Schema documentation for DR-GW-Session.xsd

november 5, 2024

Table of Contents

Namespace: "DR-GW-Interface/DR-GW-Session" 2
Schema(s) 2
Main schema DR-GW-Session.xsd 2
Element(s) 2
Element Session_Login 2
Element Session_Login / clientid 3
Element Session_Login / supervise 3
Element Session_Login / version 3
Element Session_Logout 3
Element Session_Supervise 4
Element Session_Check 4
Element Session_Check / clientid 5
Namespace: "DR-GW-Interface/DR-GW-Session.CommonTypes" 5
Schema(s) 5
Imported schema DR-GW-Session.CommonTypes.xsd 5
Simple Type(s) 5
Simple Type typeSuperviseTimeout 5
Namespace: "DR-GW-Interface/CommonTypes" 6
Schema(s) 6
Imported schema CommonTypes.xsd 6
Element(s) 6
Element ct:typeRequest / ct:requestId 6
Element ct:typeResult / ct:responseCode 6
Element ct:typeResult / ct:sourceSystem 7
Element ct:typeResult / ct:result 7
Element ct:typeTSI / ct:mnc 7
Element ct:typeTSI / ct:mcc 7
Element ct:typeTSI / ct:ssi 7
Element ct:typeExternal / ct:gatewayNumber 8
Element ct:typeExternal / ct:number 8
Element ct:typeSubscriberAddress / ct:ssi 8
Element ct:typeSubscriberAddress / ct:tsi 8
Element ct:typeAddress / ct:subscriber 9
Element ct:typeAddress / ct:alias 9
Element ct:typeAddress / ct:msisdn 10
Element ct:typeAddress / ct:fssn 10
Element ct:typeAddress / ct:external 10
Element ct:typeAddress / ct:opta 10
Element ct:typeAddress / ct:cell 11
Element ct:typeResponse / ct:requestId 11
Element ct:typeResponse / ct:result 11
Element ct:typeEvent / ct:requestId 12
Element ct:typeEvent / ct:result 12
Complex Type(s) 12
Complex Type ct:typeRequest 12
Complex Type ct:typeResult 13
Complex Type ct:typeTSI 13
Complex Type ct:typeExternal 13
Complex Type ct:typeSubscriberAddress 14
Complex Type ct:typeAddress 14
Complex Type ct:typeResponse 15
Complex Type ct:typeEvent 15
Complex Type ct:typeEmpty 15
Simple Type(s) 16
Simple Type ct:typeResponseCode 16
Simple Type ct:typeSourceSystem 16
Simple Type ct:typeDialString 16
Simple Type ct:typeOPTA 17
Simple Type ct:typeAddressingStyle 17

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

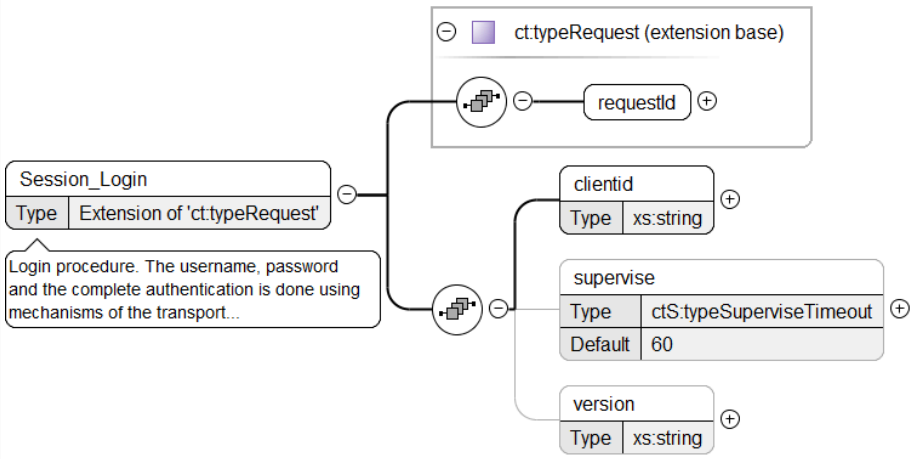
Schema(s)

Main schema `DR-GW-Session.xsd`

Namespace	DR-GW-Interface/DR-GW-Session
Annotations	`Version 1.2`
Properties	attribute form default: `unqualified` element form default: `qualified`

Element(s)

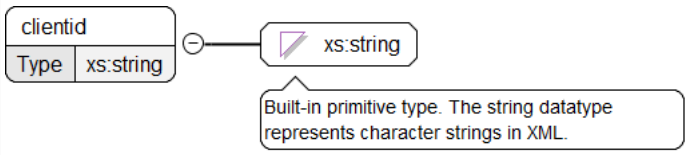
Element `Session_Login`

Namespace	DR-GW-Interface/DR-GW-Session
Annotations	`Login procedure. The username, password and the complete authentication is done using mechanisms of the transport protocol. Digest access authentication on HTTP or SIP digest authentication on SIP. Both are principialy same.`
Diagram	
Type	extension of ct:typeRequest
Type hierarchy	• ct:typeRequest
Properties	content: `complex`
Model	ct:requestId , clientid , supervise{0,1} , version{0,1}
Children	clientid, ct:requestId, supervise, version

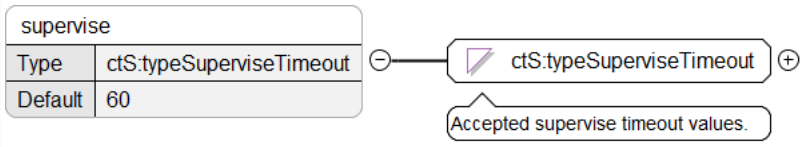
Instance	`<Session_Login xmlns="DR-GW-Interface/DR-GW-Session" xmlns:ct="DR-GW-Interface/CommonTypes">`
	`  <ct:requestId>{1,1}</ct:requestId>`
	`  <clientid>{1,1}</clientid>`
	`  <supervise>{0,1}</supervise>`
	`  <version>{0,1}</version>`
	`</Session_Login>`

Source	`<xs:element name="Session_Login">`
	`  <xs:annotation>`
	`  <xs:documentation>Login procedure. The username, password and the complete authentication is done using mechanisms of the transport protocol. Digest access authentication on HTTP or SIP digest authentication on SIP. Both are principialy same.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:complexContent>`
	`  <xs:extension base="ct:typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="clientid" type="xs:string"/>`
	`  <xs:element name="supervise" default="60" minOccurs="0" type="ctS:typeSuperviseTimeout"/>`
	`  <xs:element name="version" type="xs:string" minOccurs="0"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`  </xs:complexType>`
	`</xs:element>`

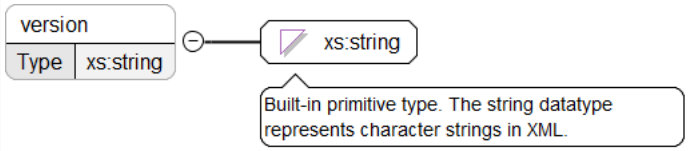
Element Session_Login / clientid

Namespace	DR-GW-Interface/DR-GW-Session
Diagram	
Type	xs:string
Properties	content: simple
Source	`<xs:element name="clientid" type="xs:string"/>`

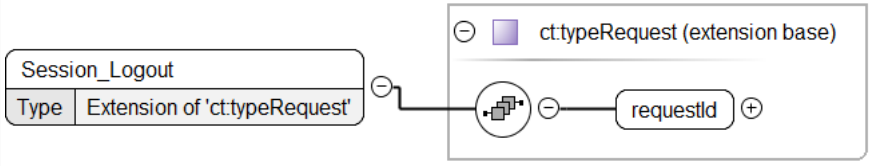
Element Session_Login / supervise

Namespace	DR-GW-Interface/DR-GW-Session
Diagram	
Type	typeSuperviseTimeout
Properties	content: simple minOccurs: 0 default: 60
Facets	enumeration 20 enumeration 30 enumeration 60
Source	`<xs:element name="supervise" default="60" minOccurs="0" type="ctS:typeSuperviseTimeout"/>`

Element Session_Login / version

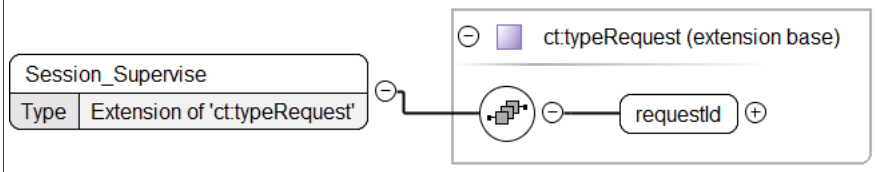
Namespace	DR-GW-Interface/DR-GW-Session
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0
Source	`<xs:element name="version" type="xs:string" minOccurs="0"/>`

Element Session_Logout

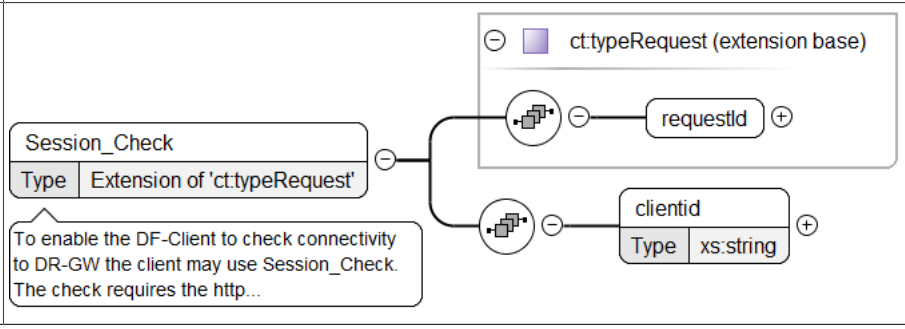
Namespace	DR-GW-Interface/DR-GW-Session
Annotations	
Diagram	
Type	extension of ct:typeRequest
Type hierarchy	• ct:typeRequest
Properties	content: complex

Model	ct:requestId
Children	ct:requestId
Instance	<Session_Logout xmlns="DR-GW-Interface/DR-GW-Session" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{1,1}</ct:requestId> </Session_Logout>
Source	<xs:element name="Session_Logout"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="ct:typeRequest"> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element>

Element Session_Supervise

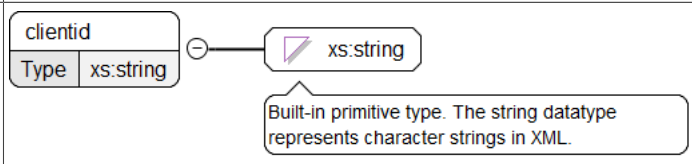
Namespace	DR-GW-Interface/DR-GW-Session
Annotations	
Diagram	
Type	extension of ct:typeRequest
Type hierarchy	• ct:typeRequest
Properties	content: complex
Model	ct:requestId
Children	ct:requestId
Instance	<Session_Supervise xmlns="DR-GW-Interface/DR-GW-Session" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{1,1}</ct:requestId> </Session_Supervise>
Source	<xs:element name="Session_Supervise"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="ct:typeRequest"> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element>

Element Session_Check

Namespace	DR-GW-Interface/DR-GW-Session
Annotations	To enable the DF-Client to check connectivity to DR-GW the client may use Session_Check. The check requires the http authentication as it would be needed for a Session_Login. Also version check could be performed. So if once the Session_Check is OK, the client should be able to login later on. There is no resource allocation associated with the session check
Diagram	

Type	extension of ct:typeRequest
Type hierarchy	• ct:typeRequest
Properties	content: complex
Model	ct:requestId , clientid
Children	clientid, ct:requestId
Instance	<Session_Check xmlns="DR-GW-Interface/DR-GW-Session" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{1,1}</ct:requestId> <clientid>{1,1}</clientid> </Session_Check>
Source	<xs:element name="Session_Check"> <xs:annotation> <xs:documentation>To enable the DF-Client to check connectivity to DR-GW the client may use Session_Check. The check requires the http authentication as it would be needed for a Session_Login. Also version check could be performed. So if once the Session_Check is OK, the client should be able to login later on. There is no resource allocation associated with the session check</xs:documentation> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="ct:typeRequest"> <xs:sequence> <xs:element name="clientid" type="xs:string"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element>

Element Session_Check / clientid

Namespace	DR-GW-Interface/DR-GW-Session
Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="clientid" type="xs:string"/>

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

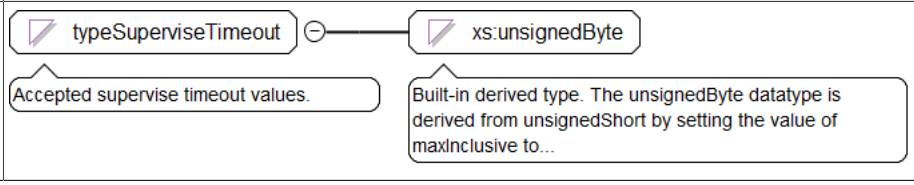
Schema(s)

Imported schema DR-GW-Session.CommonTypes.xsd

Namespace	DR-GW-Interface/DR-GW-Session.CommonTypes
Annotations	Version 1.2
Properties	attribute form default: unqualified element form default: qualified

Simple Type(s)

Simple Type typeSuperviseTimeout

Namespace	DR-GW-Interface/DR-GW-Session.CommonTypes
Annotations	Accepted supervise timeout values.
Diagram	

Type	restriction of xs:unsignedByte	
Facets	enumeration	20
	enumeration	30
	enumeration	60
Used by	Element	Session_Login/supervise

Source

```
<xs:simpleType name="typeSuperviseTimeout">
  <xs:annotation>
    <xs:documentation>Accepted supervise timeout values.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:unsignedByte">
    <xs:enumeration value="20"/>
    <xs:enumeration value="30"/>
    <xs:enumeration value="60"/>
  </xs:restriction>
</xs:simpleType>
```

Namespace: "DR-GW-Interface/CommonTypes"

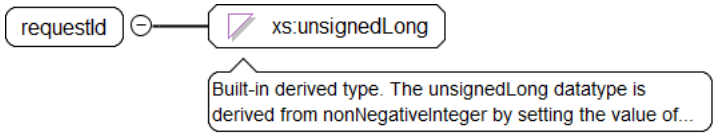
Schema(s)

Imported schema `CommonTypes.xsd`

Namespace	DR-GW-Interface/CommonTypes	
Annotations	Version 1.2	
Properties	attribute form default:	unqualified
	element form default:	qualified

Element(s)

Element `ct:typeRequest / ct:requestId`

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

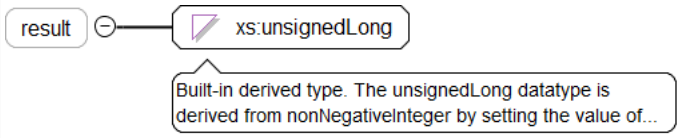
Element `ct:typeResult / ct:responseCode`

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"/>`	

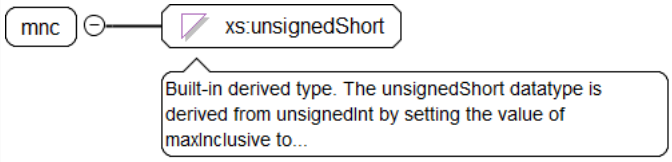
Element ct:typeResult / ct:sourceSystem

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

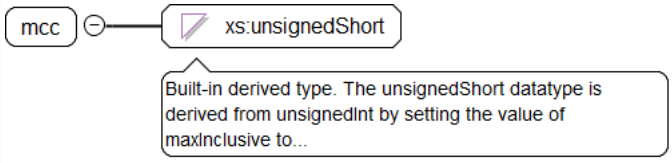
Element ct:typeResult / ct:result

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:typeTSI / ct:mnc

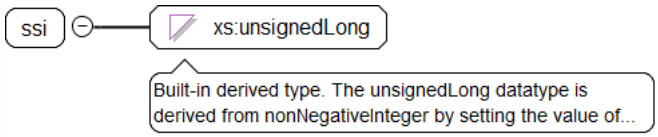
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedShort
Properties	content: simple
Source	`<xs:element name="mnc" type="xs:unsignedShort"/>`

Element ct:typeTSI / ct:mcc

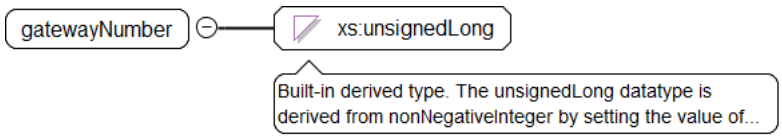
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedShort
Properties	content: simple
Source	`<xs:element name="mcc" type="xs:unsignedShort"/>`

Element ct:typeTSI / ct:ssi

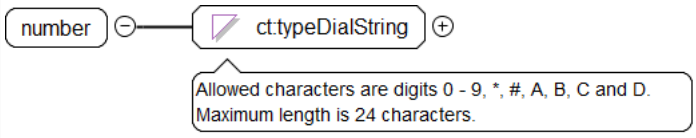
Namespace	DR-GW-Interface/CommonTypes

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

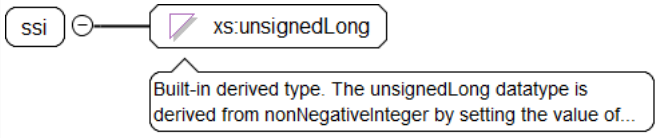
Element ct:typeExternal / ct:gatewayNumber

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

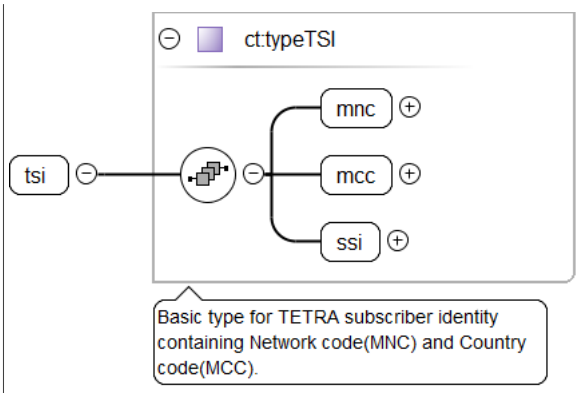
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeDialString
Properties	content: simple
Facets	maxLength 24
Source	`<xs:element name="number" type="ct:typeDialString"/>`

Element ct:typeSubscriberAddress / ct:ssi

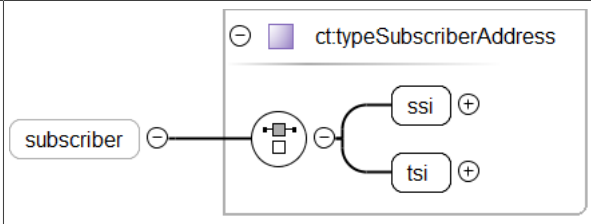
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	`<xs:element name="ssi" type="xs:unsignedLong"/>`

Element ct:typeSubscriberAddress / ct:tsi

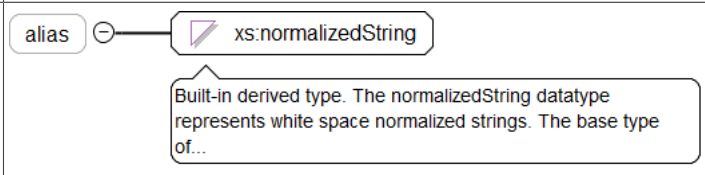
Namespace	DR-GW-Interface/CommonTypes

Diagram	
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"/>

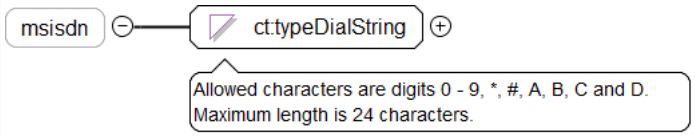
Element ct:typeAddress / ct:subscriber

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"/>

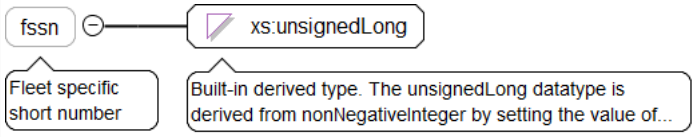
Element ct:typeAddress / ct:alias

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

Element ct:typeAddress / ct:msisdn

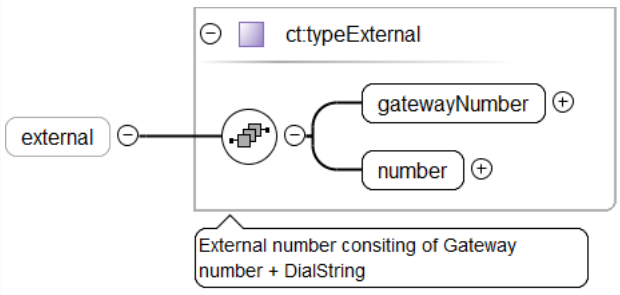
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeDialString
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	`<xs:element name="msisdn" type="ct:typeDialString" minOccurs="0"/>`

Element ct:typeAddress / ct:fssn

Namespace	DR-GW-Interface/CommonTypes
Annotations	Fleet specific short number
Diagram	
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

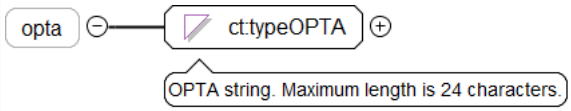
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeExternal
Properties	content: complex minOccurs: 0
Model	ct:gatewayNumber , ct:number
Children	ct:gatewayNumber, ct:number

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

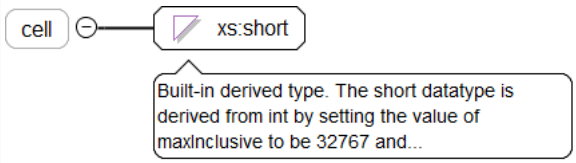
Source	`<xs:element name="external" type="ct:typeExternal" minOccurs="0"/>`

Element ct:typeAddress / ct:opta

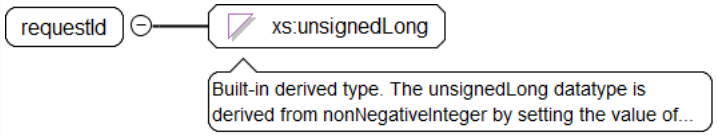
Namespace	DR-GW-Interface/CommonTypes

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

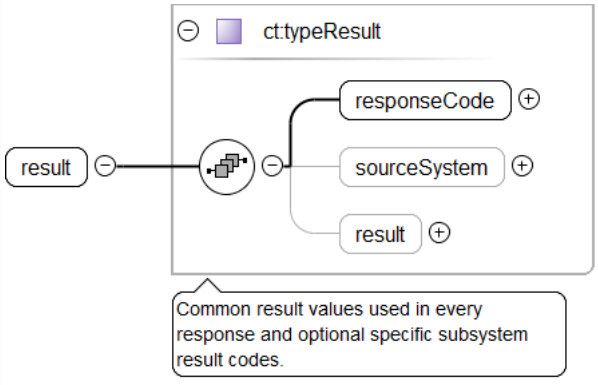
Element ct:typeAddress / ct:cell

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

Element ct:typeResponse / ct:requestId

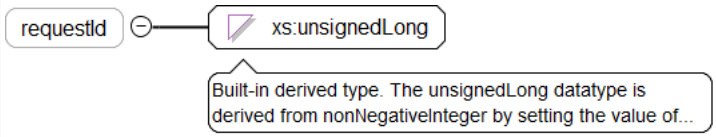
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	`<xs:element name="requestId" type="xs:unsignedLong"/>`

Element ct:typeResponse / ct:result

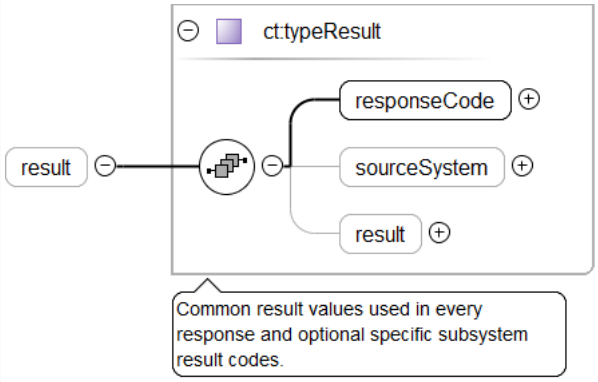
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeResult
Properties	content: complex
Model	ct:responseCode , ct:sourceSystem{0,1} , ct:result{0,1}
Children	ct:responseCode, ct:result, ct:sourceSystem
Instance	`<ct:result xmlns:ct="DR-GW-Interface/CommonTypes">`

	<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

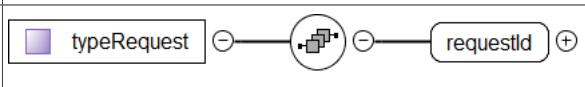
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	<xs:element name="requestId" type="xs:unsignedLong" minOccurs="0"/>

Element ct:typeEvent / ct:result

Namespace	DR-GW-Interface/CommonTypes
Diagram	
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"/>

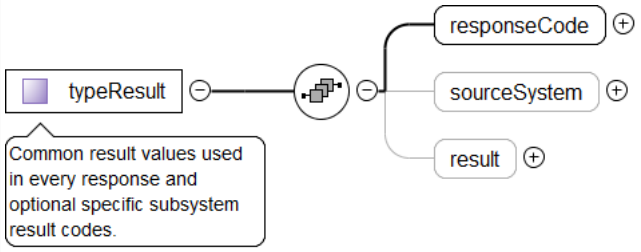
Complex Type(s)

Complex Type ct:typeRequest

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Used by	Elements Session_Check, Session_Login, Session_Logout, Session_Supervise
Model	ct:requestId
Children	ct:requestId
Source	<xs:complexType name="typeRequest"> <xs:sequence> <xs:element name="requestId" type="xs:unsignedLong"/>

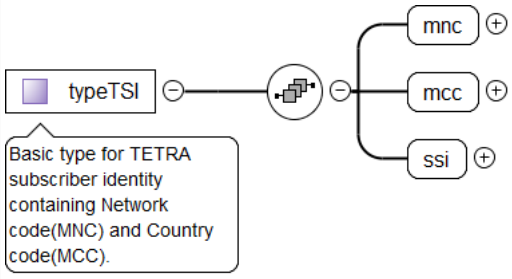
```
  </xs:sequence>
</xs:complexType>
```

Complex Type `ct: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	`<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>`

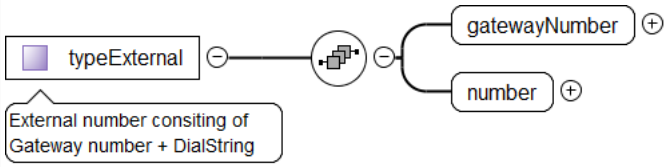
Complex Type `ct:typeTSI`

Namespace	DR-GW-Interface/CommonTypes
Annotations	`Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).`
Diagram	
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>`

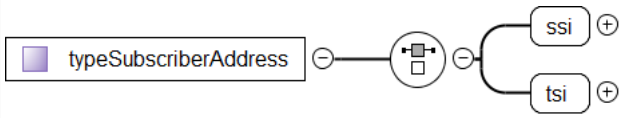
Complex Type `ct:typeExternal`

Namespace	DR-GW-Interface/CommonTypes
Annotations	`External number consiting of Gateway number + DialString`

Diagram	
Used by	Element ct:typeAddress/ct:external
Model	ct:gatewayNumber , ct:number
Children	ct:gatewayNumber, ct:number

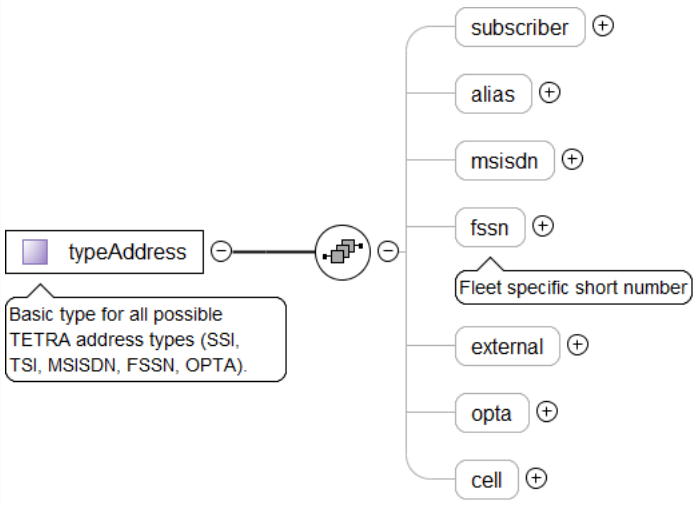
Source	`<xs:complexType name="typeExternal">`
	`  <xs:annotation>`
	`  <xs:documentation>External number consiting of Gateway number + DialString</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="gatewayNumber" type="xs:unsignedLong"/>`
	`  <xs:element name="number" type="ct:typeDialString"/>`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

Complex Type `ct: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

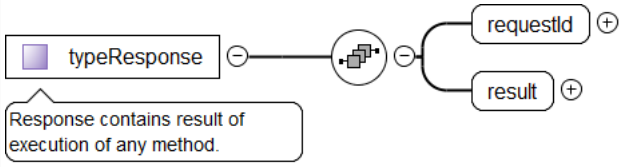
Source	`<xs:complexType name="typeSubscriberAddress">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:choice>`
	`  <xs:element name="ssi" type="xs:unsignedLong"/>`
	`  <xs:element name="tsi" type="ct:typeTSI"/>`
	`  </xs:choice>`
	`</xs:complexType>`

Complex Type `ct:typeAddress`

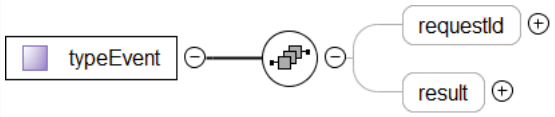
Namespace	DR-GW-Interface/CommonTypes
Annotations	`Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).`
Diagram	
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}

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>

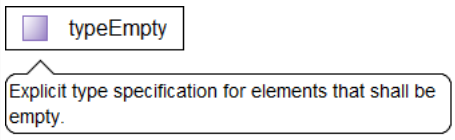
Complex Type ct:typeResponse

Namespace	DR-GW-Interface/CommonTypes
Annotations	Response contains result of execution of any method.
Diagram	
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:typeEvent

Namespace	DR-GW-Interface/CommonTypes
Diagram	
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>

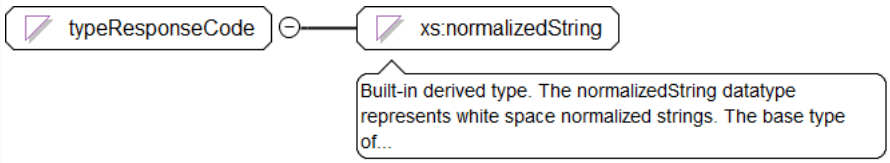
Complex Type ct:typeEmpty

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

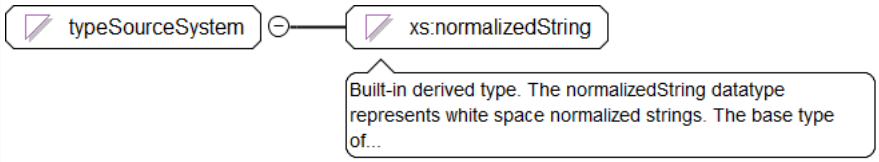
Source	<xs:complexType name="typeEmpty"> <xs:annotation> <xs:documentation>Explicit type specification for elements that shall be empty.</ xs:documentation> </xs:annotation> </xs:complexType>

Simple Type(s)

Simple Type `ct:typeResponseCode`

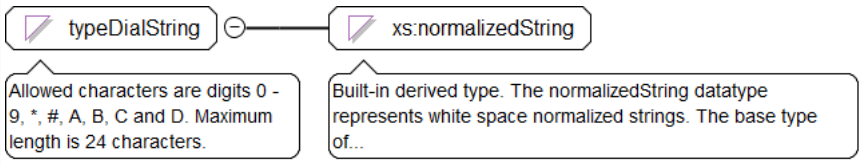
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `success` enumeration `final_response_pending` enumeration `error` enumeration `not_authorized_error` enumeration `temporary_failure` enumeration `subscription_failed`
Used by	Element ct:typeResult/ct:responseCode
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`

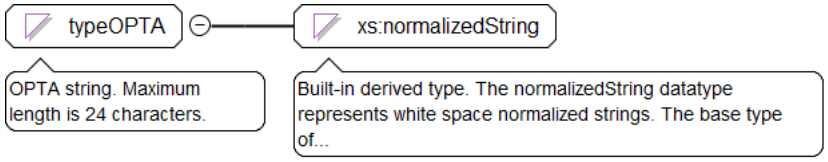
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `DR-GW` enumeration `TCS-API` enumeration `TETRA` enumeration `TACTILON-API`
Used by	Element ct:typeResult/ct:sourceSystem
Source	<xs:simpleType name="typeSourceSystem"> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="DR-GW"/> <xs:enumeration value="TCS-API"/> <xs:enumeration value="TETRA"/> <xs:enumeration value="TACTILON-API"/> </xs:restriction> </xs:simpleType>

Simple Type `ct:typeDialString`

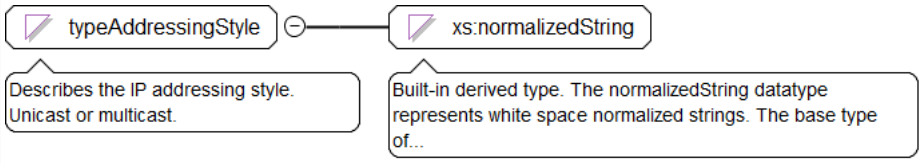
Namespace	DR-GW-Interface/CommonTypes

Annotations	Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.
Diagram	
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	
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	
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>
```