diagram	<p>Diagram illustrating the structure of the 'external' element:</p> <ul style="list-style-type: none"> The 'external' element is a type of 'ct:typeExternal'. 'ct:typeExternal' has two attributes: <ul style="list-style-type: none"> 'gatewayNumber' (Type: xs:unsignedLong) 'number' (Type: ct:typeDialString) A note below the diagram states: "External number consisting of Gateway number + DialString". 				
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:typeAddress / ct:opta

Namespace	DR-GW-Interface/CommonTypes				
Diagram	<p>Diagram illustrating the structure of the 'opta' element:</p> <ul style="list-style-type: none"> The 'opta' element is a type of 'ct:typeOPTA'. 'ct:typeOPTA' is described as an OPTA string. Maximum length is 24 characters. 				
Type	ct:typeOPTA				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	maxLength 24				
Source	<pre><xs:element name="opta" type="ct:typeOPTA" minOccurs="0"/></pre>				

Element ct:typeAddress / ct:cell

Namespace	DR-GW-Interface/CommonTypes				
Diagram	<p>Diagram illustrating the structure of the 'cell' element:</p> <ul style="list-style-type: none"> The 'cell' element is a type of 'xs:short'. 'xs:short' is described as a built-in derived type. The short datatype is derived from int by setting the value of maxInclusive to be 32767 and... 				
Type	xs:short				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre><xs:element name="cell" type="xs:short" minOccurs="0"/></pre>				

Element ct:typeRequest / ct:requestId

Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows the element 'requestId' with its type 'xs:unsignedLong'. A callout box indicates that 'xs:unsignedLong' is a built-in derived type derived from 'nonNegativeInteger'.</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="requestId" type="xs:unsignedLong"/>

Element ct:typeResponse / ct:requestId

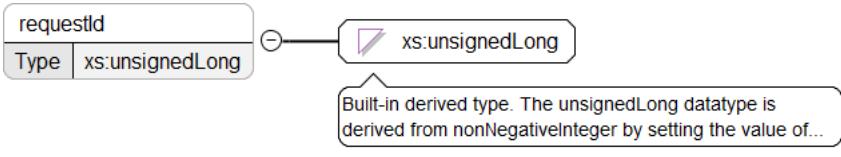
Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows the element 'requestId' with its type 'xs:unsignedLong'. A callout box indicates that 'xs:unsignedLong' is a built-in derived type derived from 'nonNegativeInteger'.</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="requestId" type="xs:unsignedLong"/>

Element ct:typeResponse / ct:result

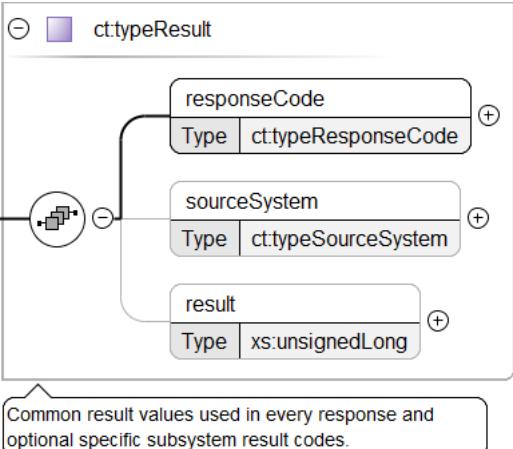
Namespace	DR-GW-Interface/CommonTypes
Diagram	<p>The diagram shows the element 'result' with its type 'ct:typeResult'. It has three children: 'responseCode', 'sourceSystem', and 'result'. A callout box indicates that 'ct:typeResult' is a common result value used in every response and optional specific subsystem result codes.</p>
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	<pre><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></pre>
Source	<xs:element name="result" type="ct:typeResult"/>

Element ct:typeEvent / ct:requestId

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

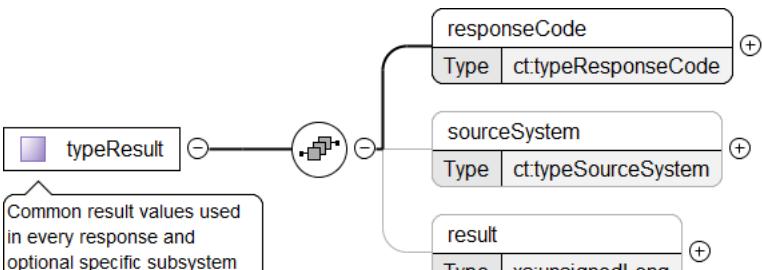
Diagram	
Type	xs:unsignedLong
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="requestId" type="xs:unsignedLong" minOccurs="0" />

Element ct:typeEvent / ct:result

Namespace	DR-GW-Interface/CommonTypes
Diagram	
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	<pre><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></pre>
Source	<xs:element name="result" type="ct:typeResult" minOccurs="0" />

Complex Type(s)

Complex Type ct:typeResult

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

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:typeTSI

Namespace	DR-GW-Interface/CommonTypes
Annotations	Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).
Diagram	<pre> classDiagram typeTSI < -- mnc : xs:unsignedShort typeTSI < -- mcc : xs:unsignedShort typeTSI < -- ssi : xs:unsignedLong note over typeTSI: Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC). </pre>
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	<pre> classDiagram typeExternal < -- gatewayNumber : xs:unsignedLong typeExternal < -- number : ct:typeDialString note over typeExternal: External number consisting of Gateway number + DialString </pre>
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></pre>

```

</xs:annotation>
<xs:sequence>
  <xs:element name="gatewayNumber" type="xs:unsignedLong" />
  <xs:element name="number" type="ct:typeDialString" />
</xs:sequence>
</xs:complexType>

```

Complex Type ct:typeSubscriberAddress

Namespace	DR-GW-Interface/CommonTypes
Annotations	
Diagram	<pre> graph LR typeSubscriberAddress --> ssi[ssi Type xs:unsignedLong] typeSubscriberAddress --> tsi[tsi Type ct:typeTSI] </pre>
Used by	Element ct:typeAddress/ct:subscriber
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:typeAddress

Namespace	DR-GW-Interface/CommonTypes
Annotations	Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).
Diagram	<pre> graph LR typeAddress --> subscriber[subscriber Type ct:typeSubscriberAddress] typeAddress --> alias[alias Type xs:normalizedString] typeAddress --> msisdn[msisdn Type ct:typeDialString] typeAddress --> fssn[fssn Type xs:unsignedLong] typeAddress --> external[external Type ct:typeExternal] typeAddress --> opta[opta Type ct:typeOPTA] typeAddress --> cell[cell Type xs:short] </pre> <p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</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

Source	<pre> <xss:complexType name="typeAddress"> <xss:annotation> <xss:documentation>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</xss:documentation> </xss:annotation> <xss:sequence> <xss:element name="subscriber" type="ct:typeSubscriberAddress" minOccurs="0"/> <xss:element name="alias" type="xs:normalizedString" minOccurs="0"/> <xss:element name="msisdn" type="ct:typeBialString" minOccurs="0"/> <xss:element name="fssn" type="xs:unsignedLong" minOccurs="0"> <xss:annotation> <xss:documentation>Fleet specific short number</xss:documentation> </xss:annotation> </xss:element> <xss:element name="external" type="ct:typeExternal" minOccurs="0"/> <xss:element name="opta" type="ct:typeOPTA" minOccurs="0"/> <xss:element name="cell" type="xs:short" minOccurs="0"/> </xss:sequence> </xss:complexType> </pre>
--------	--

Complex Type ct:typeRequest

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> classDiagram class typeRequest class requestId { <<Type xs:unsignedLong>> } typeRequest "1" -- "0..1" requestId </pre>
Model	ct:requestId
Children	ct:requestId
Source	<pre> <xss:complexType name="typeRequest"> <xss:sequence> <xss:element name="requestId" type="xs:unsignedLong"/> </xss:sequence> </xss:complexType> </pre>

Complex Type ct:typeResponse

Namespace	DR-GW-Interface/CommonTypes
Annotations	Response contains result of execution of any method.
Diagram	<pre> classDiagram class typeResponse class requestId { <<Type xs:unsignedLong>> } class result { <<Type ct:typeResult>> } typeResponse "1" -- "0..1" requestId typeResponse "1" -- "0..1" result </pre> <p>Response contains result of execution of any method.</p>
Model	ct:requestId , ct:result
Children	ct:requestId, ct:result
Source	<pre> <xss:complexType name="typeResponse"> <xss:annotation> <xss:documentation>Response contains result of execution of any method.</xss:documentation> </xss:annotation> <xss:sequence> <xss:element name="requestId" type="xs:unsignedLong"/> <xss:element name="result" type="ct:typeResult"/> </xss:sequence> </xss:complexType> </pre>

Complex Type ct:typeEvent

Namespace	DR-GW-Interface/CommonTypes
Diagram	<pre> classDiagram class typeEvent class requestId { <<Type xs:unsignedLong>> } class result { <<Type ct:typeResult>> } typeEvent "1" -- "0..1" requestId typeEvent "1" -- "0..1" result </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	<p>A UML class diagram fragment showing a class named "typeEmpty". A callout bubble points from the class to a note: "Explicit type specification for elements that shall be empty."</p>
Source	<pre><xs:complexType name="typeEmpty"> <xs:annotation> <xs:documentation>Explicit type specification for elements that shall be empty.</xs:documentation> </xs:annotation> </xs:complexType></pre>

Simple Type(s)

Simple Type ct:typeResponseCode

Namespace	DR-GW-Interface/CommonTypes												
Diagram	<p>A UML class diagram fragment showing a class named "typeResponseCode" connected via a generalization arrow to "xs:normalizedString". A callout bubble points from the connection to a note: "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>success</td></tr> <tr><td>enumeration</td><td>final_response_pending</td></tr> <tr><td>enumeration</td><td>error</td></tr> <tr><td>enumeration</td><td>not_authorized_error</td></tr> <tr><td>enumeration</td><td>temporary_failure</td></tr> <tr><td>enumeration</td><td>subscription_failed</td></tr> </table>	enumeration	success	enumeration	final_response_pending	enumeration	error	enumeration	not_authorized_error	enumeration	temporary_failure	enumeration	subscription_failed
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	<p>A UML class diagram fragment showing a class named "typeSourceSystem" connected via a generalization arrow to "xs:normalizedString". A callout bubble points from the connection to a note: "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of..."</p>

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:typeDialString

Namespace	DR-GW-Interface/CommonTypes	
Annotations	Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.	
Diagram	<p>The diagram shows two rounded rectangles connected by a line with a hollow circle. The left rectangle contains the text "typeDialString" and the right one contains "xs:normalizedString". Below the diagram are two callout boxes. The left one says "Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters." and the right one says "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</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	<p>The diagram shows two rounded rectangles connected by a line with a hollow circle. The left rectangle contains the text "typeOPTA" and the right one contains "xs:normalizedString". Below the diagram are two callout boxes. The left one says "OPTA string. Maximum length is 24 characters." and the right one says "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</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:typeAddressingStyle

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

Annotations	Describes the IP addressing style. Unicast or multicast.				
Diagram	<pre> classDiagram typeAddressingStyle < -- xs:normalizedString note over typeAddressingStyle: Describes the IP addressing style. Unicast or multicast. note over xs:normalizedString: Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of... </pre>				
Type	restriction of xs:normalizedString				
Facets	<table border="1"> <tr> <td>enumeration</td> <td>ucast</td> </tr> <tr> <td>enumeration</td> <td>mcast</td> </tr> </table>	enumeration	ucast	enumeration	mcast
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>				