Schema documentation for DR-GW-Status.Events.xsd

november 5, 2024

Table of Contents

Namespace: "DR-GW-Interface/DR-GW-Status.Events"

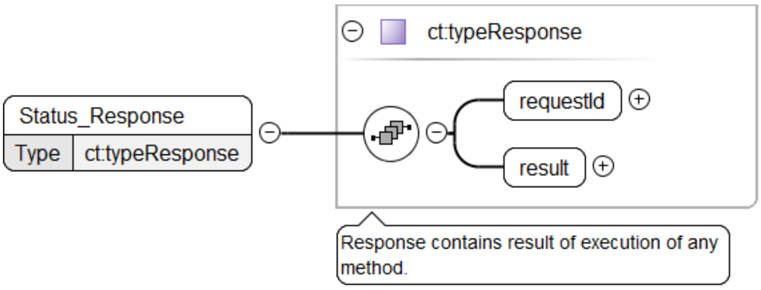
Schema(s)

Main schema `DR-GW-Status.Events.xsd`

Namespace	DR-GW-Interface/DR-GW-Status.Events
Annotations	`Version 1.1.1`
Properties	attribute form default: `unqualified` element form default: `qualified`

Element(s)

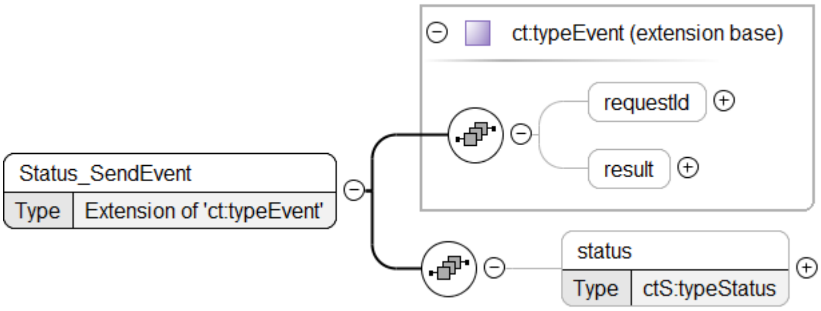
Element `Status_Response`

Namespace	DR-GW-Interface/DR-GW-Status.Events
Annotations	
Diagram	 ct:typeResponse Status_Response Type ct:typeResponse requestId result Response contains result of execution of any method.
Type	ct:typeResponse
Properties	content: `complex`
Model	ct:requestId , ct:result
Children	ct:requestId, ct:result

Instance	`<Status_Response xmlns="DR-GW-Interface/DR-GW-Status.Events" xmlns:ct="DR-GW-Interface/CommonTypes">`
	`  <ct:requestId>{1,1}</ct:requestId>`
	`  <ct:result>{1,1}</ct:result>`
	`</Status_Response>`

Source	`<xs:element name="Status_Response" type="ct:typeResponse">`
	`  <xs:annotation>`
	`    <xs:documentation/>`
	`  </xs:annotation>`
	`</xs:element>`

Element `Status_SendEvent`

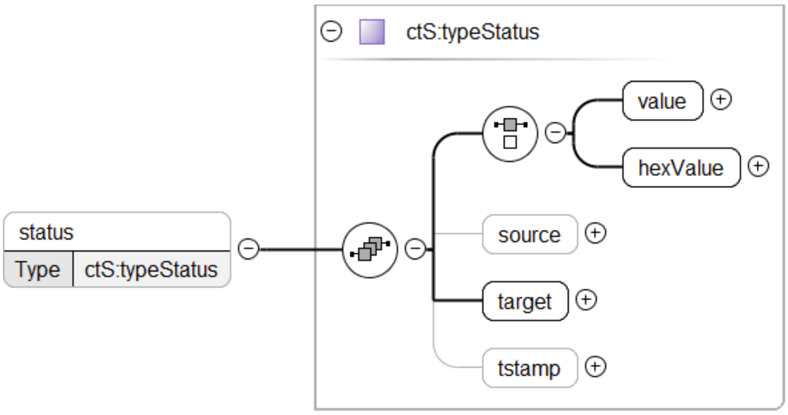
Namespace	DR-GW-Interface/DR-GW-Status.Events
Annotations	
Diagram	 ct:typeEvent (extension base) requestId result Status_SendEvent Type Extension of 'ct:typeEvent' status Type ctS:typeStatus

Type	extension of ct:typeEvent
Type hierarchy	• ct:typeEvent
Properties	content: complex
Model	ct:requestId{0,1} , ct:result{0,1} , status{0,1}
Children	ct:requestId, ct:result, status

Instance	`<Status_SendEvent xmlns="DR-GW-Interface/DR-GW-Status.Events" xmlns:ct="DR-GW-Interface/CommonTypes">`
	`<ct:requestId>{0,1}</ct:requestId>`
	`<ct:result>{0,1}</ct:result>`
	`<status>{0,1}</status>`
	`</Status_SendEvent>`

Source	`<xs:element name="Status_SendEvent">`
	`<xs:annotation>`
	`<xs:documentation/>`
	`</xs:annotation>`
	`<xs:complexType>`
	`<xs:complexContent>`
	`<xs:extension base="ct:typeEvent">`
	`<xs:sequence>`
	`<xs:element name="status" type="ctS:typeStatus" minOccurs="0"/>`
	`</xs:sequence>`
	`</xs:extension>`
	`</xs:complexContent>`
	`</xs:complexType>`
	`</xs:element>`

Element Status_SendEvent / status

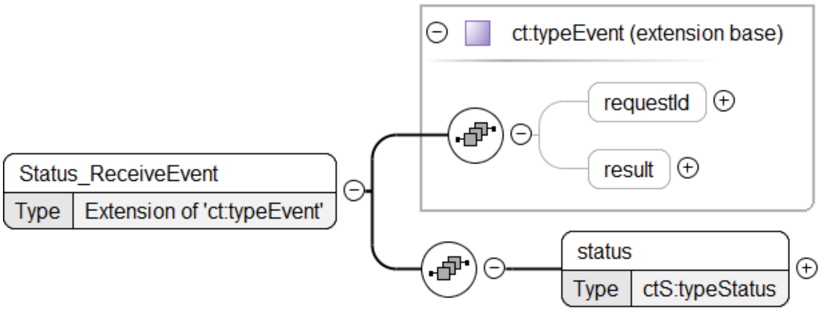
Namespace	DR-GW-Interface/DR-GW-Status.Events
Diagram	 ctS:typeStatus status Type ctS:typeStatus value hexValue source target tstamp
Type	typeStatus
Properties	content: complex minOccurs: 0
Model	(value | hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value

Instance	`<status xmlns="DR-GW-Interface/DR-GW-Status.Events" xmlns:ctS="DR-GW-Interface/DR-GW-Status.CommonTypes">`
	`<ctS:value>{1,1}</ctS:value>`
	`<ctS:hexValue>{1,1}</ctS:hexValue>`
	`<ctS:source>{0,1}</ctS:source>`
	`<ctS:target>{1,1}</ctS:target>`
	`<ctS:tstamp>{0,1}</ctS:tstamp>`
	`</status>`

Source	`<xs:element name="status" type="ctS:typeStatus" minOccurs="0"/>`

Element Status_ReceiveEvent

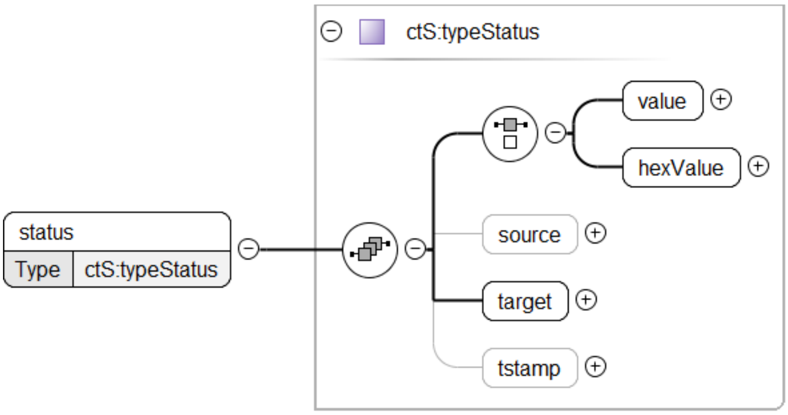
Namespace	DR-GW-Interface/DR-GW-Status.Events
Annotations	

Diagram	 ct:typeEvent (extension base) requestId result Status_ReceiveEvent Type Extension of 'ct:typeEvent' status Type ctS:typeStatus
Type	extension of ct:typeEvent
Type hierarchy	• ct:typeEvent
Properties	content: complex
Model	ct:requestId{0,1} , ct:result{0,1} , status
Children	ct:requestId, ct:result, status

Instance	`<Status_ReceiveEvent xmlns="DR-GW-Interface/DR-GW-Status.Events" xmlns:ct="DR-GW-Interface/CommonTypes">`
	`  <ct:requestId>{0,1}</ct:requestId>`
	`  <ct:result>{0,1}</ct:result>`
	`  <status>{1,1}</status>`
	`</Status_ReceiveEvent>`

Source	`<xs:element name="Status_ReceiveEvent">`
	`  <xs:annotation>`
	`    <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`    <xs:complexContent>`
	`      <xs:extension base="ct:typeEvent">`
	`        <xs:sequence>`
	`          <xs:element name="status" type="ctS:typeStatus"/>`
	`        </xs:sequence>`
	`      </xs:extension>`
	`    </xs:complexContent>`
	`  </xs:complexType>`
	`</xs:element>`

Element Status_ReceiveEvent / status

Namespace	DR-GW-Interface/DR-GW-Status.Events
Diagram	 ctS:typeStatus value hexValue source target tstamp status Type ctS:typeStatus
Type	typeStatus
Properties	content: complex
Model	(value | hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value

Instance	`<status xmlns="DR-GW-Interface/DR-GW-Status.Events" xmlns:ctS="DR-GW-Interface/DR-GW-Status.CommonTypes">`
	`  <ctS:value>{1,1}</ctS:value>`
	`  <ctS:hexValue>{1,1}</ctS:hexValue>`
	`  <ctS:source>{0,1}</ctS:source>`

	`  <ctS:target>{1,1}</ctS:target>`
	`  <ctS:tstamp>{0,1}</ctS:tstamp>`
	`</status>`

Source	`<xs:element name="status" type="ctS:typeStatus"/>`

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

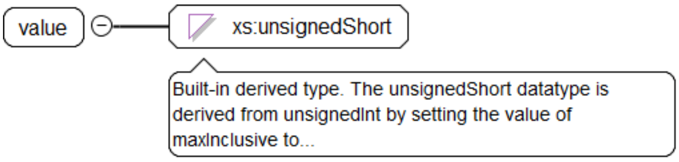
Schema(s)

Imported schema `DR-GW-Status.CommonTypes.xsd`

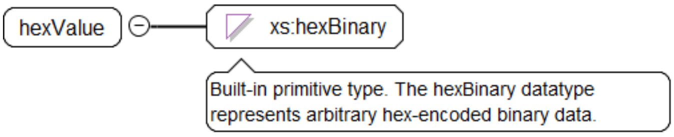
Namespace	DR-GW-Interface/DR-GW-Status.CommonTypes
Annotations	`Version 1.1.1`
Properties	attribute form default: `unqualified` element form default: `qualified`

Element(s)

Element `typeStatus / value`

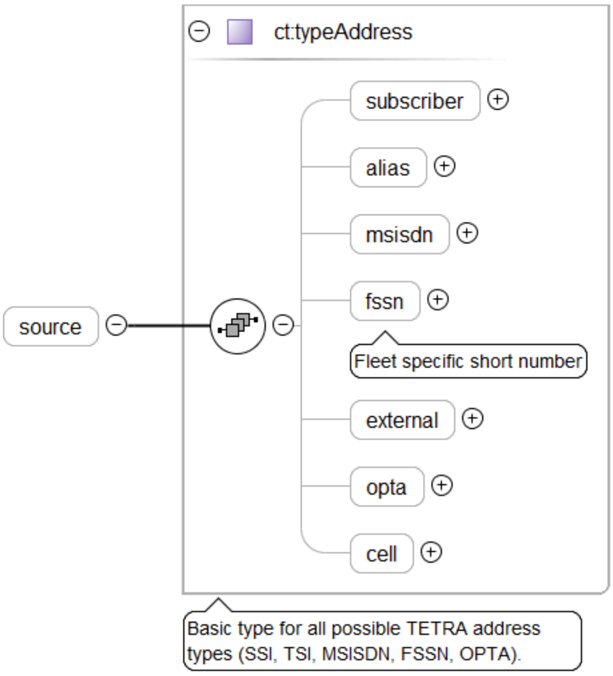
Namespace	DR-GW-Interface/DR-GW-Status.CommonTypes
Diagram	 value xs:unsignedShort Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...
Type	xs:unsignedShort
Properties	content: `simple`
Source	`<xs:element name="value" type="xs:unsignedShort"/>`

Element `typeStatus / hexValue`

Namespace	DR-GW-Interface/DR-GW-Status.CommonTypes
Diagram	 hexValue xs:hexBinary Built-in primitive type. The hexBinary datatype represents arbitrary hex-encoded binary data.
Type	xs:hexBinary
Properties	content: `simple`
Source	`<xs:element name="hexValue" type="xs:hexBinary"/>`

Element `typeStatus / source`

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

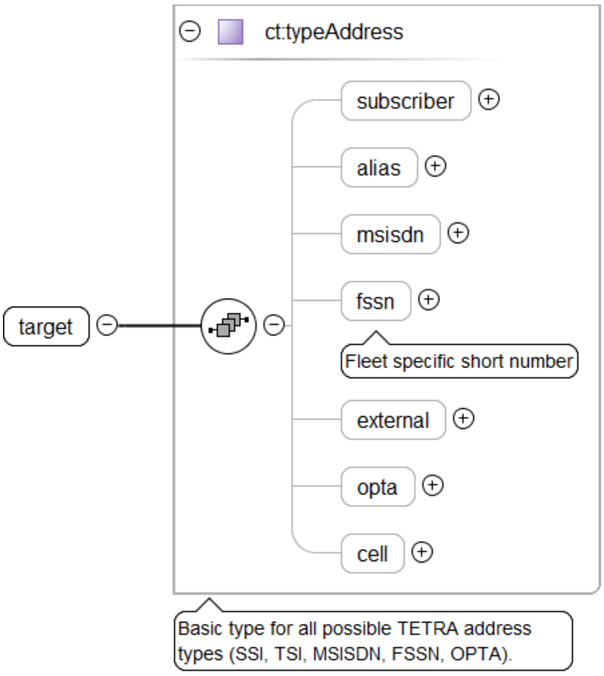
Diagram	 ct:typeAddress subscriber alias msisdn fssn Fleet specific short number external opta cell source Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).
Type	ct:typeAddress
Properties	content: `complex` minOccurs: `0`
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , 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	`<source xmlns="DR-GW-Interface/DR-GW-Status.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>`

Source	`<xs:element name="source" type="ct:typeAddress" minOccurs="0"/>`

Element `typeStatus / target`

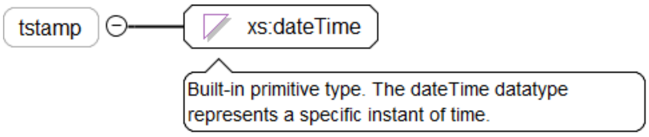
Namespace	DR-GW-Interface/DR-GW-Status.CommonTypes

Diagram	 ct:typeAddress subscriber alias msisdn fssn Fleet specific short number external opta cell target Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).
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	`<target xmlns="DR-GW-Interface/DR-GW-Status.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>`

Source	`<xs:element name="target" type="ct:typeAddress"/>`

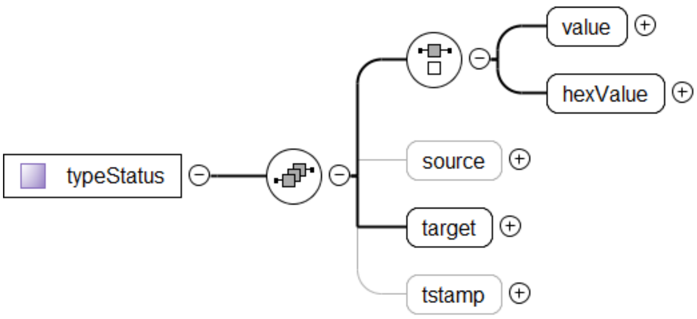
Element typeStatus / tstamp

Namespace	DR-GW-Interface/DR-GW-Status.CommonTypes
Diagram	 tstamp xs:dateTime Built-in primitive type. The dateTime datatype represents a specific instant of time.
Type	xs:dateTime
Properties	content: simple minOccurs: 0
Source	`<xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/>`

Complex Type(s)

Complex Type typeStatus

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

Diagram	 typeStatus value hexValue source target tstamp
Used by	Elements Status_ReceiveEvent/status, Status_SendEvent/status
Model	(value | hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value

Source	`<xs:complexType name="typeStatus">`
	`  <xs:sequence>`
	`    <xs:choice>`
	`      <xs:element name="value" type="xs:unsignedShort"/>`
	`      <xs:element name="hexValue" type="xs:hexBinary"/>`
	`    </xs:choice>`
	`    <xs:element name="source" type="ct:typeAddress" minOccurs="0"/>`
	`    <xs:element name="target" type="ct:typeAddress"/>`
	`    <xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Namespace: "DR-GW-Interface/CommonTypes"

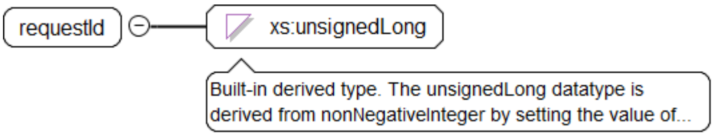
Schema(s)

Imported schema `CommonTypes.xsd`

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

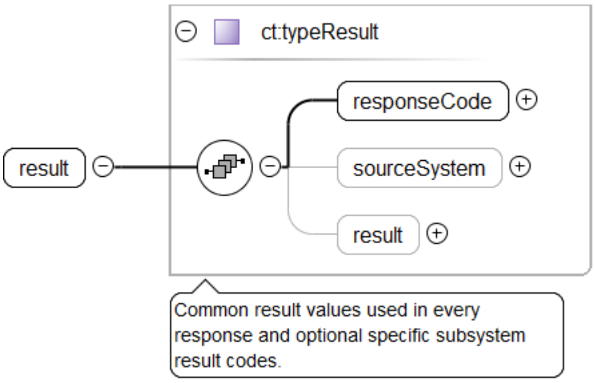
Element(s)

Element `ct:typeResponse / ct:requestId`

Namespace	DR-GW-Interface/CommonTypes
Diagram	 requestId xs:unsignedLong Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...
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	 ct:typeResult result responseCode sourceSystem result Common result values used in every response and optional specific subsystem result codes.	
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:typeResult / ct:responseCode

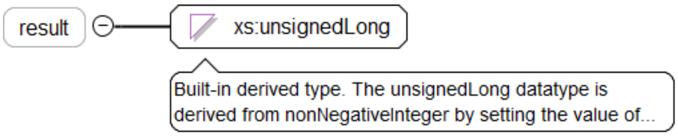
Namespace	DR-GW-Interface/CommonTypes	
Diagram		
Type	ct:typeResponseCode	
Properties	content:	simple
Facets	enumeration	success
	enumeration	final_response_pending
	enumeration	error
	enumeration	not_authorized_error
	enumeration	temporary_failure
	enumeration	subscription_failed
Source	`<xs:element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

Element ct:typeResult / ct:sourceSystem

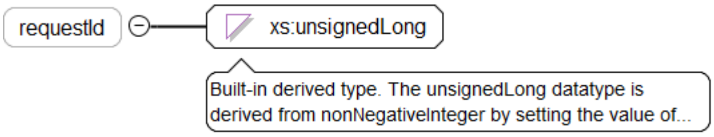
Namespace	DR-GW-Interface/CommonTypes	
Diagram		
Type	ct:typeSourceSystem	
Properties	content:	simple
	minOccurs:	0
Facets	enumeration	DR-GW
	enumeration	TCS-API
	enumeration	TETRA
Source	`<xs:element name="sourceSystem" type="ct:typeSourceSystem" minOccurs="0"/>`	

Element ct:typeResult / ct:result

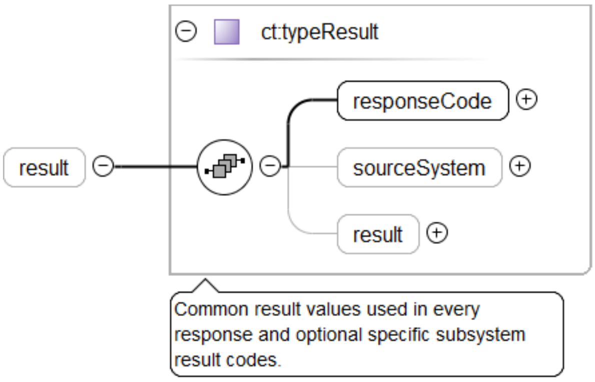
Namespace	DR-GW-Interface/CommonTypes

Diagram	 result xs:unsignedLong Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	`<xs:element name="result" type="xs:unsignedLong" minOccurs="0"/>`

Element `ct:typeEvent / ct:requestId`

Namespace	DR-GW-Interface/CommonTypes
Diagram	 requestId xs:unsignedLong Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	`<xs:element name="requestId" type="xs:unsignedLong" minOccurs="0"/>`

Element `ct:typeEvent / ct:result`

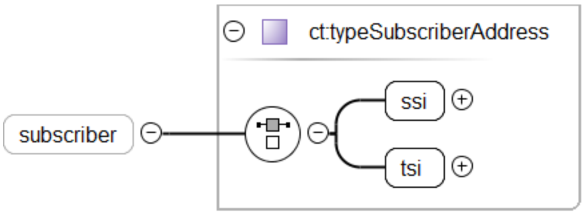
Namespace	DR-GW-Interface/CommonTypes
Diagram	 ct:typeResult responseCode sourceSystem result result Common result values used in every response and optional specific subsystem result codes.
Type	ct:typeResult
Properties	content: complex minOccurs: 0
Model	ct:responseCode , ct:sourceSystem{0,1} , ct:result{0,1}
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"/>`

Element `ct:typeAddress / ct:subscriber`

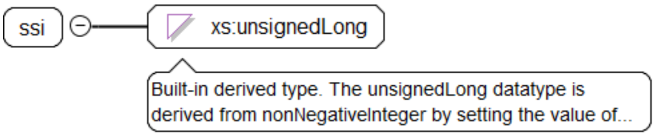
Namespace	DR-GW-Interface/CommonTypes

Diagram	
Type	ct:typeSubscriberAddress
Properties	content: complex minOccurs: 0
Model	ct:ssi | ct:tsi
Children	ct:ssi, ct:tsi

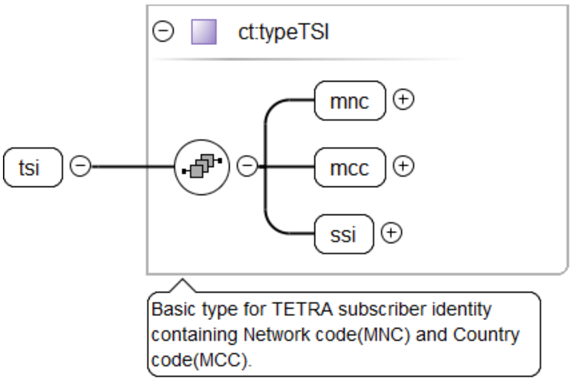
Instance	<ct:subscriber xmlns:ct="DR-GW-Interface/CommonTypes">
	  <ct:ssi>{1,1}</ct:ssi>
	  <ct:tsi>{1,1}</ct:tsi>
	</ct:subscriber>

Source	<xs:element name="subscriber" type="ct:typeSubscriberAddress" minOccurs="0"/>

Element ct:typeSubscriberAddress / ct:ssi

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

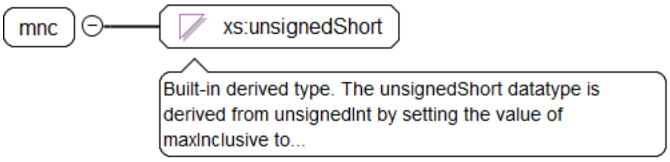
Element ct:typeSubscriberAddress / ct:tsi

Namespace	DR-GW-Interface/CommonTypes
Diagram	 ct:typeTSI tsi mnc mcc ssi Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).
Type	ct:typeTSI
Properties	content: complex
Model	ct:mnc , ct:mcc , ct:ssi
Children	ct:mcc, ct:mnc, ct:ssi

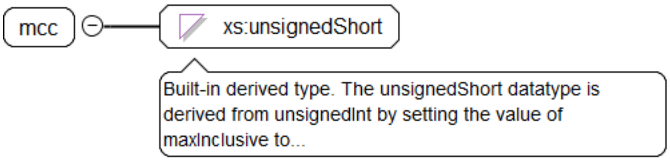
Instance	<ct:tsi xmlns:ct="DR-GW-Interface/CommonTypes">
	  <ct:mnc>{1,1}</ct:mnc>
	  <ct:mcc>{1,1}</ct:mcc>
	  <ct:ssi>{1,1}</ct:ssi>
	</ct:tsi>

Source	<xs:element name="tsi" type="ct:typeTSI"/>

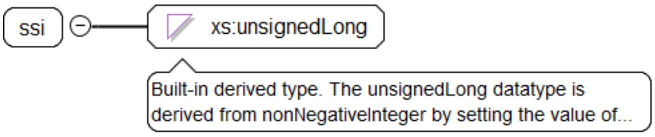
Element `ct:typeTSI / ct:mnc`

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

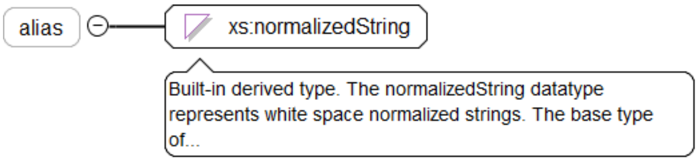
Element `ct:typeTSI / ct:mcc`

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

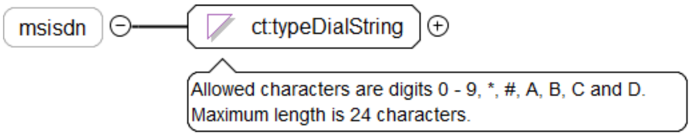
Element `ct:typeTSI / ct:ssi`

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

Element `ct:typeAddress / ct:alias`

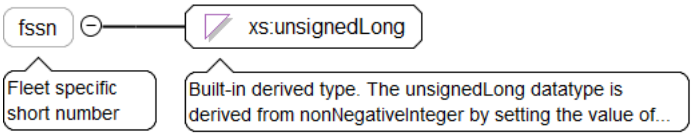
Namespace	DR-GW-Interface/CommonTypes
Diagram	 alias xs:normalizedString Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...
Type	xs:normalizedString
Properties	content: `simple` minOccurs: `0`
Source	`<xs:element name="alias" type="xs:normalizedString" minOccurs="0"/>`

Element `ct:typeAddress / ct:msisdn`

Namespace	DR-GW-Interface/CommonTypes
Diagram	 msisdn ct:typeDialString Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.

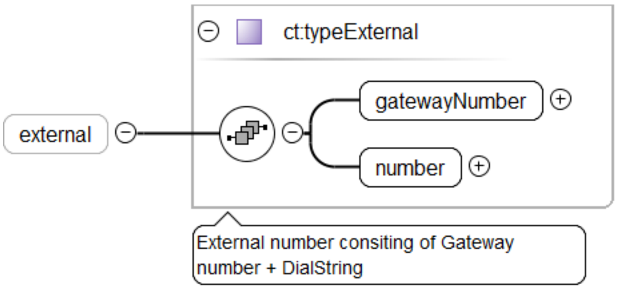
Type	ct:typeDialString
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	`<xs:element name="msisdn" type="ct:typeDialString" minOccurs="0"/>`

Element ct:typeAddress / ct:fssn

Namespace	DR-GW-Interface/CommonTypes
Annotations	Fleet specific short number
Diagram	 fssn xs:unsignedLong Fleet specific short number Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0

Source	`<xs:element name="fssn" type="xs:unsignedLong" minOccurs="0">`
	`  <xs:annotation>`
	`    <xs:documentation>Fleet specific short number</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

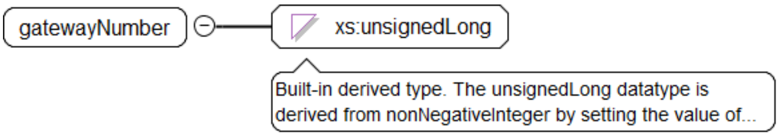
Element ct:typeAddress / ct:external

Namespace	DR-GW-Interface/CommonTypes
Diagram	 external ct:typeExternal gatewayNumber number External number consiting of Gateway number + DialString
Type	ct:typeExternal
Properties	content: complex minOccurs: 0
Model	ct:gatewayNumber , ct:number
Children	ct:gatewayNumber, ct:number

Instance	`<ct:external xmlns:ct="DR-GW-Interface/CommonTypes">`
	`  <ct:gatewayNumber>{1,1}</ct:gatewayNumber>`
	`  <ct:number>{1,1}</ct:number>`
	`</ct:external>`

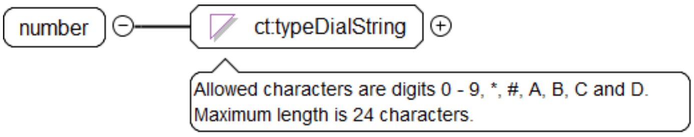
Source	`<xs:element name="external" type="ct:typeExternal" minOccurs="0"/>`

Element ct:typeExternal / ct:gatewayNumber

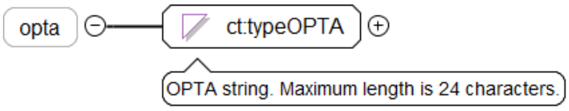
Namespace	DR-GW-Interface/CommonTypes
Diagram	 gatewayNumber xs:unsignedLong Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...
Type	xs:unsignedLong

Properties	content: simple
Source	`<xs:element name="gatewayNumber" type="xs:unsignedLong"/>`

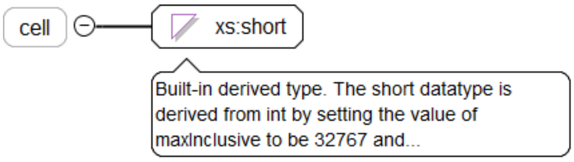
Element ct:typeExternal / ct:number

Namespace	DR-GW-Interface/CommonTypes
Diagram	 number ct:typeDialString Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.
Type	ct:typeDialString
Properties	content: simple
Facets	maxLength 24
Source	`<xs:element name="number" type="ct:typeDialString"/>`

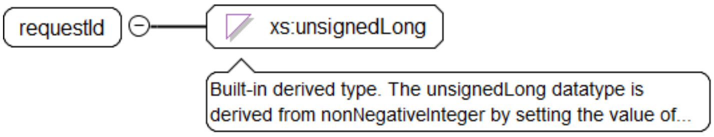
Element ct:typeAddress / ct:opta

Namespace	DR-GW-Interface/CommonTypes
Diagram	 opta ct:typeOPTA OPTA string. Maximum length is 24 characters.
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

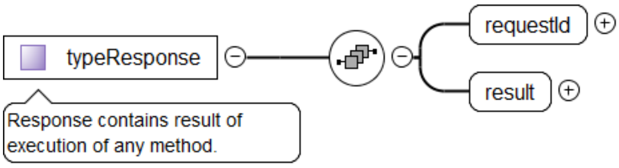
Namespace	DR-GW-Interface/CommonTypes
Diagram	 cell xs:short Built-in derived type. The short datatype is derived from int by setting the value of maxInclusive to be 32767 and...
Type	xs:short
Properties	content: simple minOccurs: 0
Source	`<xs:element name="cell" type="xs:short" minOccurs="0"/>`

Element ct:typeRequest / ct:requestId

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

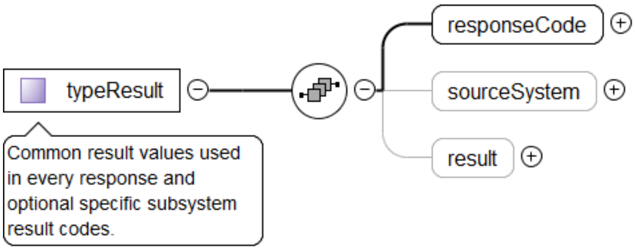
Complex Type(s)

Complex Type `ct:typeResponse`

Namespace	DR-GW-Interface/CommonTypes
Annotations	Response contains result of execution of any method.
Diagram	 typeResponse requestId result Response contains result of execution of any method.
Used by	Element Status_Response
Model	ct:requestId , ct:result
Children	ct:requestId, ct:result

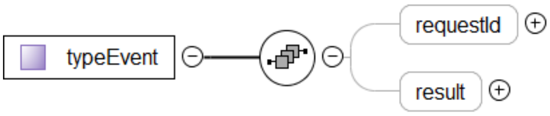
Source	<xs:complexType name="typeResponse">
	  <xs:annotation>
	    <xs:documentation>Response contains result of execution of any method.</xs:documentation>
	  </xs:annotation>
	  <xs:sequence>
	    <xs:element name="requestId" type="xs:unsignedLong"/>
	    <xs:element name="result" type="ct:typeResult"/>
	  </xs:sequence>
	</xs:complexType>

Complex Type `ct:typeResult`

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

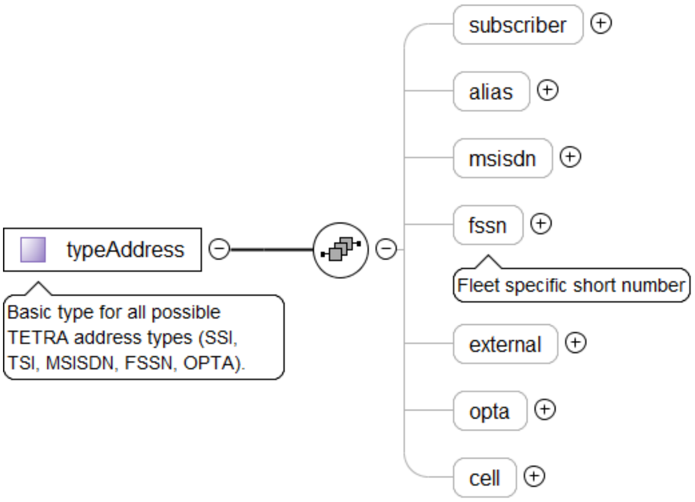
Complex Type `ct:typeEvent`

Namespace	DR-GW-Interface/CommonTypes
Diagram	

Used by	Elements Status_ReceiveEvent, Status_SendEvent
Model	ct:requestId{0,1} , ct:result{0,1}
Children	ct:requestId, ct:result

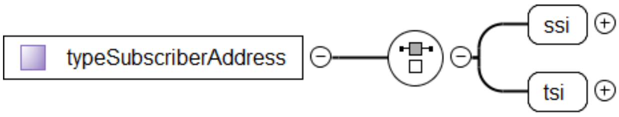
Source	`<xs:complexType name="typeEvent">`
	`  <xs:sequence>`
	`    <xs:element name="requestId" type="xs:unsignedLong" minOccurs="0"/>`
	`    <xs:element name="result" type="ct:typeResult" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `ct:typeAddress`

Namespace	DR-GW-Interface/CommonTypes
Annotations	Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).
Diagram	 subscriber alias msisdn fssn Fleet specific short number external opta cell typeAddress Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).
Used by	Elements typeStatus/source, typeStatus/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	`<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>`

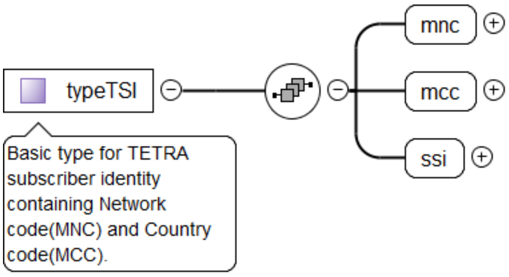
Complex Type `ct:typeSubscriberAddress`

Namespace	DR-GW-Interface/CommonTypes
Annotations	
Diagram	
Used by	Element ct:typeAddress/ct:subscriber

Model	ct:ssi | ct:tsi
Children	ct:ssi, ct:tsi

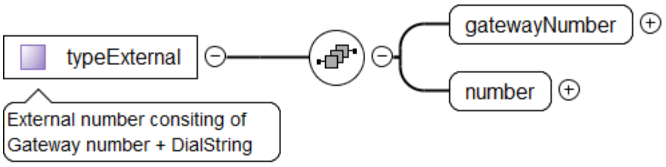
Source	<xs:complexType name="typeSubscriberAddress">
	  <xs:annotation>
	    <xs:documentation/>
	  </xs:annotation>
	  <xs:choice>
	    <xs:element name="ssi" type="xs:unsignedLong"/>
	    <xs:element name="tsi" type="ct:typeTSI"/>
	  </xs:choice>
	</xs:complexType>

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	 mnc typeTSI mcc ssi Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).	
Used by	Element	ct:typeSubscriberAddress/ct:tsi
Model	ct:mnc , ct:mcc , ct:ssi	
Children	ct:mcc, ct:mnc, ct:ssi	

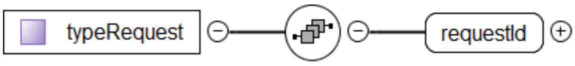
Source	<xs:complexType name="typeTSI">
	  <xs:annotation>
	    <xs:documentation>Basic type for TETRA subscriber identity containing Network code(MNC) and
	 Country code(MCC).</xs:documentation>
	  </xs:annotation>
	  <xs:sequence>
	    <xs:element name="mnc" type="xs:unsignedShort"/>
	    <xs:element name="mcc" type="xs:unsignedShort"/>
	    <xs:element name="ssi" type="xs:unsignedLong"/>
	  </xs:sequence>
	</xs:complexType>	

Complex Type `ct:typeExternal`

Namespace	DR-GW-Interface/CommonTypes	
Annotations	External number consiting of Gateway number + DialString	
Diagram	 gatewayNumber typeExternal number External number consiting of Gateway number + DialString	
Used by	Element	ct:typeAddress/ct:external
Model	ct:gatewayNumber , ct:number	
Children	ct:gatewayNumber, ct:number	

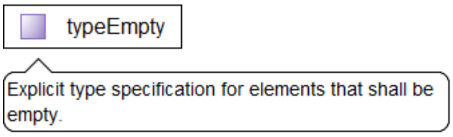
Source	<xs:complexType name="typeExternal">
	  <xs:annotation>
	    <xs:documentation>External number consiting of Gateway number + DialString</xs:documentation>
	  </xs:annotation>
	  <xs:sequence>
	    <xs:element name="gatewayNumber" type="xs:unsignedLong"/>
	    <xs:element name="number" type="ct:typeDialString"/>
	  </xs:sequence>
	</xs:complexType>	

Complex Type ct:typeRequest

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Model	ct:requestId
Children	ct:requestId

Source	`<xs:complexType name="typeRequest">`
	`  <xs:sequence>`
	`    <xs:element name="requestId" type="xs:unsignedLong"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

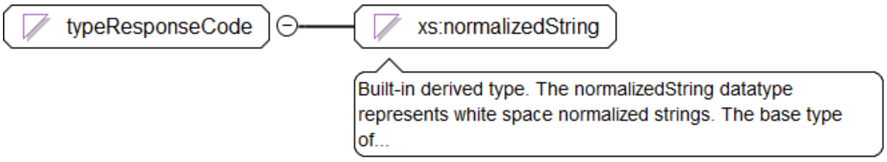
Complex Type ct:typeEmpty

Namespace	DR-GW-Interface/CommonTypes
Annotations	Explicit type specification for elements that shall be empty.
Diagram	 typeEmpty Explicit type specification for elements that shall be empty.

Source	`<xs:complexType name="typeEmpty">`
	`  <xs:annotation>`
	`    <xs:documentation>Explicit type specification for elements that shall be empty.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`</xs:complexType>`

Simple Type(s)

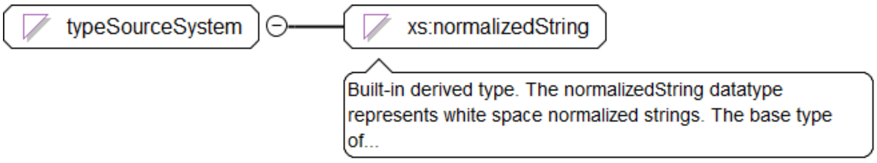
Simple Type ct:typeResponseCode

Namespace	DR-GW-Interface/CommonTypes
Diagram	 typeResponseCode xs:normalizedString Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...
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	`<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>`

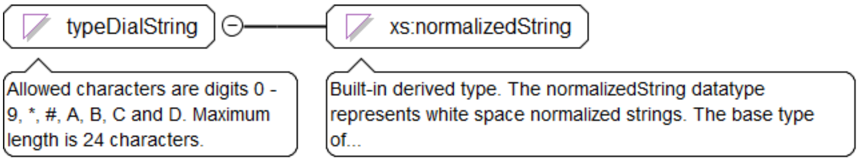
Simple Type ct:typeSourceSystem

Namespace	DR-GW-Interface/CommonTypes

Diagram	 typeSourceSystem xs:normalizedString 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

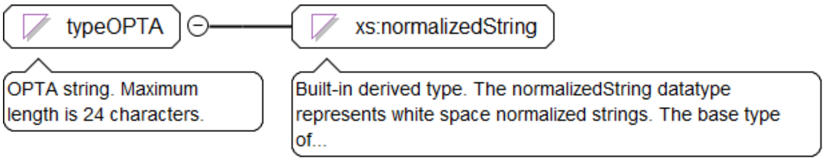
Source	`<xs:simpleType name="typeSourceSystem">`
	`  <xs:restriction base="xs:normalizedString">`
	`    <xs:enumeration value="DR-GW"/>`
	`    <xs:enumeration value="TCS-API"/>`
	`    <xs:enumeration value="TETRA"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

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	 typeDialString Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters. xs:normalizedString Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...
Type	restriction of xs:normalizedString
Facets	maxLength 24
Used by	Elements ct:typeAddress/ct:msisdn, ct:typeExternal/ct:number

Source	`<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>`

Simple Type `ct:typeOPTA`

Namespace	DR-GW-Interface/CommonTypes
Annotations	OPTA string. Maximum length is 24 characters.
Diagram	 typeOPTA OPTA string. Maximum length is 24 characters. xs:normalizedString Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...
Type	restriction of xs:normalizedString
Facets	maxLength 24
Used by	Element ct:typeAddress/ct:opta

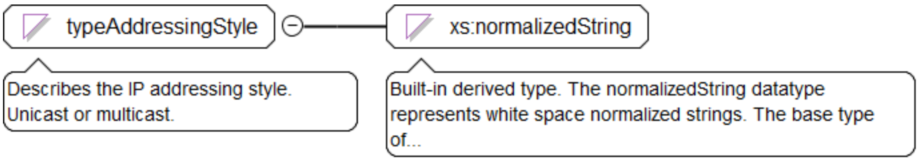
Source	`<xs:simpleType name="typeOPTA">`
	`  <xs:annotation>`
	`    <xs:documentation>OPTA string. Maximum length is 24 characters.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`

	`    <xs:maxLength value="24"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `ct:typeAddressingStyle`

Namespace	DR-GW-Interface/CommonTypes

Annotations	`Describes the IP addressing style.`
	`Unicast or multicast.`

Diagram	 typeAddressingStyle xs:normalizedString Describes the IP addressing style. Unicast or multicast. 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	`<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>`