Schema documentation for openSDX_00-00-00-01.xsd

august 16, 2011

Table of Contents

Namespace: ""

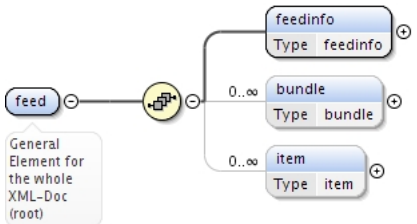
Schema(s)

Main schema openSDX_00-00-00-01.xsd

Namespace	No namespace
Properties	attribute form default: unqualified element form default: unqualified

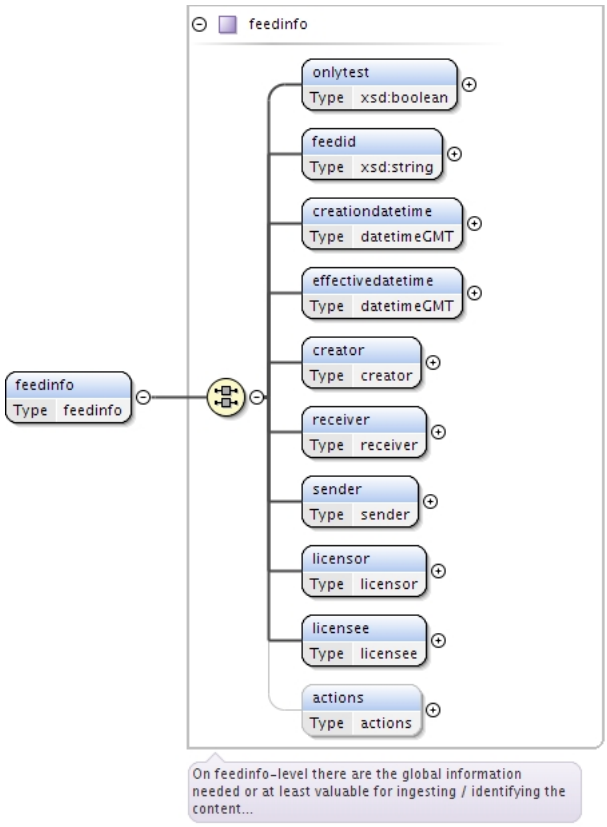
Element(s)

Element feed

Namespace	No namespace
Annotations	General Element for the whole XML-Doc (root)
Diagram	
Properties	content: complex

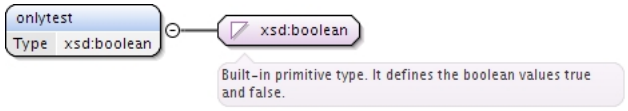
Model	feedinfo , bundle* , item*
Children	bundle, feedinfo, item
Instance	<feed> <feedinfo>{1,1}</feedinfo> <bundle>{0,unbounded}</bundle> <item>{0,unbounded}</item> </feed>
Source	<xsd:element name="feed"> <xsd:annotation> <xsd:documentation xml:lang="en">General Element for the whole XML-Doc (root)</xsd:documentation> </xsd:annotation> <xsd:complexType> <xsd:sequence> <xsd:element name="feedinfo" type="feedinfo"/> <xsd:element name="bundle" type="bundle" maxOccurs="unbounded" minOccurs="0"/> <xsd:element name="item" type="item" maxOccurs="unbounded" minOccurs="0"/> </xsd:sequence> </xsd:complexType> </xsd:element>

Element feed / feedinfo

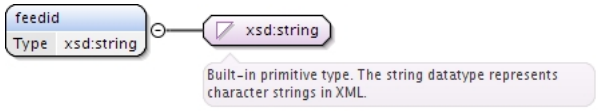
Namespace	No namespace
Diagram	
Type	feedinfo
Properties	content: complex
Model	ALL(onlytest feedid creationdatetime effectivedatetime creator receiver sender licensor licensee actions{0,1})
Children	actions, creationdatetime, creator, effectivedatetime, feedid, licensee, licensor, onlytest, receiver, sender
Instance	<feedinfo> <onlytest>{1,1}</onlytest> <feedid>{1,1}</feedid> <creationdatetime>{1,1}</creationdatetime> <effectivedatetime>{1,1}</effectivedatetime> <creator>{1,1}</creator> <receiver>{1,1}</receiver> <sender>{1,1}</sender> <licensor>{1,1}</licensor> <licensee>{1,1}</licensee>

	<actions>{0,1}</actions> </feedinfo>
Source	<xsd:element name="feedinfo" type="feedinfo"/>

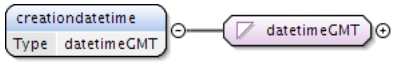
Element feedinfo / onlytest

Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple
Source	<xsd:element name="onlytest" type="xsd:boolean"/>

Element feedinfo / feedid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="feedid" type="xsd:string"/>

Element feedinfo / creationdatetime

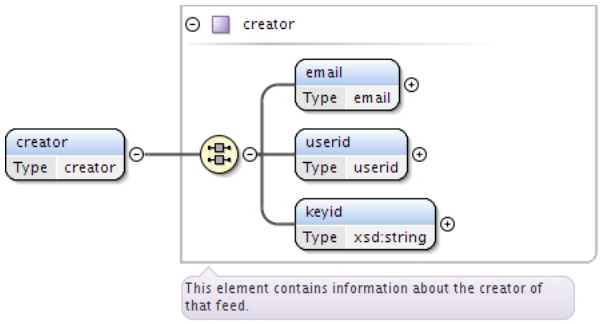
Namespace	No namespace
Diagram	
Type	datetimeGMT
Properties	content: simple
Facets	pattern \d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}
Source	<xsd:element name="creationdatetime" type="datetimeGMT"/>

Element feedinfo / effectivedatetime

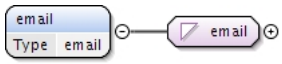
Namespace	No namespace
Diagram	
Type	datetimeGMT
Properties	content: simple
Facets	pattern \d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}
Source	<xsd:element name="effectivedatetime" type="datetimeGMT"/>

Element feedinfo / creator

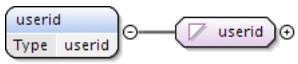
Namespace	No namespace

Diagram	
Type	creator
Properties	content: complex
Model	ALL(email userid keyid)
Children	email, keyid, userid
Instance	<creator> <email>{1,1}</email> <userid>{1,1}</userid> <keyid>{1,1}</keyid> </creator>
Source	<xsd:element name="creator" type="creator"/>

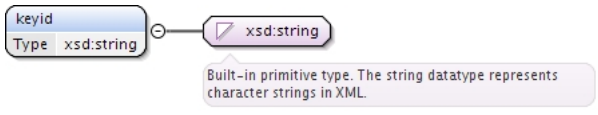
Element creator / email

Namespace	No namespace
Diagram	
Type	email
Properties	content: simple
Source	<xsd:element name="email" type="email"/>

Element creator / userid

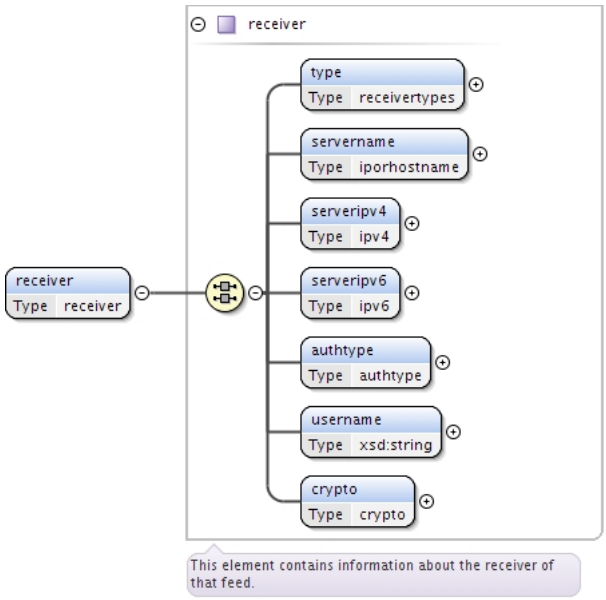
Namespace	No namespace
Diagram	
Type	userid
Properties	content: simple
Source	<xsd:element name="userid" type="userid"/>

Element creator / keyid

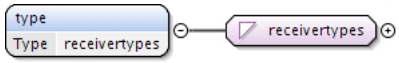
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="keyid" type="xsd:string"/>

Element feedinfo / receiver

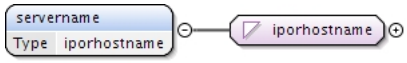
Namespace	No namespace

Diagram	
Type	receiver
Properties	content: complex
Model	ALL(type servername serveripv4 serveripv6 authtype username crypto)
Children	authtype, crypto, serveripv4, serveripv6, servername, type, username
Instance	<receiver> <type>{1,1}</type> <servername>{1,1}</servername> <serveripv4>{1,1}</serveripv4> <serveripv6>{1,1}</serveripv6> <authtype>{1,1}</authtype> <username>{1,1}</username> <crypto>{1,1}</crypto> </receiver>
Source	<xsd:element name="receiver" type="receiver"/>

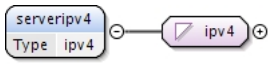
Element receiver / type

Namespace	No namespace
Diagram	
Type	receivertypes
Properties	content: simple
Facets	enumeration ftp enumeration ftps enumeration sftp enumeration openSDX-Beam
Source	<xsd:element name="type" type="receivertypes"/>

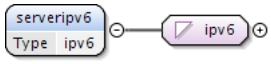
Element receiver / servername

Namespace	No namespace
Diagram	
Type	iporhostname
Properties	content: simple
Source	<xsd:element name="servername" type="iporhostname"/>

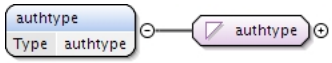
Element `receiver / serveripv4`

Namespace	No namespace
Diagram	
Type	ipv4
Properties	content: simple
Facets	pattern \d{3}\.\d{3}\.\d{3}\.\d{3}
Source	`<xsd:element name="serveripv4" type="ipv4"/>`

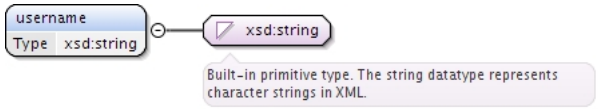
Element `receiver / serveripv6`

Namespace	No namespace
Diagram	
Type	ipv6
Properties	content: simple
Source	`<xsd:element name="serveripv6" type="ipv6"/>`

Element `receiver / authtype`

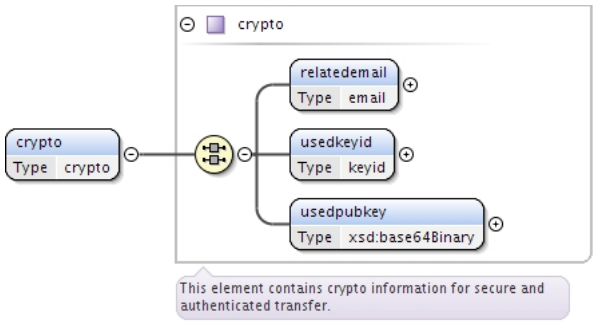
Namespace	No namespace
Diagram	
Type	authtype
Properties	content: simple
Facets	enumeration login enumeration keyfile enumeration kerberos enumeration keyfile+login enumeration keyfile+username
Source	`<xsd:element name="authtype" type="authtype"/>`

Element `receiver / username`

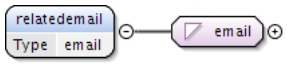
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="username" type="xsd:string"/>`

Element `receiver / crypto`

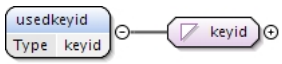
Namespace	No namespace

Diagram	
Type	crypto
Properties	content: complex
Model	ALL(relatedemail usedkeyid usedpubkey)
Children	relatedemail, usedkeyid, usedpubkey
Instance	<crypto> <relatedemail>{1,1}</relatedemail> <usedkeyid>{1,1}</usedkeyid> <usedpubkey>{1,1}</usedpubkey> </crypto>
Source	<xsd:element name="crypto" type="crypto"/>

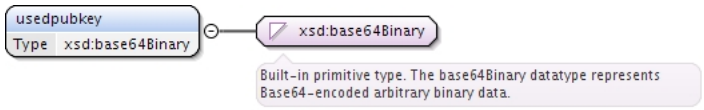
Element crypto / relatedemail

Namespace	No namespace
Diagram	
Type	email
Properties	content: simple
Source	<xsd:element name="relatedemail" type="email"/>

Element crypto / usedkeyid

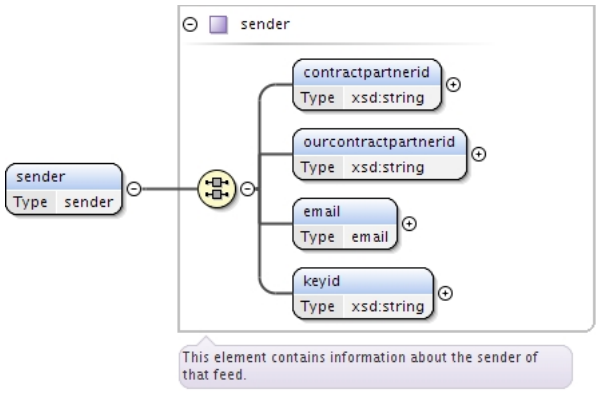
Namespace	No namespace
Diagram	
Type	keyid
Properties	content: simple
Source	<xsd:element name="usedkeyid" type="keyid"/>

Element crypto / usedpubkey

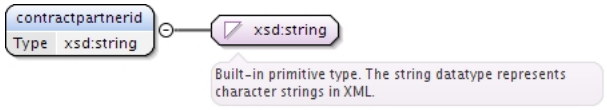
Namespace	No namespace
Diagram	
Type	xsd:base64Binary
Properties	content: simple
Source	<xsd:element name="usedpubkey" type="xsd:base64Binary"/>

Element feedinfo / sender

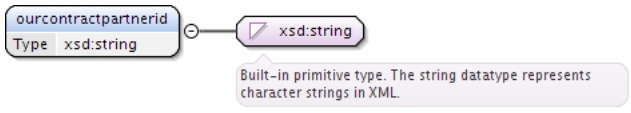
Namespace	No namespace

Diagram	
Type	sender
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid email keyid)
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Instance	<sender> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid> <email>{1,1}</email> <keyid>{1,1}</keyid> </sender>
Source	<xsd:element name="sender" type="sender"/>

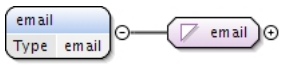
Element sender / contractpartnerid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="contractpartnerid" type="xsd:string"/>

Element sender / ourcontractpartnerid

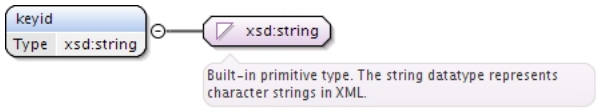
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="ourcontractpartnerid" type="xsd:string"/>

Element sender / email

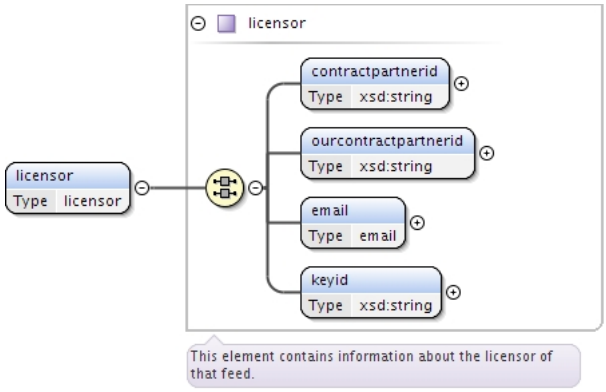
Namespace	No namespace
Diagram	
Type	email
Properties	content: simple
Source	<xsd:element name="email" type="email"/>

Element sender / keyid

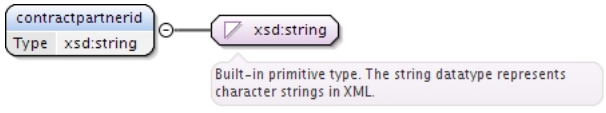
Namespace	No namespace

Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="keyid" type="xsd:string"/>

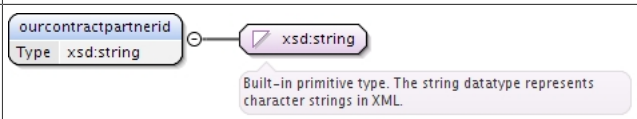
Element feedinfo / licensor

Namespace	No namespace
Diagram	
Type	licensor
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid email keyid)
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Instance	<licensor> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid> <email>{1,1}</email> <keyid>{1,1}</keyid> </licensor>
Source	<xsd:element name="licensor" type="licensor"/>

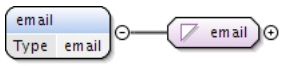
Element licensor / contractpartnerid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="contractpartnerid" type="xsd:string"/>

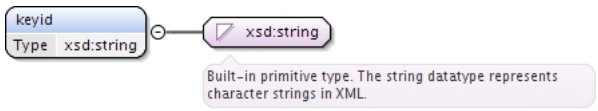
Element licensor / ourcontractpartnerid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="ourcontractpartnerid" type="xsd:string"/>

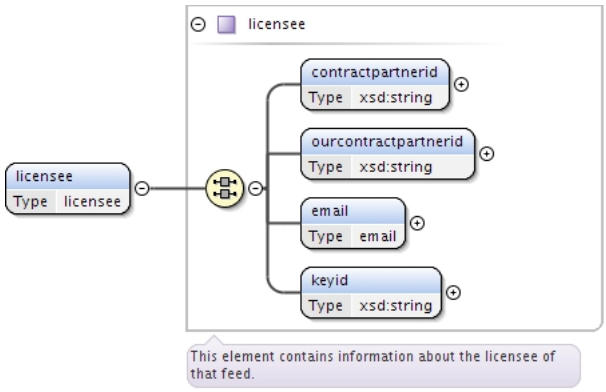
Element licensor / email

Namespace	No namespace
Diagram	
Type	email
Properties	content: simple
Source	`<xsd:element name="email" type="email"/>`

Element licensor / keyid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="keyid" type="xsd:string"/>`

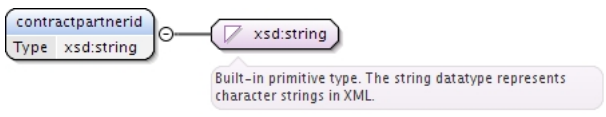
Element feedinfo / licensee

Namespace	No namespace
Diagram	
Type	licensee
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid email keyid)
Children	contractpartnerid, email, keyid, ourcontractpartnerid

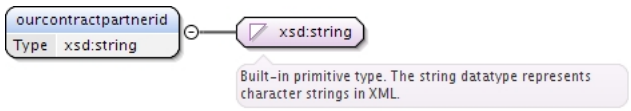
Instance	`<licensee>`
	`  <contractpartnerid>{1,1}</contractpartnerid>`
	`  <ourcontractpartnerid>{1,1}</ourcontractpartnerid>`
	`  <email>{1,1}</email>`
	`  <keyid>{1,1}</keyid>`
	`</licensee>`

Source	`<xsd:element name="licensee" type="licensee"/>`

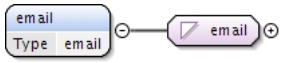
Element licensee / contractpartnerid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="contractpartnerid" type="xsd:string"/>`

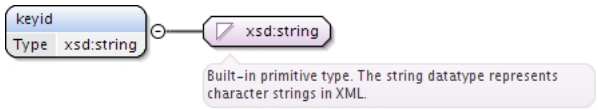
Element licensee / ourcontractpartnerid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="ourcontractpartnerid" type="xsd:string"/>

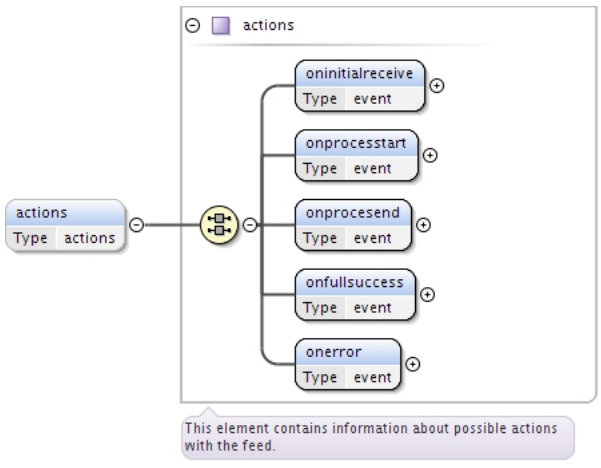
Element licensee / email

Namespace	No namespace
Diagram	
Type	email
Properties	content: simple
Source	<xsd:element name="email" type="email"/>

Element licensee / keyid

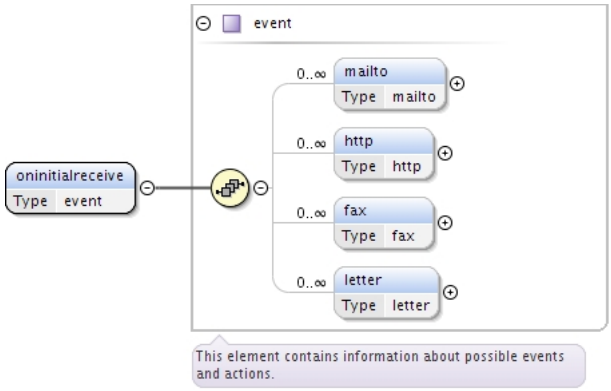
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="keyid" type="xsd:string"/>

Element feedinfo / actions

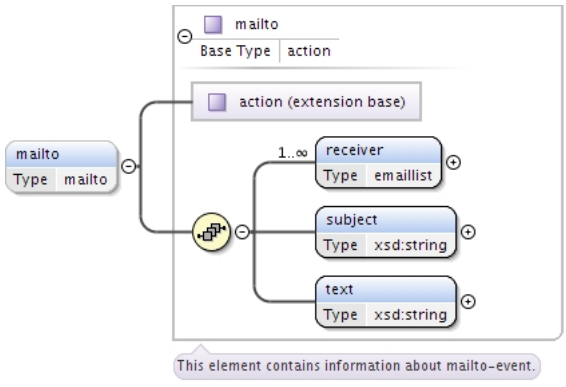
Namespace	No namespace
Diagram	
Type	actions
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(oninitialreceive onprocesstart onprocesend onfullsuccess onerror)
Children	onerror, onfullsuccess, oninitialreceive, onprocesend, onprocesstart
Instance	<actions>

	<oninitialreceive>{1,1}</oninitialreceive> <onprocesstart>{1,1}</onprocesstart> <onprocesend>{1,1}</onprocesend> <onfullsuccess>{1,1}</onfullsuccess> <onerror>{1,1}</onerror> </actions>
Source	<xsd:element name="actions" type="actions" maxOccurs="1" minOccurs="0"/>

Element actions / oninitialreceive

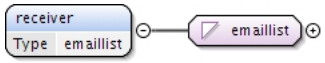
Namespace	No namespace
Diagram	
Type	event
Properties	content: complex
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto
Instance	<oninitialreceive> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </oninitialreceive>
Source	<xsd:element name="oninitialreceive" type="event"/>

Element event / mailto

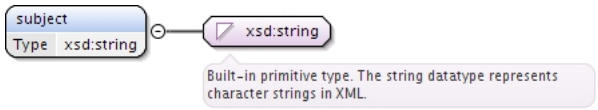
Namespace	No namespace
Diagram	
Type	mailto
Type hierarchy	• action • mailto
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	receiver+ , subject , text
Children	receiver, subject, text

Instance	<mailto> <receiver>{1,unbounded}</receiver> <subject>{1,1}</subject> <text>{1,1}</text> </mailto>
Source	<xsd:element name="mailto" type="mailto" minOccurs="0" maxOccurs="unbounded"/>

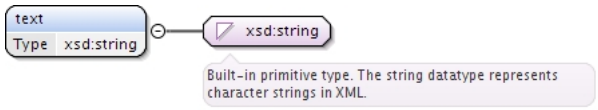
Element mailto / receiver

Namespace	No namespace
Diagram	
Type	emaillist
Properties	content: simple minOccurs: 1 maxOccurs: unbounded
Source	<xsd:element name="receiver" type="emaillist" minOccurs="1" maxOccurs="unbounded"/>

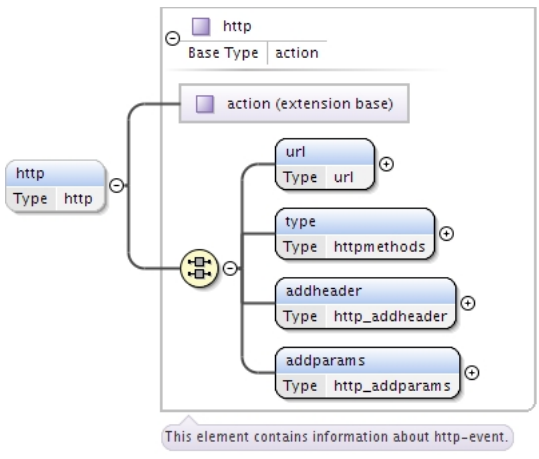
Element mailto / subject

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="subject" type="xsd:string"/>

Element mailto / text

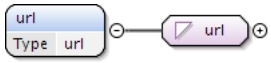
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="text" type="xsd:string"/>

Element event / http

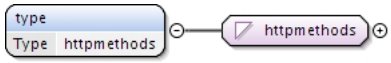
Namespace	No namespace
Diagram	

Type	http
Type hierarchy	• action • http
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(url type addheader addparams)
Children	addheader, addparams, type, url
Instance	<http> <url>{1,1}</url> <type>{1,1}</type> <addheader>{1,1}</addheader> <addparams>{1,1}</addparams> </http>
Source	<xsd:element name="http" type="http" minOccurs="0" maxOccurs="unbounded"/>

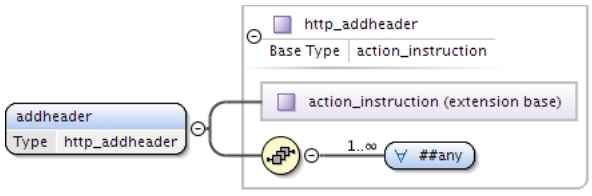
Element http / url

Namespace	No namespace
Diagram	
Type	url
Properties	content: simple
Source	<xsd:element name="url" type="url"/>

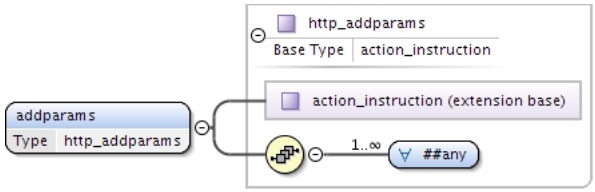
Element http / type

Namespace	No namespace
Diagram	
Type	httpmethods
Properties	content: simple
Facets	enumeration: GET enumeration: POST enumeration: HEAD
Source	<xsd:element name="type" type="httpmethods"/>

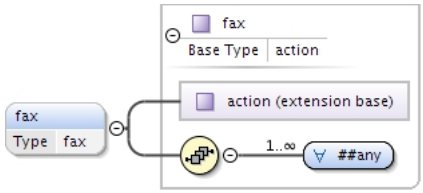
Element http / addheader

Namespace	No namespace
Diagram	
Type	http_addheader
Type hierarchy	• action_instruction • http_addheader
Properties	content: complex
Model	ANY element from ANY namespace
Source	<xsd:element name="addheader" type="http_addheader"/>

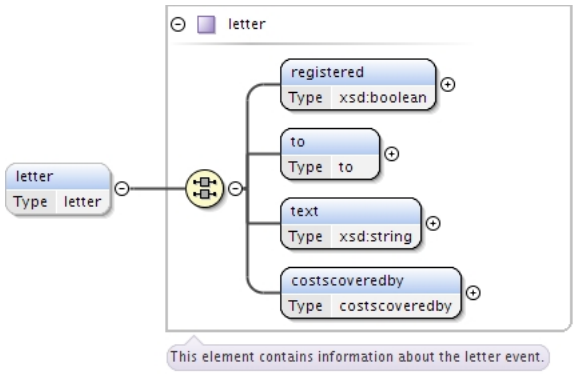
Element `http / addparams`

Namespace	No namespace
Diagram	
Type	http_addparams
Type hierarchy	• action_instruction • http_addparams
Properties	content: complex
Model	ANY element from ANY namespace
Source	`<xsd:element name="addparams" type="http_addparams"/>`

Element `event / fax`

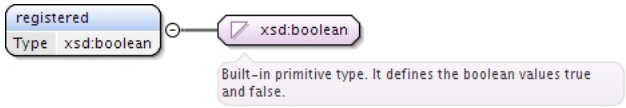
Namespace	No namespace
Diagram	
Type	fax
Type hierarchy	• action • fax
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ANY element from ANY namespace
Source	`<xsd:element name="fax" type="fax" minOccurs="0" maxOccurs="unbounded"/>`

Element `event / letter`

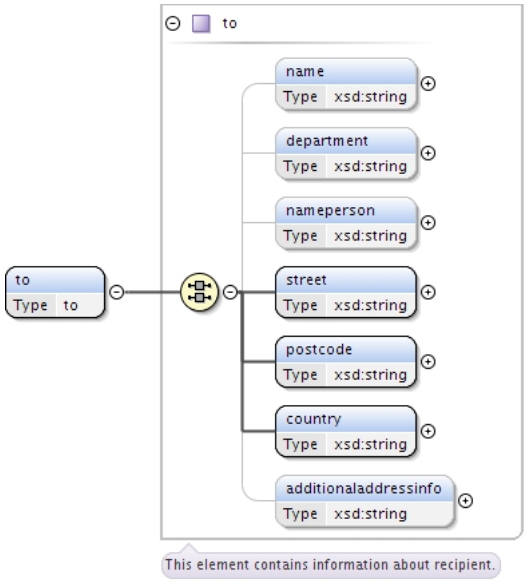
Namespace	No namespace
Diagram	
Type	letter
Properties	content: complex minOccurs: 0 maxOccurs: unbounded

Model	ALL(registered to text costscoveredby)
Children	costscoveredby, registered, text, to
Instance	<letter> <registered>{1,1}</registered> <to>{1,1}</to> <text>{1,1}</text> <costscoveredby>{1,1}</costscoveredby> </letter>
Source	<xsd:element name="letter" type="letter" minOccurs="0" maxOccurs="unbounded"/>

Element letter / registered

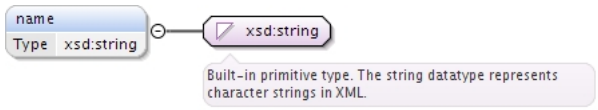
Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple
Source	<xsd:element name="registered" type="xsd:boolean"/>

Element letter / to

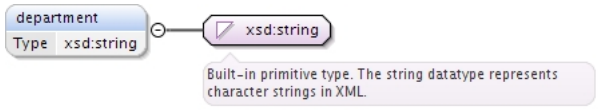
Namespace	No namespace
Diagram	
Type	to
Properties	content: complex
Model	ALL(name{0,1} department{0,1} nameperson{0,1} street postcode country additionaladdressinfo{0,1})
Children	additionaladdressinfo, country, department, name, nameperson, postcode, street
Instance	<to> <name>{0,1}</name> <department>{0,1}</department> <nameperson>{0,1}</nameperson> <street>{1,1}</street> <postcode>{1,1}</postcode> <country>{1,1}</country> <additionaladdressinfo>{0,1}</additionaladdressinfo> </to>
Source	<xsd:element name="to" type="to"/>

Element to / name

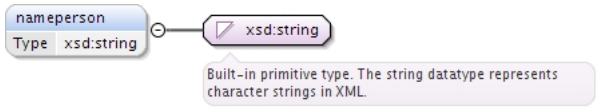
Namespace	No namespace

Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>`

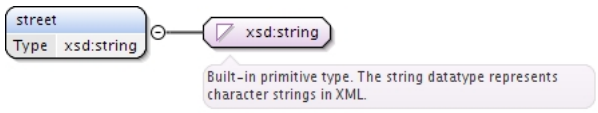
Element to / department

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="department" type="xsd:string" minOccurs="0" maxOccurs="1"/>`

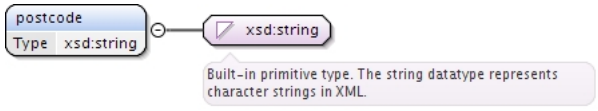
Element to / nameperson

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="nameperson" type="xsd:string" minOccurs="0" maxOccurs="1"/>`

Element to / street

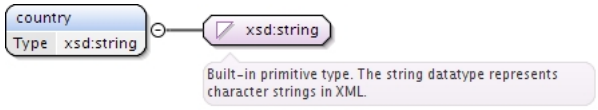
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="street" type="xsd:string"/>`

Element to / postcode

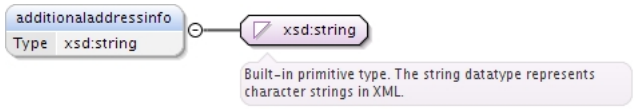
Namespace	No namespace
Diagram	
Type	xsd:string

Properties	content: simple
Source	<xsd:element name="postcode" type="xsd:string"/>

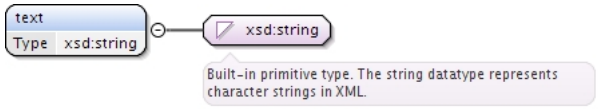
Element to / country

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="country" type="xsd:string"/>

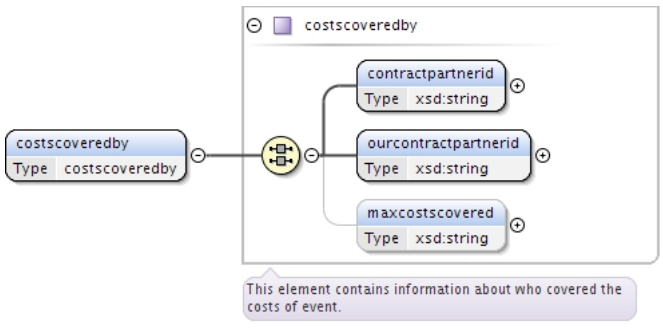
Element to / additionaladdressinfo

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="additionaladdressinfo" type="xsd:string" minOccurs="0" maxOccurs="1"/>

Element letter / text

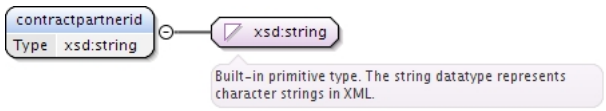
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="text" type="xsd:string"/>

Element letter / costscoveredby

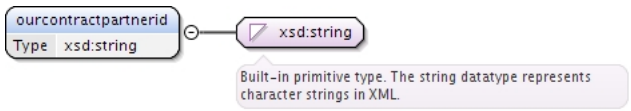
Namespace	No namespace
Diagram	
Type	costscoveredby
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid maxcostscovered{0,1})
Children	contractpartnerid, maxcostscovered, ourcontractpartnerid

Instance	<costscoveredby> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid> <maxcostscovered>{0,1}</maxcostscovered> </costscoveredby>
Source	<xsd:element name="costscoveredby" type="costscoveredby"/>

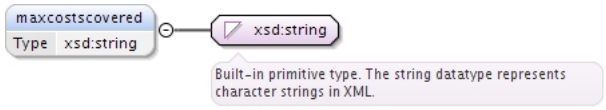
Element costscoveredby / contractpartnerid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="contractpartnerid" type="xsd:string"/>

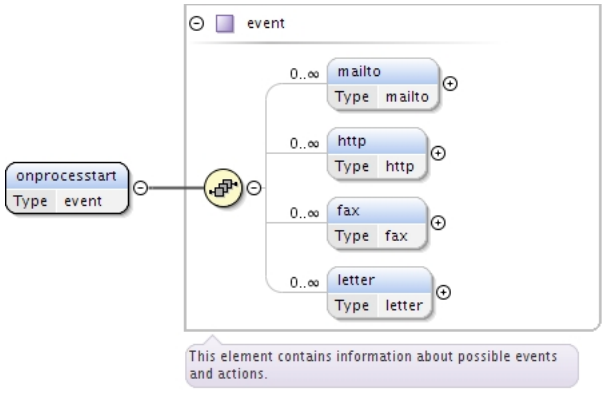
Element costscoveredby / ourcontractpartnerid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="ourcontractpartnerid" type="xsd:string"/>

Element costscoveredby / maxcostscovered

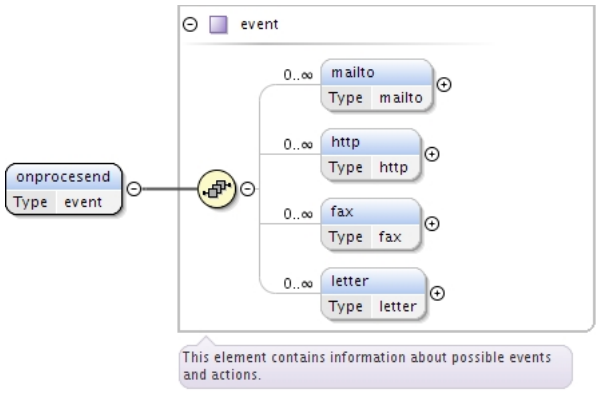
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="maxcostscovered" type="xsd:string" minOccurs="0" maxOccurs="1"/>

Element actions / onprocesstart

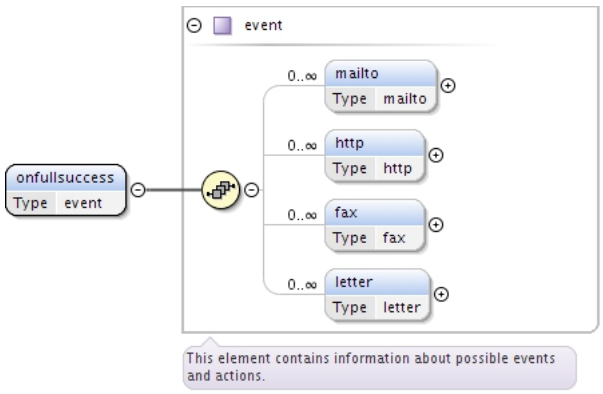
Namespace	No namespace
Diagram	

Type	event
Properties	content: complex
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto
Instance	<onprocesstart> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </onprocesstart>
Source	<xsd:element name="onprocesstart" type="event"/>

Element actions / onprocesend

Namespace	No namespace
Diagram	
Type	event
Properties	content: complex
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto
Instance	<onprocesend> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </onprocesend>
Source	<xsd:element name="onprocesend" type="event"/>

Element actions / onfullsuccess

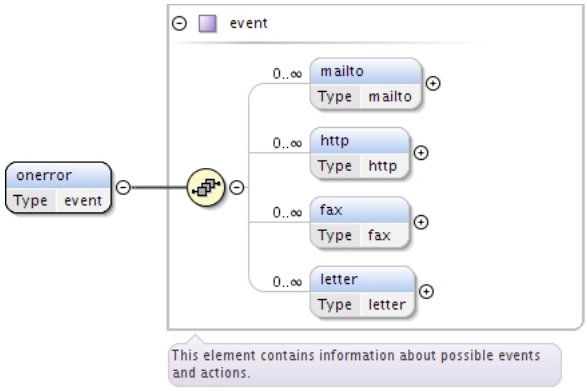
Namespace	No namespace
Diagram	
Type	event
Properties	content: complex

Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto

`Instance`	`<onfullsuccess>`
	`  <mailto>{0,unbounded}</mailto>`
	`  <http>{0,unbounded}</http>`
	`  <fax>{0,unbounded}</fax>`
	`  <letter>{0,unbounded}</letter>`
	`</onfullsuccess>`

`Source`	`<xsd:element name="onfullsuccess" type="event"/>`

Element `actions / onerror`

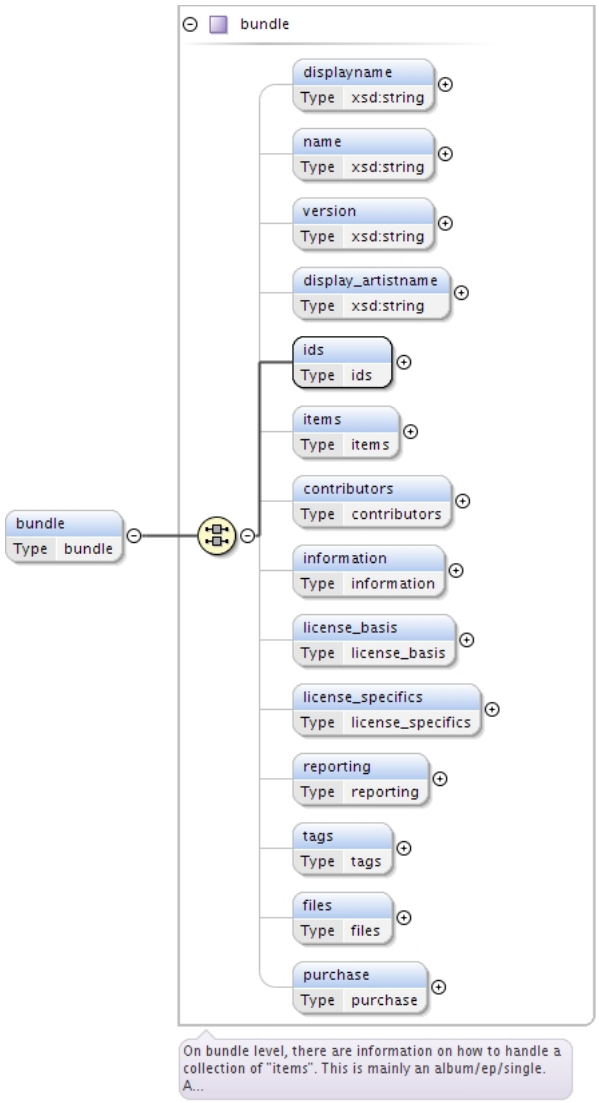
Namespace	No namespace
Diagram	
Type	event
Properties	content: `complex`
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto

`Instance`	`<onerror>`
	`  <mailto>{0,unbounded}</mailto>`
	`  <http>{0,unbounded}</http>`
	`  <fax>{0,unbounded}</fax>`
	`  <letter>{0,unbounded}</letter>`
	`</onerror>`

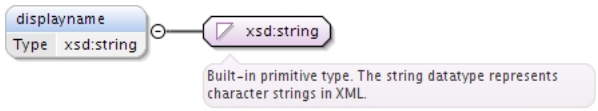
`Source`	`<xsd:element name="onerror" type="event"/>`

Element `feed / bundle`

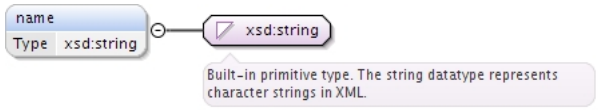
Namespace	No namespace

Diagram	
Type	bundle
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(displayname{0,1} name{0,1} version{0,1} display_artistname{0,1} ids items{0,1} contributors{0,1} information{0,1} license_basis{0,1} license_specifics{0,1} reporting{0,1} tags{0,1} files{0,1} purchase{0,1})
Children	contributors, display_artistname, displayname, files, ids, information, items, license_basis, license_specifics, name, purchase, reporting, tags, version
Instance	<bundle> <displayname>{0,1}</displayname> <name>{0,1}</name> <version>{0,1}</version> <display_artistname>{0,1}</display_artistname> <ids>{1,1}</ids> <items>{0,1}</items> <contributors>{0,1}</contributors> <information>{0,1}</information> <license_basis>{0,1}</license_basis> <license_specifics>{0,1}</license_specifics> <reporting>{0,1}</reporting> <tags>{0,1}</tags> <files>{0,1}</files> <purchase>{0,1}</purchase> </bundle>
Source	<xsd:element name="bundle" type="bundle" maxOccurs="unbounded" minOccurs="0"/>

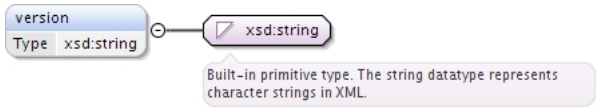
Element bundle / displayname

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="displayname" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

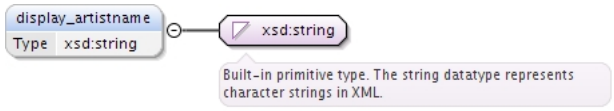
Element bundle / name

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="name" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element bundle / version

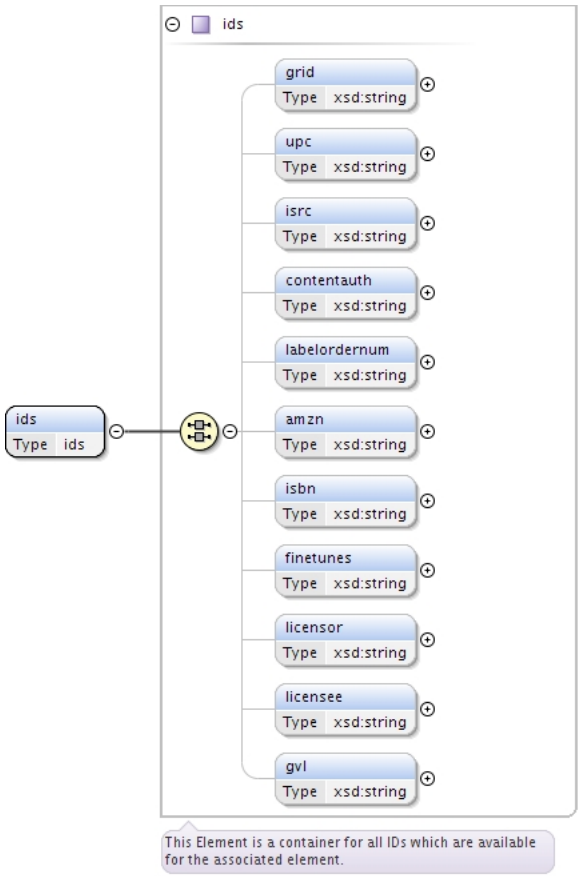
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="version" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element bundle / display_artistname

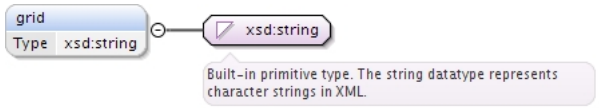
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="display_artistname" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element bundle / ids

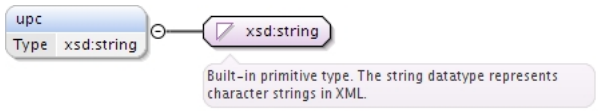
Namespace	No namespace

Diagram	
Type	ids
Properties	content: complex
Model	ALL(grid{0,1} upc{0,1} isrc{0,1} contentauth{0,1} labelordernum{0,1} amzn{0,1} isbn{0,1} finetunes{0,1} licensor{0,1} licensee{0,1} gvl{0,1})
Children	amzn, contentauth, finetunes, grid, gvl, isbn, isrc, labelordernum, licensee, licensor, upc
Instance	<ids> <grid>{0,1}</grid> <upc>{0,1}</upc> <isrc>{0,1}</isrc> <contentauth>{0,1}</contentauth> <labelordernum>{0,1}</labelordernum> <amzn>{0,1}</amzn> <isbn>{0,1}</isbn> <finetunes>{0,1}</finetunes> <licensor>{0,1}</licensor> <licensee>{0,1}</licensee> <gvl>{0,1}</gvl> </ids>
Source	<xsd:element name="ids" type="ids"/>

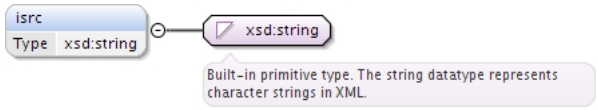
Element ids / grid

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="grid" type="xsd:string" maxOccurs="1" minOccurs="0"/>

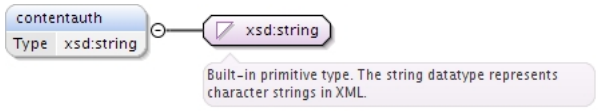
Element ids / upc

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="upc" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

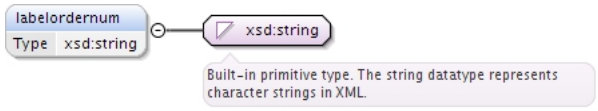
Element ids / isrc

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="isrc" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element ids / contentauth

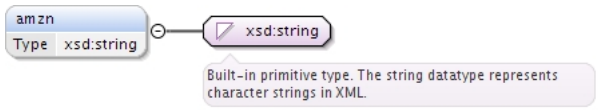
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="contentauth" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element ids / labelordernum

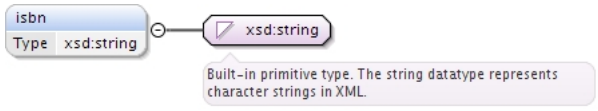
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="labelordernum" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element ids / amzn

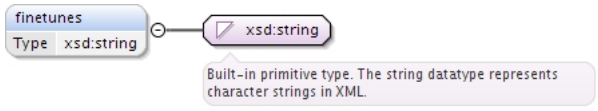
Namespace	No namespace

Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="amzn" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

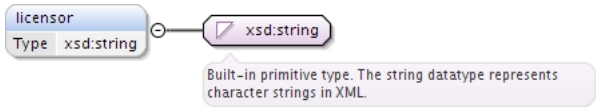
Element ids / isbn

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="isbn" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element ids / finetunes

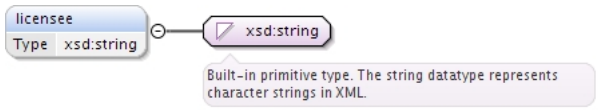
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="finetunes" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element ids / licensor

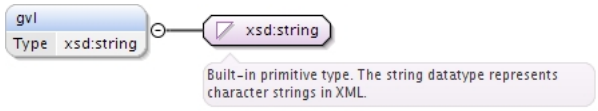
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="licensor" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element ids / licensee

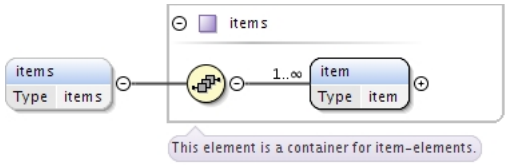
Namespace	No namespace

Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="licensee" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element ids / gvl

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="gvl" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element bundle / items

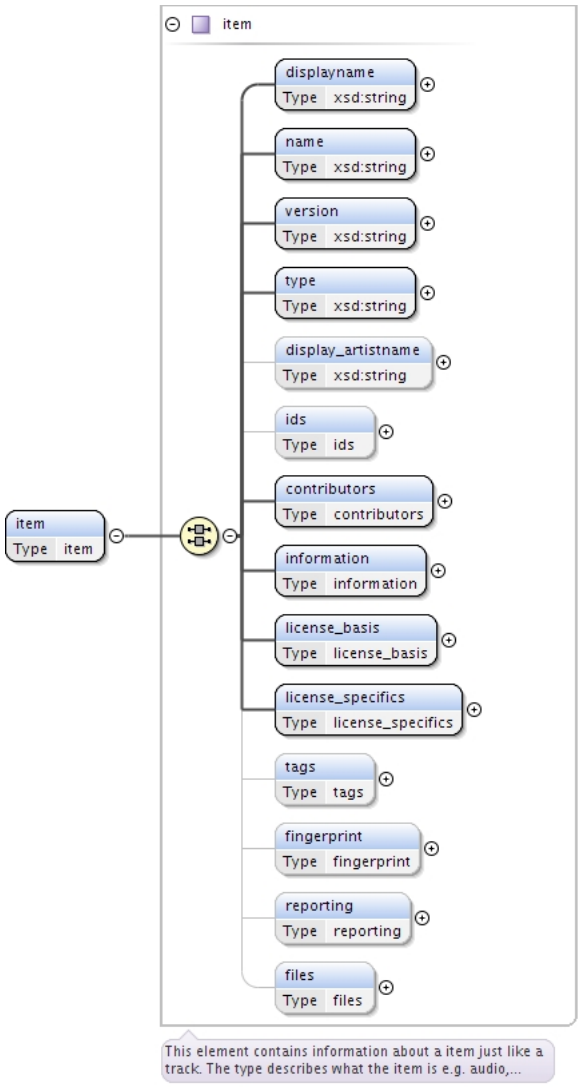
Namespace	No namespace
Diagram	
Type	items
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	item+
Children	item

Instance	`<items>`
	`  <item>{1,unbounded}</item>`
	`</items>`

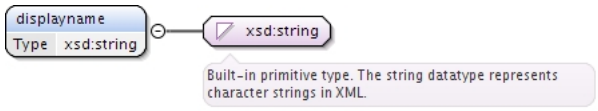
Source	`<xsd:element name="items" type="items" maxOccurs="1" minOccurs="0"/>`

Element items / item

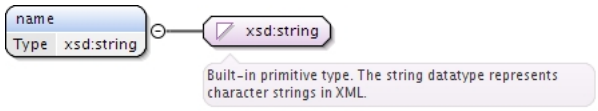
Namespace	No namespace

Diagram	
Type	item
Properties	content: complex minOccurs: 1 maxOccurs: unbounded
Model	ALL(displayname name version type display_artistname{0,1} ids{0,1} contributors information license_basis license_specifics tags{0,1} fingerprint{0,1} reporting{0,1} files{0,1})
Children	contributors, display_artistname, displayname, files, fingerprint, ids, information, license_basis, license_specifics, name, reporting, tags, type, version
Instance	<item> <displayname>{1,1}</displayname> <name>{1,1}</name> <version>{1,1}</version> <type>{1,1}</type> <display_artistname>{0,1}</display_artistname> <ids>{0,1}</ids> <contributors>{1,1}</contributors> <information>{1,1}</information> <license_basis>{1,1}</license_basis> <license_specifics>{1,1}</license_specifics> <tags>{0,1}</tags> <fingerprint>{0,1}</fingerprint> <reporting>{0,1}</reporting> <files>{0,1}</files> </item>
Source	<xsd:element name="item" type="item" maxOccurs="unbounded" minOccurs="1"/>

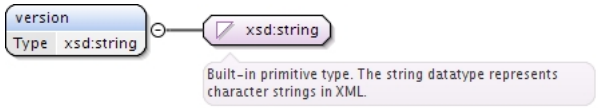
Element item / displayname

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="displayname" type="xsd:string"/>`

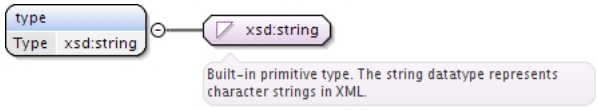
Element item / name

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="name" type="xsd:string"/>`

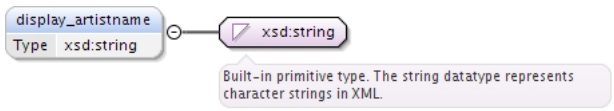
Element item / version

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="version" type="xsd:string"/>`

Element item / type

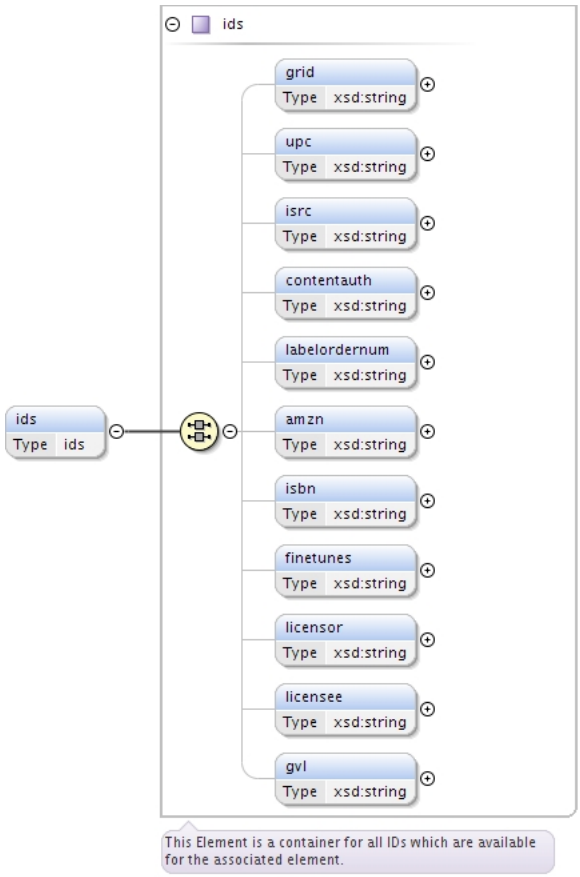
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="type" type="xsd:string"/>`

Element item / display_artistname

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple

	minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="display_artistname" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element item / ids

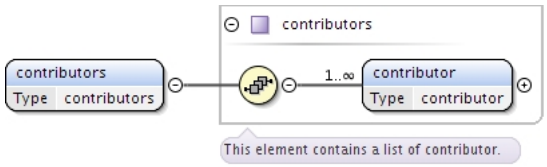
Namespace	No namespace
Diagram	
Type	ids
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(grid{0,1} upc{0,1} isrc{0,1} contentauth{0,1} labelordernum{0,1} amzn{0,1} isbn{0,1} finetunes{0,1} licensor{0,1} licensee{0,1} gvl{0,1})
Children	amzn, contentauth, finetunes, grid, gvl, isbn, isrc, labelordernum, licensee, licensor, upc

Instance	`<ids>`
	`  <grid>{0,1}</grid>`
	`  <upc>{0,1}</upc>`
	`  <isrc>{0,1}</isrc>`
	`  <contentauth>{0,1}</contentauth>`
	`  <labelordernum>{0,1}</labelordernum>`
	`  <amzn>{0,1}</amzn>`
	`  <isbn>{0,1}</isbn>`
	`  <finetunes>{0,1}</finetunes>`
	`  <licensor>{0,1}</licensor>`
	`  <licensee>{0,1}</licensee>`
	`  <gvl>{0,1}</gvl>`
	`</ids>`

Source	`<xsd:element name="ids" type="ids" maxOccurs="1" minOccurs="0"/>`

Element item / contributors

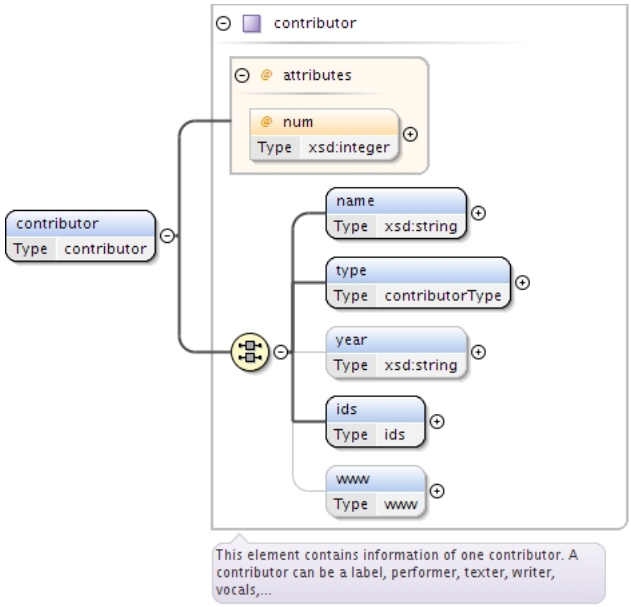
Namespace	No namespace

Diagram	
Type	contributors
Properties	content: complex
Model	contributor+
Children	contributor

Instance	`<contributors>`
	`  <contributor num="">{1,unbounded}</contributor>`
	`</contributors>`

Source	`<xsd:element name="contributors" type="contributors"/>`

Element contributors / contributor

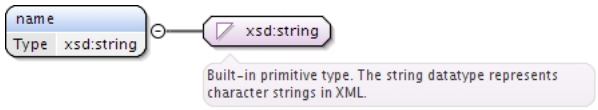
Namespace	No namespace
Diagram	
Type	contributor
Properties	content: complex minOccurs: 1 maxOccurs: unbounded
Model	ALL(name type year{0,1} ids www{0,1})
Children	ids, name, type, www, year

Instance	`<contributor num="">`
	`  <name>{1,1}</name>`
	`  <type>{1,1}</type>`
	`  <year>{0,1}</year>`
	`  <ids>{1,1}</ids>`
	`  <www>{0,1}</www>`
	`</contributor>`

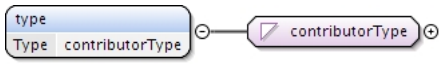
Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **num** / xsd:integer / / / optional
Source	`<xsd:element name="contributor" type="contributor" maxOccurs="unbounded" minOccurs="1"/>`

Element contributor / name

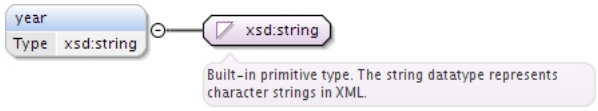
Namespace	No namespace

Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="name" type="xsd:string"/>

Element contributor / type

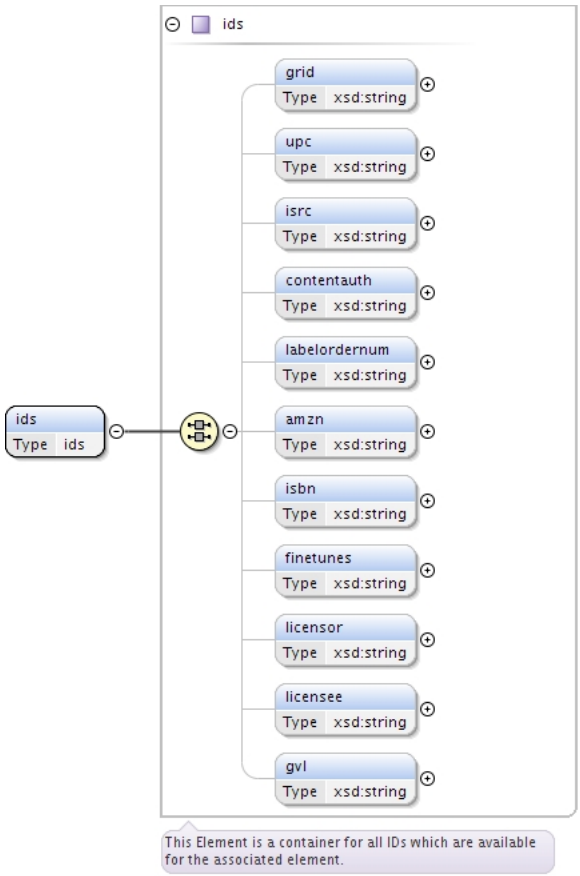
Namespace	No namespace
Diagram	
Type	contributorType
Properties	content: simple
Facets	enumeration: label enumeration: performer enumeration: texter enumeration: writer enumeration: vocals enumeration: conductor enumeration: display_artist enumeration: compilator enumeration: copyright enumeration: production enumeration: clearinghouse
Source	<xsd:element name="type" type="contributorType"/>

Element contributor / year

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="year" type="xsd:string" maxOccurs="1" minOccurs="0"/>

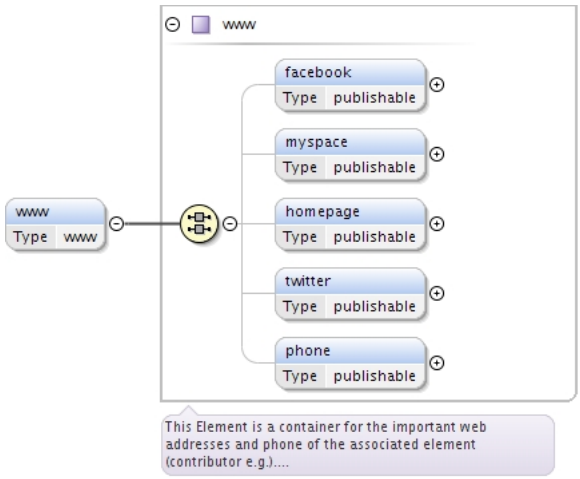
Element contributor / ids

Namespace	No namespace

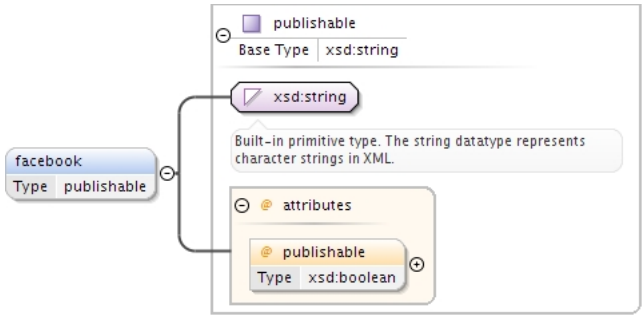
Diagram	
Type	ids
Properties	content: complex
Model	ALL(grid{0,1} upc{0,1} isrc{0,1} contentauth{0,1} labelordernum{0,1} amzn{0,1} isbn{0,1} finetunes{0,1} licensor{0,1} licensee{0,1} gvl{0,1})
Children	amzn, contentauth, finetunes, grid, gvl, isbn, isrc, labelordernum, licensee, licensor, upc
Instance	<ids> <grid>{0,1}</grid> <upc>{0,1}</upc> <isrc>{0,1}</isrc> <contentauth>{0,1}</contentauth> <labelordernum>{0,1}</labelordernum> <amzn>{0,1}</amzn> <isbn>{0,1}</isbn> <finetunes>{0,1}</finetunes> <licensor>{0,1}</licensor> <licensee>{0,1}</licensee> <gvl>{0,1}</gvl> </ids>
Source	<xsd:element name="ids" type="ids"/>

Element contributor / www

Namespace	No namespace

Diagram	
Type	www
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(facebook{0,1} myspace{0,1} homepage{0,1} twitter{0,1} phone{0,1})
Children	facebook, homepage, myspace, phone, twitter
Instance	<www> <facebook publishable="">{0,1}</facebook> <myspace publishable="">{0,1}</myspace> <homepage publishable="">{0,1}</homepage> <twitter publishable="">{0,1}</twitter> <phone publishable="">{0,1}</phone> </www>
Source	<xsd:element name="www" type="www" maxOccurs="1" minOccurs="0"/>

Element www / facebook

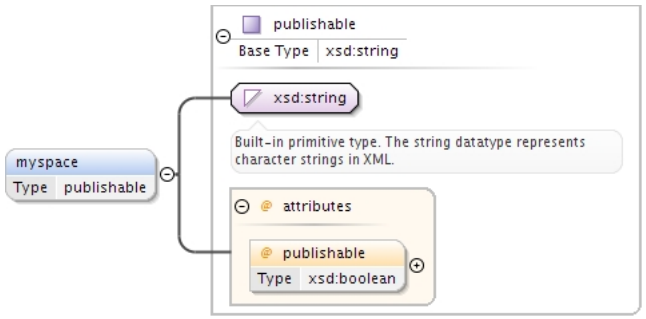
Namespace	No namespace
Diagram	
Type	publishable
Properties	content: complex minOccurs: 0 maxOccurs: 1

Attributes	QName	Type	Fixed	Default	Use
	publishable	xsd:boolean			optional

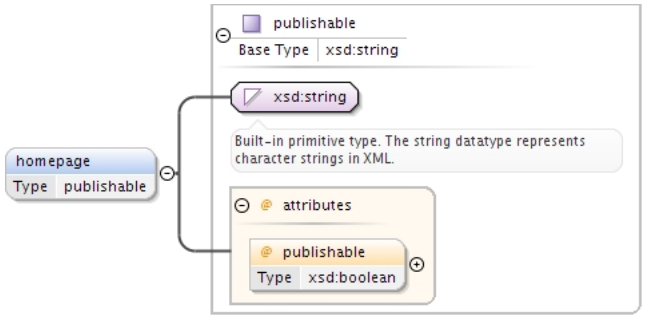
Source	<xsd:element name="facebook" type="publishable" maxOccurs="1" minOccurs="0"/>

Element www / myspace

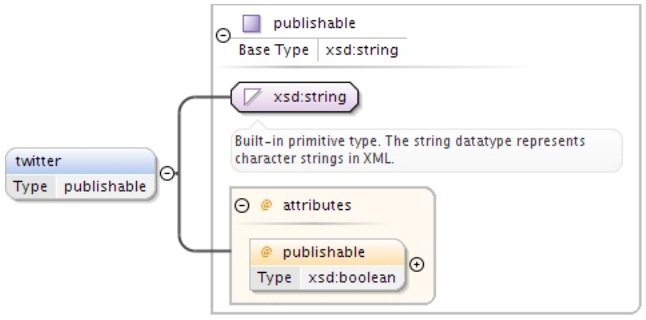
Namespace	No namespace

Diagram	
Type	publishable
Properties	content: complex minOccurs: 0 maxOccurs: 1
Attributes	**QName** publishable; **Type** xsd:boolean; **Fixed**; **Default**; **Use** optional
Source	`<xsd:element name="myspace" type="publishable" maxOccurs="1" minOccurs="0"/>`

Element www / homepage

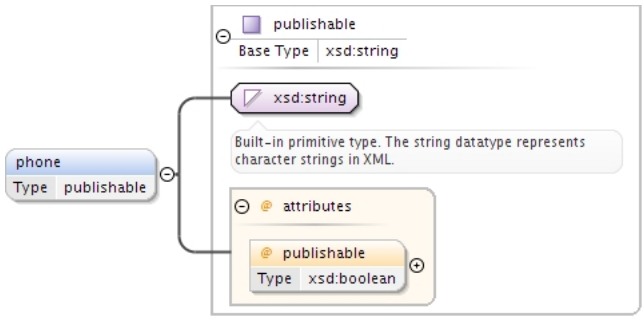
Namespace	No namespace
Diagram	
Type	publishable
Properties	content: complex minOccurs: 0 maxOccurs: 1
Attributes	**QName** publishable; **Type** xsd:boolean; **Fixed**; **Default**; **Use** optional
Source	`<xsd:element name="homepage" type="publishable" maxOccurs="1" minOccurs="0"/>`

Element www / twitter

Namespace	No namespace
Diagram	
Type	publishable

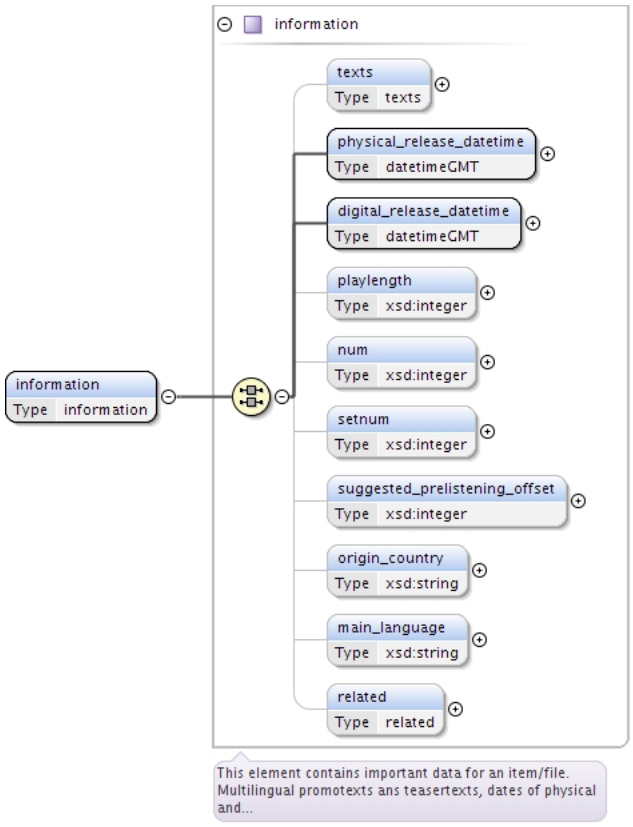
Properties	content:	complex			
	minOccurs:	0			
	maxOccurs:	1			
Attributes	**QName**	**Type**	**Fixed**	**Default**	**Use**
	publishable	xsd:boolean			optional
Source	<xsd:element name="twitter" type="publishable" maxOccurs="1" minOccurs="0"/>				

Element www / phone

Namespace	No namespace
Diagram	
Type	publishable
Properties	content: complex minOccurs: 0 maxOccurs: 1
Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **publishable** / xsd:boolean / / / optional
Source	<xsd:element name="phone" type="publishable" maxOccurs="1" minOccurs="0"/>

Element item / information

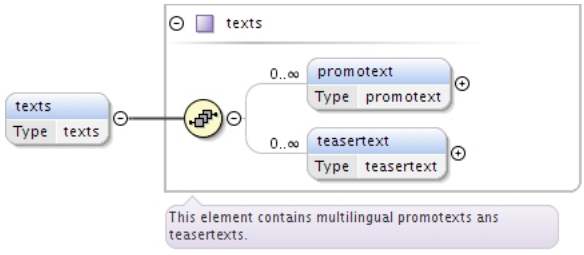
Namespace	No namespace

Diagram	
Type	information
Properties	content: complex
Model	ALL(texts{0,1} physical_release_datetime digital_release_datetime playlength{0,1} num{0,1} setnum{0,1} suggested_prelistening_offset{0,1} origin_country{0,1} main_language{0,1} related{0,1})
Children	digital_release_datetime, main_language, num, origin_country, physical_release_datetime, playlength, related, setnum, suggested_prelistening_offset, texts

Instance	`<information>`
	`  <texts>{0,1}</texts>`
	`  <physical_release_datetime>{1,1}</physical_release_datetime>`
	`  <digital_release_datetime>{1,1}</digital_release_datetime>`
	`  <playlength>{0,1}</playlength>`
	`  <num>{0,1}</num>`
	`  <setnum>{0,1}</setnum>`
	`  <suggested_prelistening_offset>{0,1}</suggested_prelistening_offset>`
	`  <origin_country>{0,1}</origin_country>`
	`  <main_language>{0,1}</main_language>`
	`  <related>{0,1}</related>`
	`</information>`

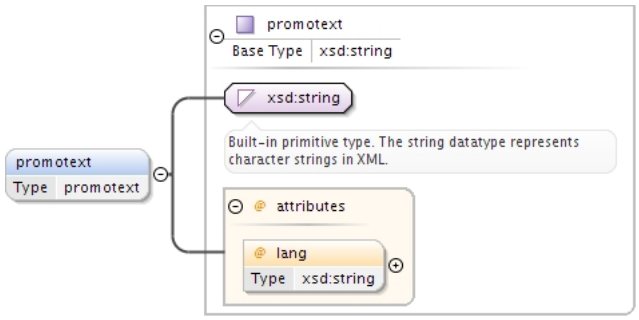
Source	`<xsd:element name="information" type="information"/>`

Element information / texts

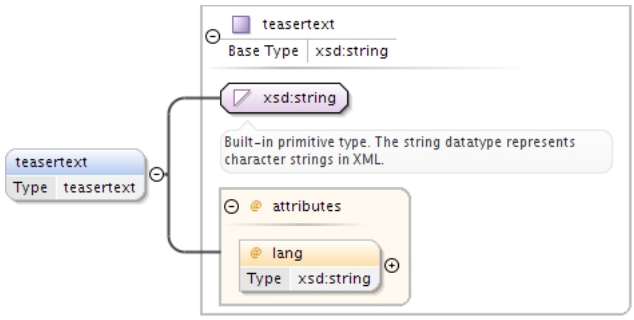
Namespace	No namespace
Diagram	
Type	texts
Properties	content: complex

	minOccurs: 0 maxOccurs: 1
Model	promotext* , teasertext*
Children	promotext, teasertext
Instance	<texts> <promotext lang="">{0,unbounded}</promotext> <teasertext lang="">{0,unbounded}</teasertext> </texts>
Source	<xsd:element name="texts" type="texts" maxOccurs="1" minOccurs="0"/>

Element texts / promotext

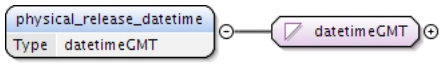
Namespace	No namespace
Diagram	
Type	promotext
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **lang** / xsd:string / / / optional
Source	<xsd:element name="promotext" type="promotext" maxOccurs="unbounded" minOccurs="0"/>

Element texts / teasertext

Namespace	No namespace
Diagram	
Type	teasertext
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Attributes	**QName** / **Type** / **Fixed** / **Default** / **Use** **lang** / xsd:string / / / optional
Source	<xsd:element name="teasertext" type="teasertext" maxOccurs="unbounded" minOccurs="0"/>

Element information / physical_release_datetime

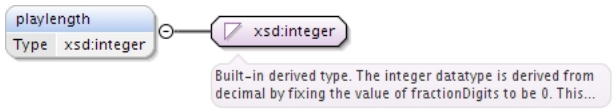
Namespace	No namespace

Diagram	
Type	datetimeGMT
Properties	content: simple
Facets	pattern \d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}
Source	<xsd:element name="physical_release_datetime" type="datetimeGMT"/>

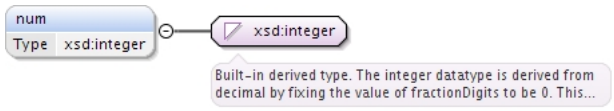
Element information / digital_release_datetime

Namespace	No namespace
Diagram	
Type	datetimeGMT
Properties	content: simple
Facets	pattern \d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}
Source	<xsd:element name="digital_release_datetime" type="datetimeGMT"/>

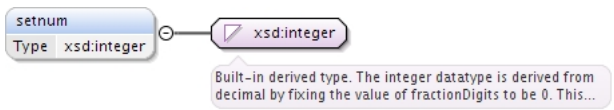
Element information / playlength

Namespace	No namespace
Diagram	
Type	xsd:integer
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="playlength" type="xsd:integer" maxOccurs="1" minOccurs="0"/>

Element information / num

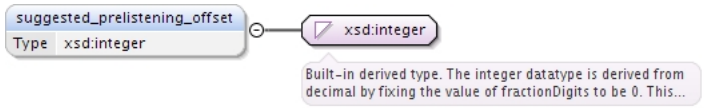
Namespace	No namespace
Diagram	
Type	xsd:integer
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="num" type="xsd:integer" maxOccurs="1" minOccurs="0"/>

Element information / setnum

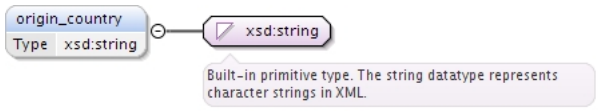
Namespace	No namespace
Diagram	

Type	xsd:integer
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="setnum" type="xsd:integer" maxOccurs="1" minOccurs="0"/>`

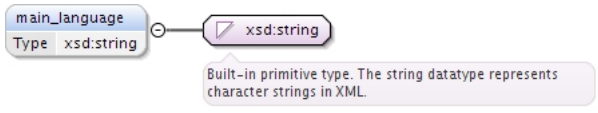
Element information / suggested_prelistening_offset

Namespace	No namespace
Diagram	
Type	xsd:integer
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="suggested_prelistening_offset" type="xsd:integer" maxOccurs="1" minOccurs="0"/>`

Element information / origin_country

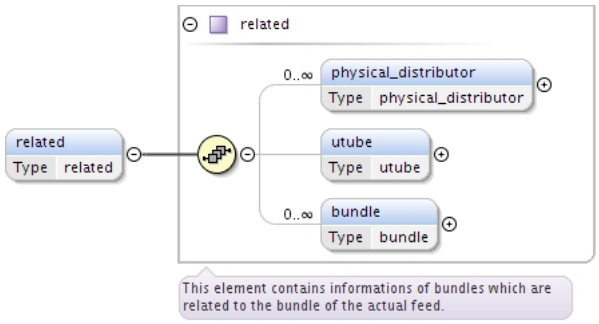
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="origin_country" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element information / main_language

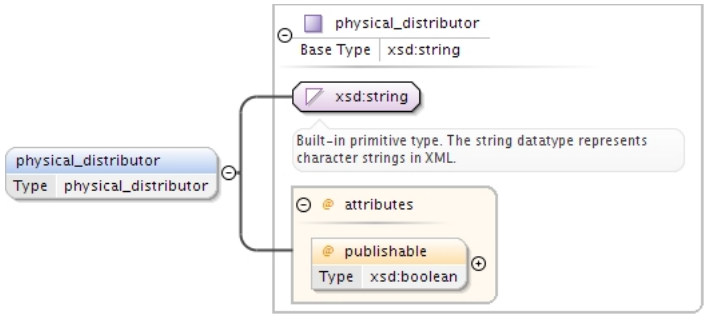
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="main_language" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element information / related

Namespace	No namespace

Diagram	
Type	related
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	physical_distributor* , utube{0,1} , bundle*
Children	bundle, physical_distributor, utube
Instance	<related> <physical_distributor publishable="">{0,unbounded}</physical_distributor> <utube>{0,1}</utube> <bundle>{0,unbounded}</bundle> </related>
Source	<xsd:element name="related" type="related" maxOccurs="1" minOccurs="0"/>

Element related / physical_distributor

Namespace	No namespace
Diagram	
Type	physical_distributor
Properties	content: complex minOccurs: 0 maxOccurs: unbounded

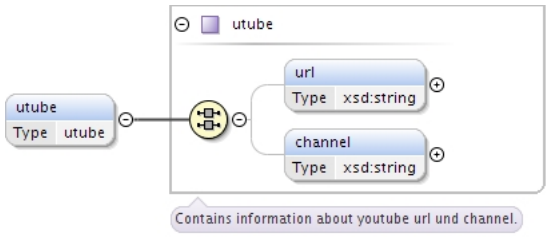
Attributes

QName	Type	Fixed	Default	Use
publishable	xsd:boolean			optional

Source

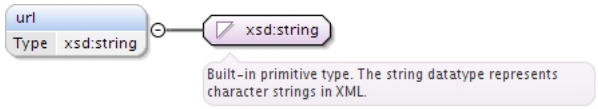
```
<xsd:element name="physical_distributor" type="physical_distributor" maxOccurs="unbounded"
 minOccurs="0"/>
```

Element related / utube

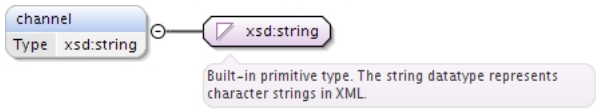
Namespace	No namespace
Diagram	

Type	utube
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(url{0,1} channel{0,1})
Children	channel, url
Instance	<utube> <url>{0,1}</url> <channel>{0,1}</channel> </utube>
Source	<xsd:element name="utube" type="utube" maxOccurs="1" minOccurs="0"/>

Element utube / url

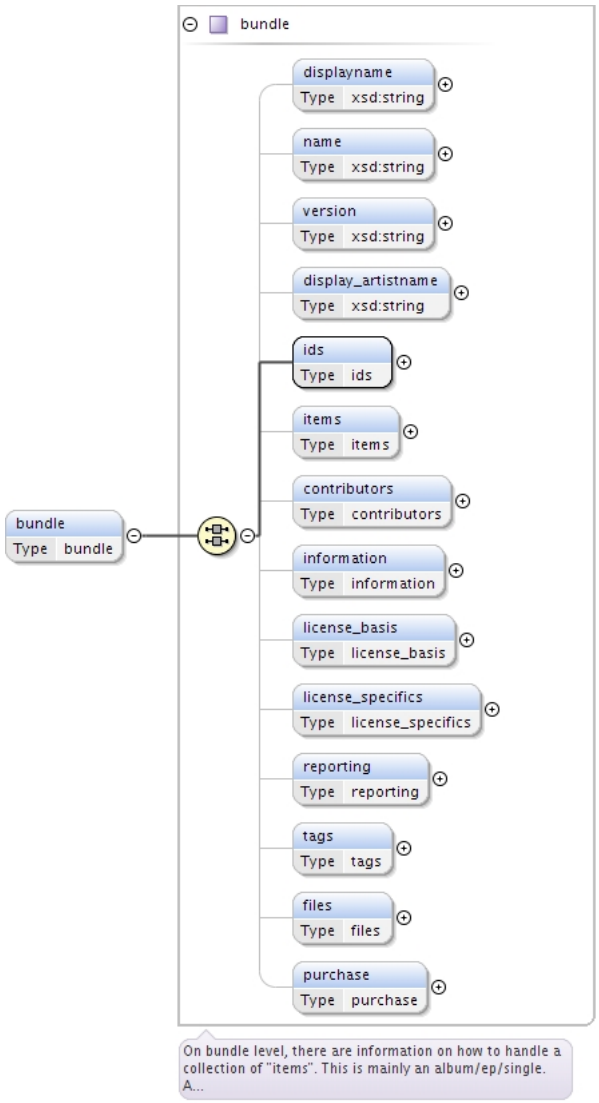
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="url" type="xsd:string" maxOccurs="1" minOccurs="0"/>

Element utube / channel

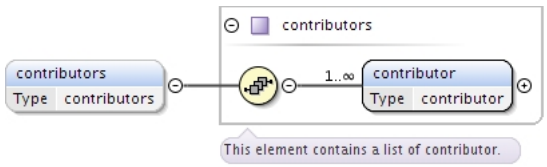
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="channel" type="xsd:string" maxOccurs="1" minOccurs="0"/>

Element related / bundle

Namespace	No namespace

Diagram	
Type	bundle
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(displayname{0,1} name{0,1} version{0,1} display_artistname{0,1} ids items{0,1} contributors{0,1} information{0,1} license_basis{0,1} license_specifics{0,1} reporting{0,1} tags{0,1} files{0,1} purchase{0,1})
Children	contributors, display_artistname, displayname, files, ids, information, items, license_basis, license_specifics, name, purchase, reporting, tags, version
Instance	<bundle> <displayname>{0,1}</displayname> <name>{0,1}</name> <version>{0,1}</version> <display_artistname>{0,1}</display_artistname> <ids>{1,1}</ids> <items>{0,1}</items> <contributors>{0,1}</contributors> <information>{0,1}</information> <license_basis>{0,1}</license_basis> <license_specifics>{0,1}</license_specifics> <reporting>{0,1}</reporting> <tags>{0,1}</tags> <files>{0,1}</files> <purchase>{0,1}</purchase> </bundle>
Source	<xsd:element name="bundle" type="bundle" maxOccurs="unbounded" minOccurs="0"/>

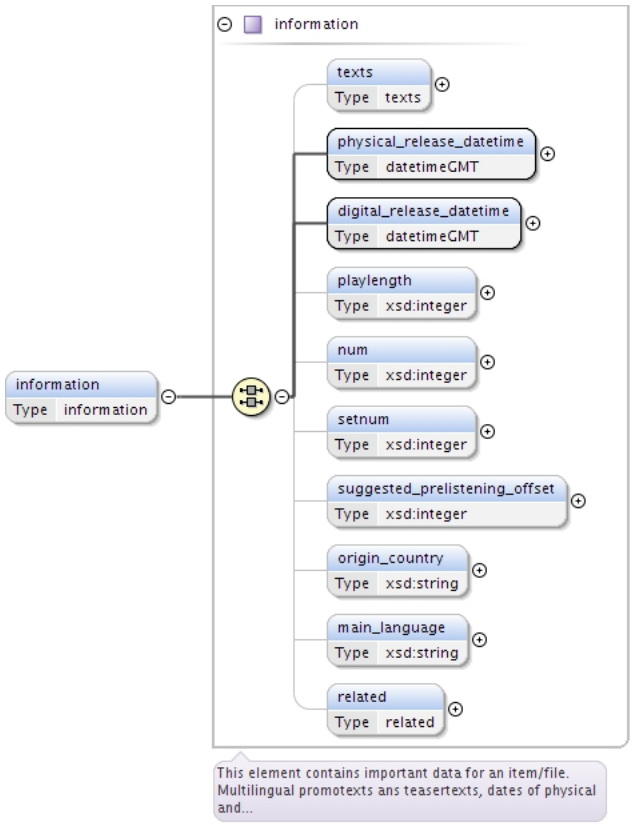
Element bundle / contributors

Namespace	No namespace
Diagram	
Type	contributors
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	contributor+
Children	contributor

Instance	`<contributors>`
	`  <contributor num="">{1,unbounded}</contributor>`
	`</contributors>`

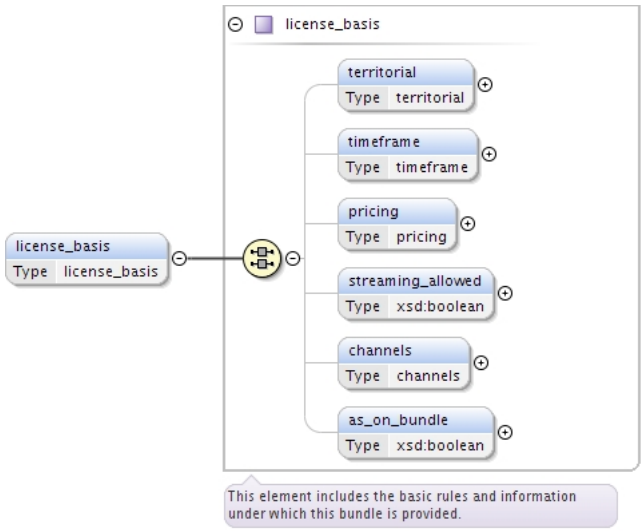
Source	`<xsd:element name="contributors" type="contributors" maxOccurs="1" minOccurs="0"/>`

Element bundle / information

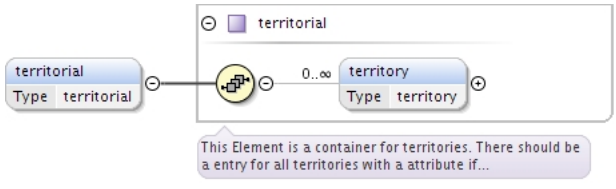
Namespace	No namespace
Diagram	
Type	information
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(texts{0,1} physical_release_datetime digital_release_datetime playlength{0,1} num{0,1} setnum{0,1} suggested_prelistening_offset{0,1} origin_country{0,1} main_language{0,1} related{0,1})
Children	digital_release_datetime, main_language, num, origin_country, physical_release_datetime, playlength, related, setnum, suggested_prelistening_offset, texts

Instance	<information> <texts>{0,1}</texts> <physical_release_datetime>{1,1}</physical_release_datetime> <digital_release_datetime>{1,1}</digital_release_datetime> <playlength>{0,1}</playlength> <num>{0,1}</num> <setnum>{0,1}</setnum> <suggested_prelistening_offset>{0,1}</suggested_prelistening_offset> <origin_country>{0,1}</origin_country> <main_language>{0,1}</main_language> <related>{0,1}</related> </information>
Source	<xsd:element name="information" type="information" maxOccurs="1" minOccurs="0"/>

Element bundle / license_basis

Namespace	No namespace
Diagram	
Type	license_basis
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(territorial{0,1} timeframe{0,1} pricing{0,1} streaming_allowed{0,1} channels{0,1} as_on_bundle{0,1})
Children	as_on_bundle, channels, pricing, streaming_allowed, territorial, timeframe
Instance	<license_basis> <territorial>{0,1}</territorial> <timeframe>{0,1}</timeframe> <pricing>{0,1}</pricing> <streaming_allowed>{0,1}</streaming_allowed> <channels>{0,1}</channels> <as_on_bundle>{0,1}</as_on_bundle> </license_basis>
Source	<xsd:element name="license_basis" type="license_basis" maxOccurs="1" minOccurs="0"/>

Element license_basis / territorial

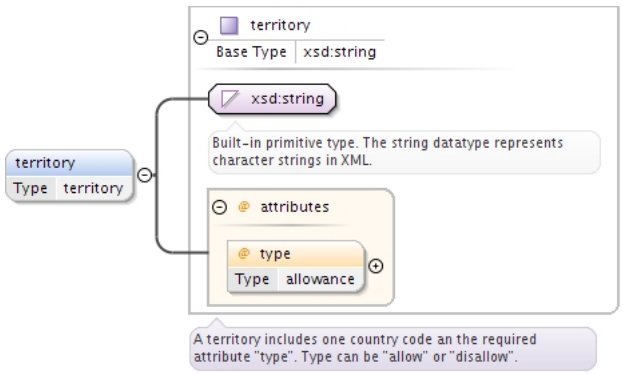
Namespace	No namespace
Diagram	
Type	territorial
Properties	content: complex

	minOccurs: 0 maxOccurs: 1
Model	territory*
Children	territory

Instance	`<territorial>`
	`  <territory type="">{0,unbounded}</territory>`
	`</territorial>`

Source	`<xsd:element name="territorial" type="territorial" maxOccurs="1" minOccurs="0"/>`

Element territorial / territory

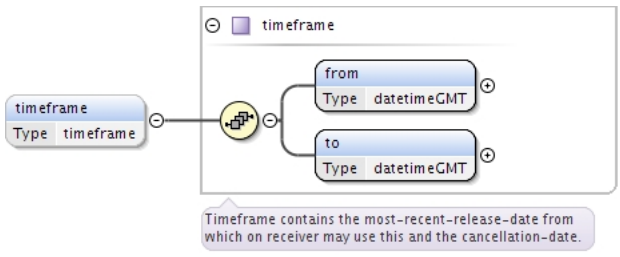
Namespace	No namespace
Diagram	
Type	territory
Properties	content: complex minOccurs: 0 maxOccurs: unbounded

Attributes

QName	Type	Fixed	Default	Use
type	allowance			optional

Source	`<xsd:element name="territory" type="territory" maxOccurs="unbounded" minOccurs="0"/>`

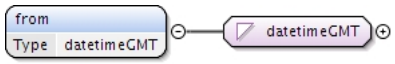
Element license_basis / timeframe

Namespace	No namespace
Diagram	
Type	timeframe
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	from , to
Children	from, to

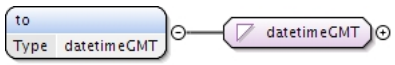
Instance	`<timeframe>`
	`  <from>{1,1}</from>`
	`  <to>{1,1}</to>`
	`</timeframe>`

Source	`<xsd:element name="timeframe" type="timeframe" maxOccurs="1" minOccurs="0"/>`

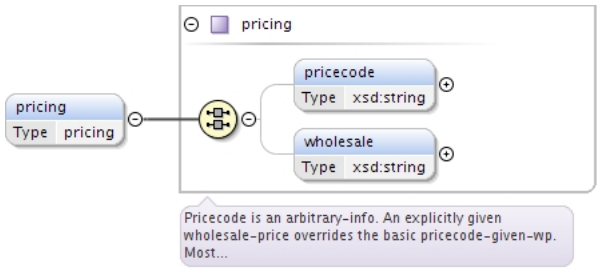
Element `timeframe / from`

Namespace	No namespace
Diagram	
Type	datetimeGMT
Properties	content: `simple`
Facets	pattern `\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}`
Source	`<xsd:element name="from" type="datetimeGMT"/>`

Element `timeframe / to`

Namespace	No namespace
Diagram	
Type	datetimeGMT
Properties	content: `simple`
Facets	pattern `\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}`
Source	`<xsd:element name="to" type="datetimeGMT"/>`

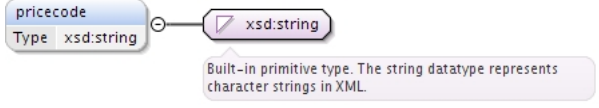
Element `license_basis / pricing`

Namespace	No namespace
Diagram	
Type	pricing
Properties	content: `complex` minOccurs: `0` maxOccurs: `1`
Model	ALL(pricecode{0,1} wholesale{0,1})
Children	pricecode, wholesale

Instance	`<pricing>`
	`  <pricecode>{0,1}</pricecode>`
	`  <wholesale>{0,1}</wholesale>`
	`</pricing>`

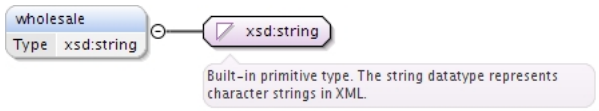
Source	`<xsd:element name="pricing" type="pricing" maxOccurs="1" minOccurs="0"/>`

Element `pricing / pricecode`

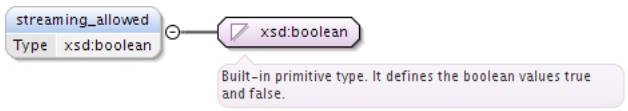
Namespace	No namespace
Diagram	

Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="pricecode" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

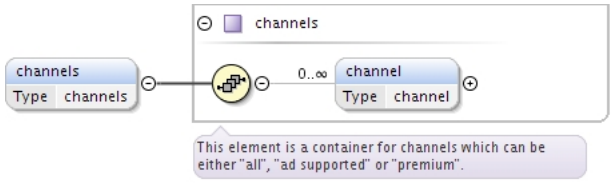
Element pricing / wholesale

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="wholesale" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element license_basis / streaming_allowed

Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="streaming_allowed" type="xsd:boolean" maxOccurs="1" minOccurs="0"/>`

Element license_basis / channels

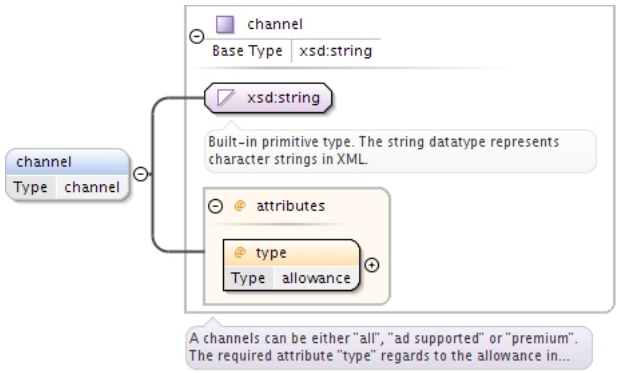
Namespace	No namespace
Diagram	
Type	channels
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	channel*
Children	channel

Instance	`<channels>`
	`  <channel type="">{0,unbounded}</channel>`
	`</channels>`

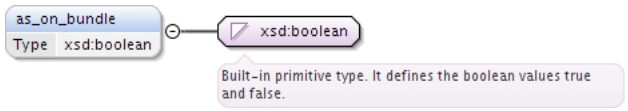
Source	`<xsd:element name="channels" type="channels" maxOccurs="1" minOccurs="0"/>`

Element channels / channel

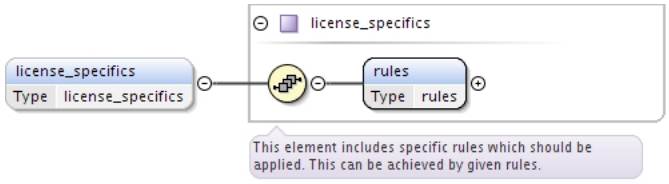
Namespace	No namespace

Diagram	
Type	channel
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **type** | allowance | | | required
Source	`<xsd:element name="channel" type="channel" maxOccurs="unbounded" minOccurs="0"/>`

Element license_basis / as_on_bundle

Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="as_on_bundle" type="xsd:boolean" maxOccurs="1" minOccurs="0"/>`

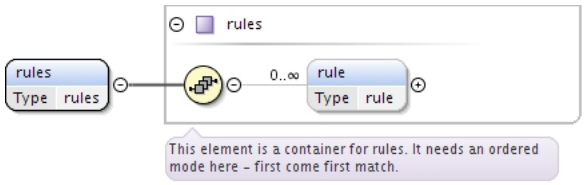
Element bundle / license_specifics

Namespace	No namespace
Diagram	
Type	license_specifics
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	rules
Children	rules

Instance	`<license_specifics>`
	`  <rules>{1,1}</rules>`
	`</license_specifics>`

Source	`<xsd:element name="license_specifics" type="license_specifics" maxOccurs="1" minOccurs="0"/>`

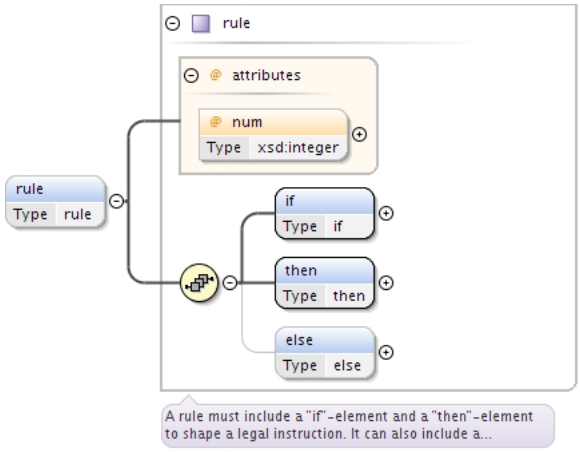
Element `license_specifics / rules`

Namespace	No namespace
Diagram	
Type	rules
Properties	content: complex
Model	rule*
Children	rule

Instance	`<rules>`
	`  <rule num="">{0,unbounded}</rule>`
	`</rules>`

Source	`<xsd:element name="rules" type="rules"/>`

Element `rules / rule`

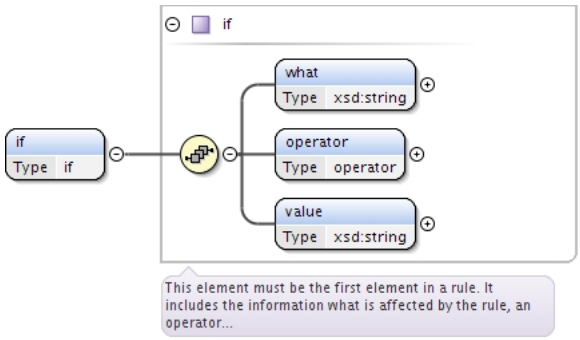
Namespace	No namespace
Diagram	
Type	rule
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	if , then , else{0,1}
Children	else, if, then

Instance	`<rule num="">`
	`  <if>{1,1}</if>`
	`  <then>{1,1}</then>`
	`  <else>{0,1}</else>`
	`</rule>`

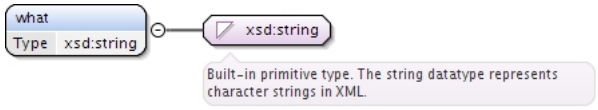
Attributes	**QName** num; **Type** xsd:integer; **Fixed**; **Default**; **Use** optional
Source	`<xsd:element name="rule" type="rule" maxOccurs="unbounded" minOccurs="0"/>`

Element `rule / if`

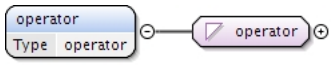
Namespace	No namespace

Diagram	
Type	if
Properties	content: complex
Model	what , operator , value
Children	operator, value, what
Instance	<if> <what>{1,1}</what> <operator>{1,1}</operator> <value>{1,1}</value> </if>
Source	<xsd:element name="if" type="if"/>

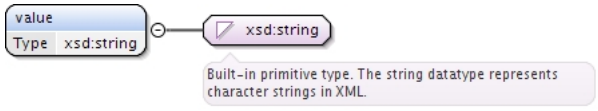
Element if / what

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="what" type="xsd:string"/>

Element if / operator

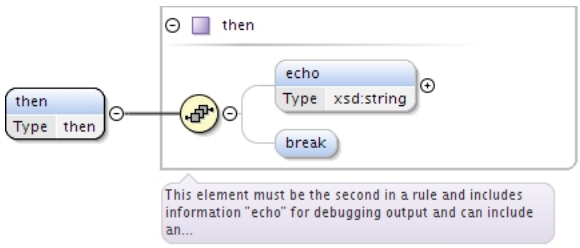
Namespace	No namespace
Diagram	
Type	operator
Properties	content: simple
Facets	enumeration equals enumeration before enumeration after enumeration contains enumeration containedin
Source	<xsd:element name="operator" type="operator"/>

Element if / value

Namespace	No namespace
Diagram	
Type	xsd:string

Properties	content: simple
Source	`<xsd:element name="value" type="xsd:string"/>`

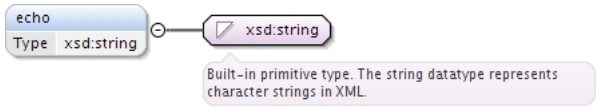
Element rule / then

Namespace	No namespace
Diagram	
Type	then
Properties	content: complex
Model	echo{0,1} , break{0,1}
Children	break, echo

Instance	`<then>`
	`  <echo>{0,1}</echo>`
	`  <break>{0,1}</break>`
	`</then>`

Source	`<xsd:element name="then" type="then"/>`

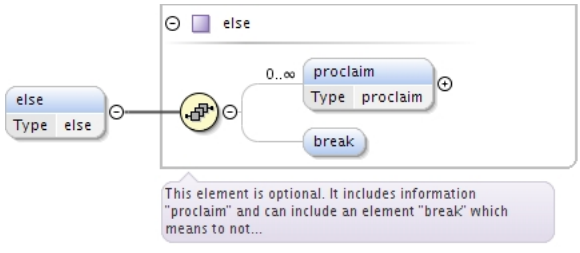
Element then / echo

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="echo" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element then / break

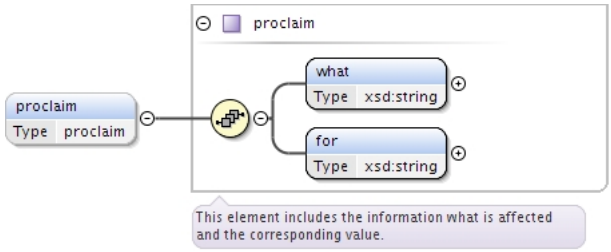
Namespace	No namespace
Diagram	
Properties	minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="break" maxOccurs="1" minOccurs="0"/>`

Element rule / else

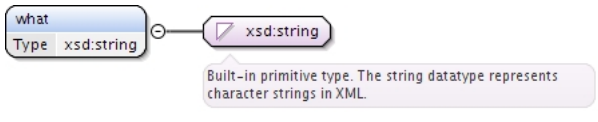
Namespace	No namespace
Diagram	

Type	else
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	proclaim* , break{0,1}
Children	break, proclaim
Instance	<else> <proclaim>{0,unbounded}</proclaim> <break>{0,1}</break> </else>
Source	<xsd:element name="else" type="else" maxOccurs="1" minOccurs="0"/>

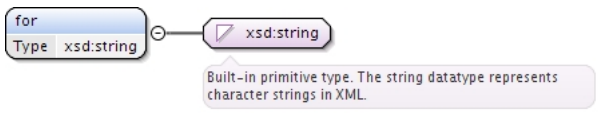
Element else / proclaim

Namespace	No namespace
Diagram	
Type	proclaim
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	what , for
Children	for, what
Instance	<proclaim> <what>{1,1}</what> <for>{1,1}</for> </proclaim>
Source	<xsd:element name="proclaim" type="proclaim" maxOccurs="unbounded" minOccurs="0"/>

Element proclaim / what

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="what" type="xsd:string"/>

Element proclaim / for

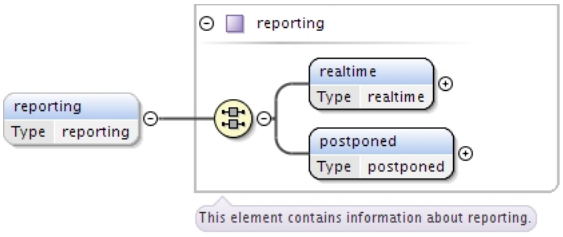
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple

Source	`<xsd:element name="for" type="xsd:string"/>`

Element `else / break`

Namespace	No namespace
Diagram	
Properties	minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="break" maxOccurs="1" minOccurs="0"/>`

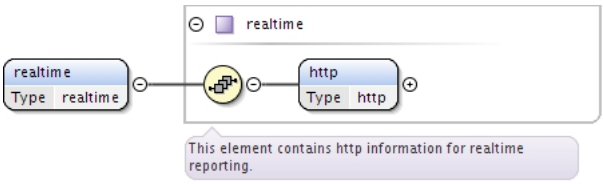
Element `bundle / reporting`

Namespace	No namespace
Diagram	
Type	reporting
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(realtime postponed)
Children	postponed, realtime

Instance	`<reporting>`
	`  <realtime>{1,1}</realtime>`
	`  <postponed>{1,1}</postponed>`
	`</reporting>`

Source	`<xsd:element name="reporting" type="reporting" maxOccurs="1" minOccurs="0"/>`

Element `reporting / realtime`

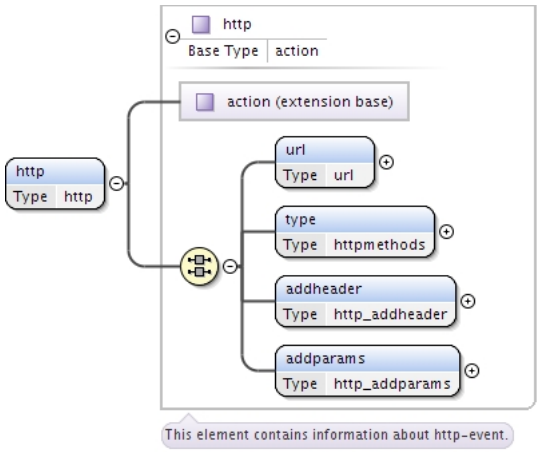
Namespace	No namespace
Diagram	
Type	realtime
Properties	content: complex
Model	http
Children	http

Instance	`<realtime>`
	`  <http>{1,1}</http>`
	`</realtime>`

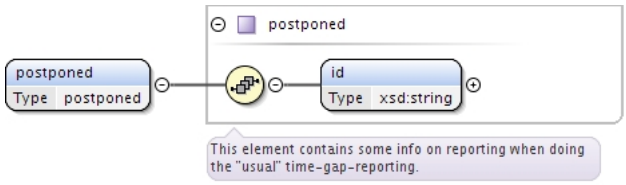
Source	`<xsd:element name="realtime" type="realtime"/>`

Element `realtime / http`

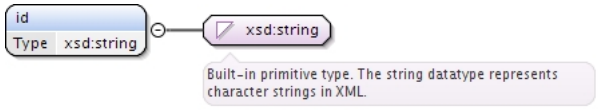
Namespace	No namespace

Diagram	
Type	http
Type hierarchy	• action • http
Properties	content: complex
Model	ALL(url type addheader addparams)
Children	addheader, addparams, type, url
Instance	<http> <url>{1,1}</url> <type>{1,1}</type> <addheader>{1,1}</addheader> <addparams>{1,1}</addparams> </http>
Source	<xsd:element name="http" type="http"/>

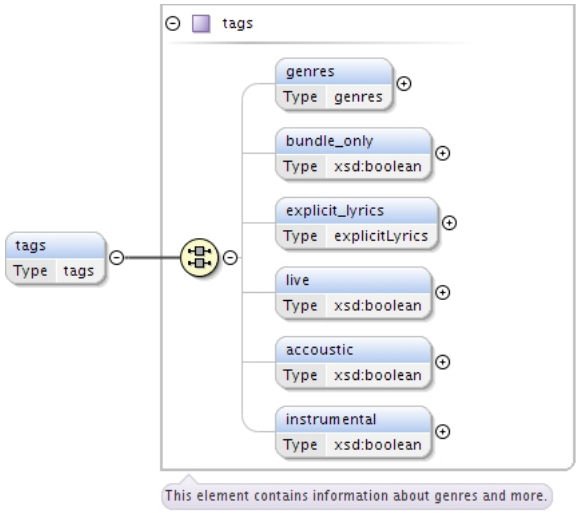
Element reporting / postponed

Namespace	No namespace
Diagram	
Type	postponed
Properties	content: complex
Model	id
Children	id
Instance	<postponed> <id>{1,1}</id> </postponed>
Source	<xsd:element name="postponed" type="postponed"/>

Element postponed / id

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="id" type="xsd:string"/>

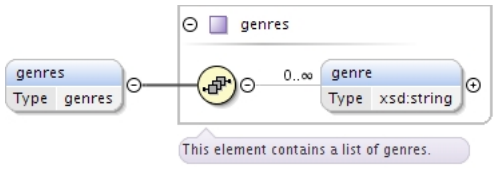
Element `bundle / tags`

Namespace	No namespace
Diagram	
Type	tags
Properties	content: `complex` minOccurs: `0` maxOccurs: `1`
Model	ALL(genres{0,1} bundle_only{0,1} explicit_lyrics{0,1} live{0,1} accoustic{0,1} instrumental{0,1})
Children	accoustic, bundle_only, explicit_lyrics, genres, instrumental, live

`Instance`	`<tags>`
	`  <genres>{0,1}</genres>`
	`  <bundle_only>{0,1}</bundle_only>`
	`  <explicit_lyrics>{0,1}</explicit_lyrics>`
	`  <live>{0,1}</live>`
	`  <accoustic>{0,1}</accoustic>`
	`  <instrumental>{0,1}</instrumental>`
	`</tags>`

`Source`	`<xsd:element name="tags" type="tags" maxOccurs="1" minOccurs="0"/>`

Element `tags / genres`

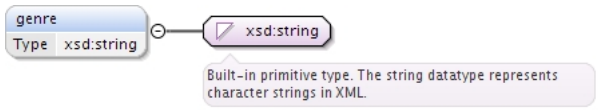
Namespace	No namespace
Diagram	
Type	genres
Properties	content: `complex` minOccurs: `0` maxOccurs: `1`
Model	genre*
Children	genre

`Instance`	`<genres>`
	`  <genre>{0,unbounded}</genre>`
	`</genres>`

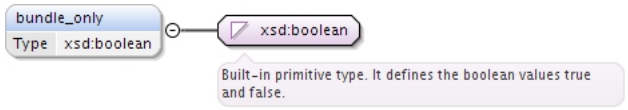
`Source`	`<xsd:element name="genres" type="genres" maxOccurs="1" minOccurs="0"/>`

Element `genres / genre`

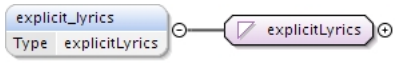
Namespace	No namespace

Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: unbounded
Source	`<xsd:element name="genre" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>`

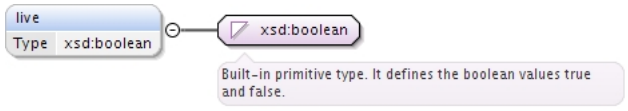
Element tags / bundle_only

Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="bundle_only" type="xsd:boolean" maxOccurs="1" minOccurs="0"/>`

Element tags / explicit_lyrics

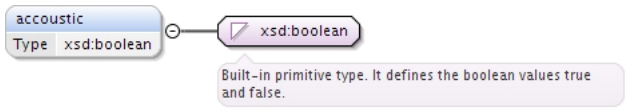
Namespace	No namespace
Diagram	
Type	explicitLyrics
Properties	content: simple minOccurs: 0 maxOccurs: 1
Facets	enumeration true enumeration false enumeration cleaned
Source	`<xsd:element name="explicit_lyrics" type="explicitLyrics" maxOccurs="1" minOccurs="0"/>`

Element tags / live

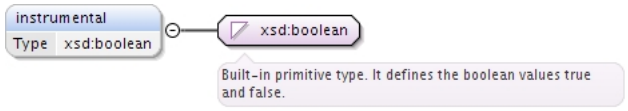
Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="live" type="xsd:boolean" maxOccurs="1" minOccurs="0"/>`

Element tags / accoustic

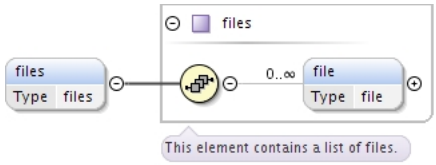
Namespace	No namespace

Diagram	
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="accoustic" type="xsd:boolean" maxOccurs="1" minOccurs="0"/>`

Element tags / instrumental

Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="instrumental" type="xsd:boolean" maxOccurs="1" minOccurs="0"/>`

Element bundle / files

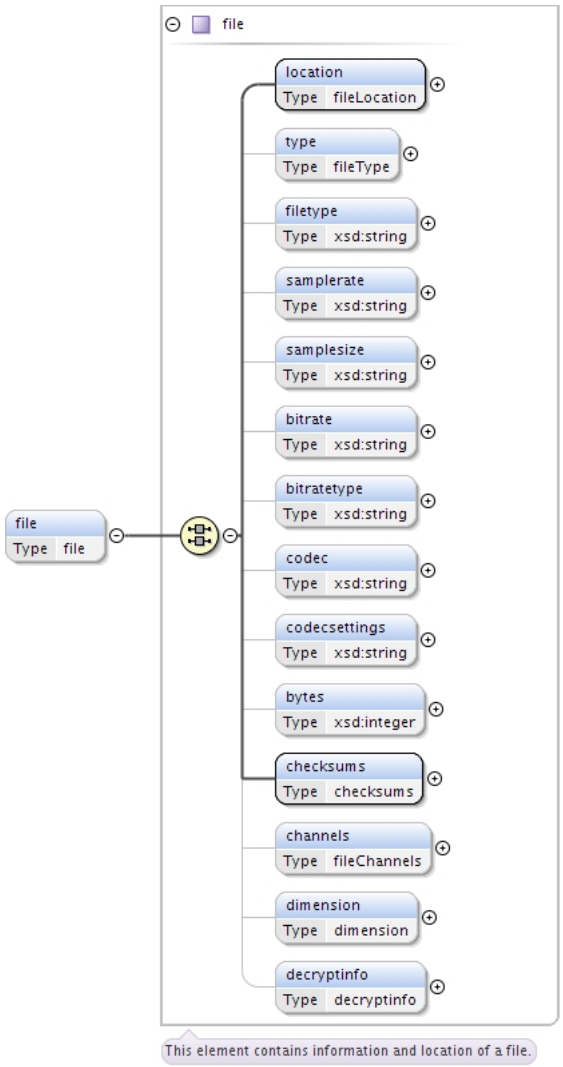
Namespace	No namespace
Diagram	
Type	files
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	file*
Children	file

Instance	`<files>`
	`  <file>{0,unbounded}</file>`
	`</files>`

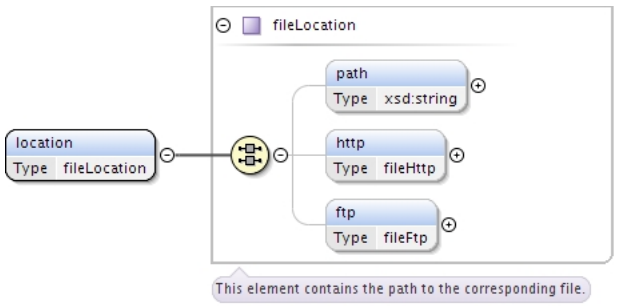
Source	`<xsd:element name="files" type="files" maxOccurs="1" minOccurs="0"/>`

Element files / file

Namespace	No namespace

Diagram	
Type	file
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(location type{0,1} filetype{0,1} samplerate{0,1} samplesize{0,1} bitrate{0,1} bitratetype{0,1} codec{0,1} codecsettings{0,1} bytes{0,1} checksums channels{0,1} dimension{0,1} decryptinfo{0,1})
Children	bitrate, bitratetype, bytes, channels, checksums, codec, codecsettings, decryptinfo, dimension, filetype, location, samplerate, samplesize, type
Instance	<file> <location>{1,1}</location> <type>{0,1}</type> <filetype>{0,1}</filetype> <samplerate>{0,1}</samplerate> <samplesize>{0,1}</samplesize> <bitrate>{0,1}</bitrate> <bitratetype>{0,1}</bitratetype> <codec>{0,1}</codec> <codecsettings>{0,1}</codecsettings> <bytes>{0,1}</bytes> <checksums>{1,1}</checksums> <channels>{0,1}</channels> <dimension>{0,1}</dimension> <decryptinfo>{0,1}</decryptinfo> </file>
Source	<xsd:element name="file" type="file" maxOccurs="unbounded" minOccurs="0"/>

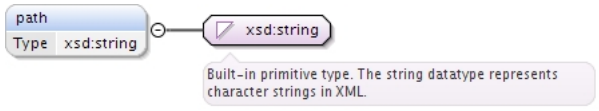
Element `file / location`

Namespace	No namespace
Diagram	
Type	fileLocation
Properties	content: `complex`
Model	ALL(path{0,1} http{0,1} ftp{0,1})
Children	ftp, http, path

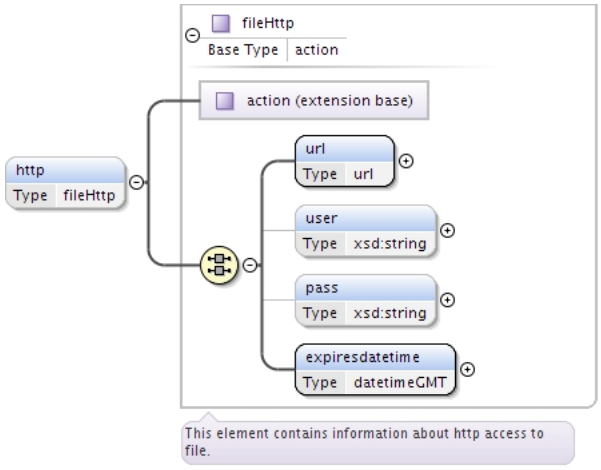
`Instance`	`<location>`
	`  <path>{0,1}</path>`
	`  <http>{0,1}</http>`
	`  <ftp>{0,1}</ftp>`
	`</location>`

`Source`	`<xsd:element name="location" type="fileLocation"/>`

Element `fileLocation / path`

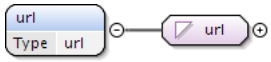
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: `simple` minOccurs: `0` maxOccurs: `1`
`Source`	`<xsd:element name="path" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element `fileLocation / http`

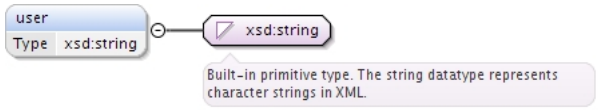
Namespace	No namespace
Diagram	
Type	fileHttp
Type hierarchy	• action

	• fileHttp
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(url user{0,1} pass{0,1} expiresdatetime)
Children	expiresdatetime, pass, url, user
Instance	<http> <url>{1,1}</url> <user>{0,1}</user> <pass>{0,1}</pass> <expiresdatetime>{1,1}</expiresdatetime> </http>
Source	<xsd:element name="http" type="fileHttp" maxOccurs="1" minOccurs="0"/>

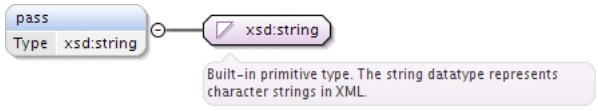
Element fileHttp / url

Namespace	No namespace
Diagram	
Type	url
Properties	content: simple
Source	<xsd:element name="url" type="url"/>

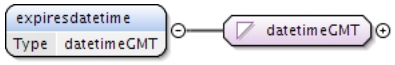
Element fileHttp / user

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="user" type="xsd:string" maxOccurs="1" minOccurs="0"/>

Element fileHttp / pass

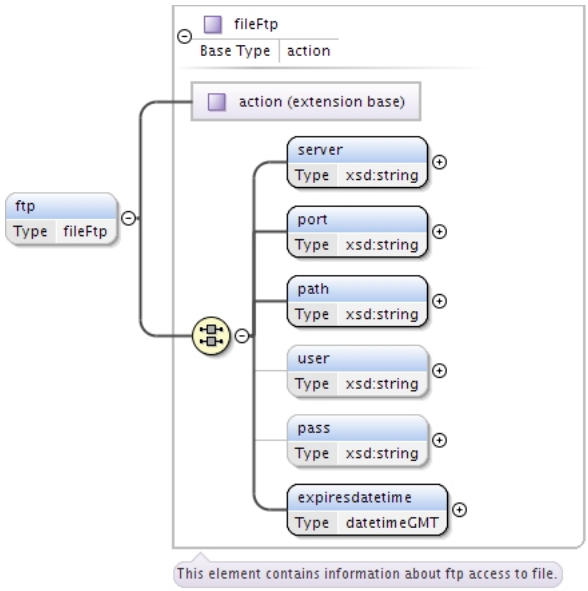
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="pass" type="xsd:string" maxOccurs="1" minOccurs="0"/>

Element fileHttp / expiresdatetime

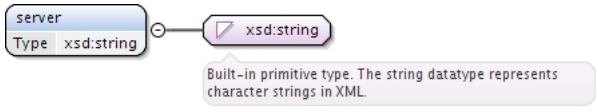
Namespace	No namespace
Diagram	

Type	datetimeGMT
Properties	content: simple
Facets	pattern \d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}
Source	<xsd:element name="expiresdatetime" type="datetimeGMT"/>

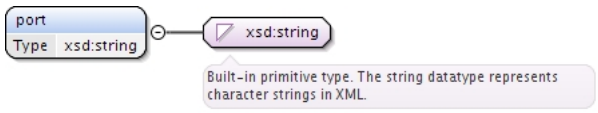
Element fileLocation / ftp

Namespace	No namespace
Diagram	
Type	fileFtp
Type hierarchy	• action • fileFtp
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(server port path user{0,1} pass{0,1} expiresdatetime)
Children	expiresdatetime, pass, path, port, server, user
Instance	<ftp> <server>{1,1}</server> <port>{1,1}</port> <path>{1,1}</path> <user>{0,1}</user> <pass>{0,1}</pass> <expiresdatetime>{1,1}</expiresdatetime> </ftp>
Source	<xsd:element name="ftp" type="fileFtp" maxOccurs="1" minOccurs="0"/>

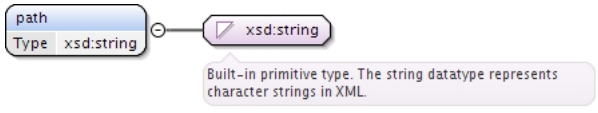
Element fileFtp / server

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="server" type="xsd:string"/>

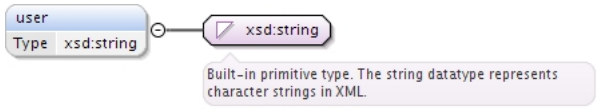
Element `fileFtp / port`

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="port" type="xsd:string"/>`

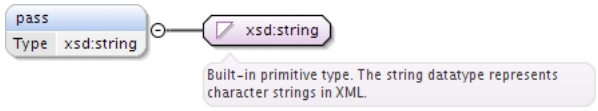
Element `fileFtp / path`

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="path" type="xsd:string"/>`

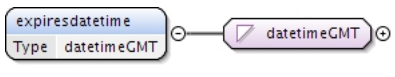
Element `fileFtp / user`

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="user" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element `fileFtp / pass`

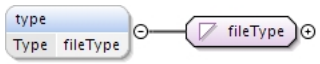
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="pass" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element `fileFtp / expiresdatetime`

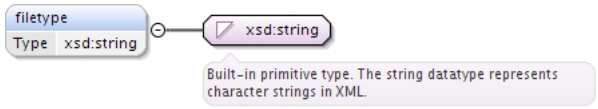
Namespace	No namespace
Diagram	
Type	datetimeGMT
Properties	content: simple

Facets	pattern	\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}
Source	<xsd:element name="expiresdatetime" type="datetimeGMT"/>	

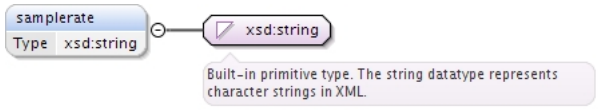
Element file / type

Namespace	No namespace	
Diagram		
Type	fileType	
Properties	content:	simple
	minOccurs:	0
	maxOccurs:	1
Facets	enumeration	full
	enumeration	prelistening
	enumeration	cover
	enumeration	booklet
Source	<xsd:element name="type" type="fileType" maxOccurs="1" minOccurs="0"/>	

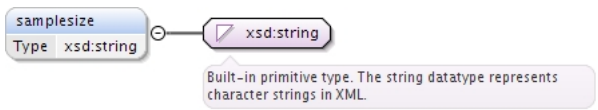
Element file / filetype

Namespace	No namespace	
Diagram		
Type	xsd:string	
Properties	content:	simple
	minOccurs:	0
	maxOccurs:	1
Source	<xsd:element name="filetype" type="xsd:string" maxOccurs="1" minOccurs="0"/>	

Element file / samplerate

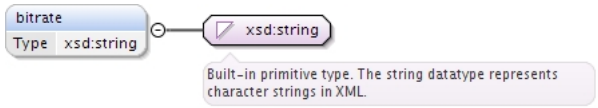
Namespace	No namespace	
Diagram		
Type	xsd:string	
Properties	content:	simple
	minOccurs:	0
	maxOccurs:	1
Source	<xsd:element name="samplerate" type="xsd:string" maxOccurs="1" minOccurs="0"/>	

Element file / samplesize

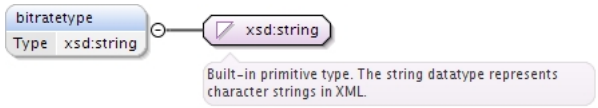
Namespace	No namespace	
Diagram		
Type	xsd:string	
Properties	content:	simple

	minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="samplesize" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

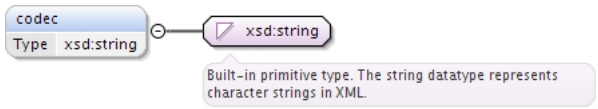
Element file / bitrate

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="bitrate" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

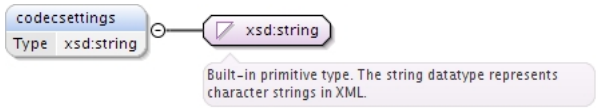
Element file / bitratetype

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="bitratetype" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

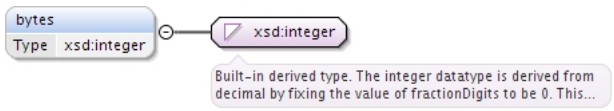
Element file / codec

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="codec" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

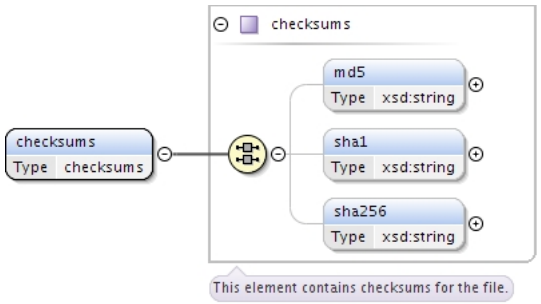
Element file / codecsettings

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="codecsettings" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element `file / bytes`

Namespace	No namespace
Diagram	
Type	xsd:integer
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="bytes" type="xsd:integer" maxOccurs="1" minOccurs="0"/>`

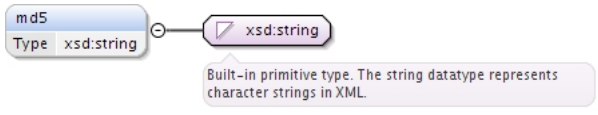
Element `file / checksums`

Namespace	No namespace
Diagram	
Type	checksums
Properties	content: complex
Model	ALL(md5{0,1} sha1{0,1} sha256{0,1})
Children	md5, sha1, sha256

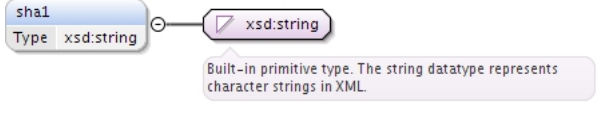
Instance	`<checksums>`
	`  <md5>{0,1}</md5>`
	`  <sha1>{0,1}</sha1>`
	`  <sha256>{0,1}</sha256>`
	`</checksums>`

Source	`<xsd:element name="checksums" type="checksums"/>`

Element `checksums / md5`

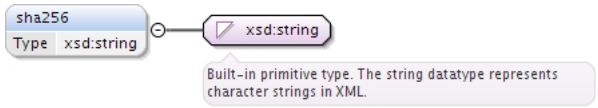
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="md5" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element `checksums / sha1`

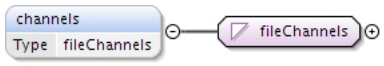
Namespace	No namespace
Diagram	

Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="sha1" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

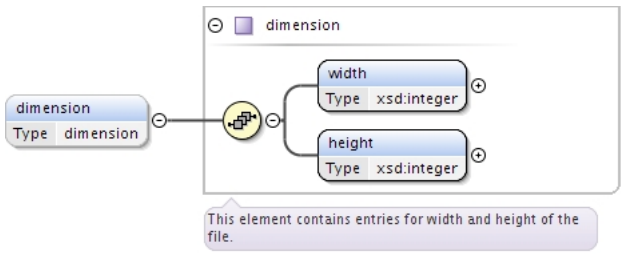
Element checksums / sha256

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="sha256" type="xsd:string" maxOccurs="1" minOccurs="0"/>`

Element file / channels

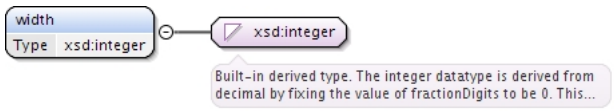
Namespace	No namespace
Diagram	
Type	fileChannels
Properties	content: simple minOccurs: 0 maxOccurs: 1
Facets	enumeration mono enumeration stereo enumeration joint-stereo enumeration 5.1
Source	`<xsd:element name="channels" type="fileChannels" maxOccurs="1" minOccurs="0"/>`

Element file / dimension

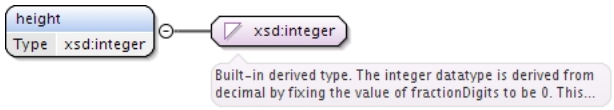
Namespace	No namespace
Diagram	
Type	dimension
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	width , height
Children	height, width
Instance	`<dimension>` `  <width>{1,1}</width>`

	<height>{1,1}</height> </dimension>
Source	<xsd:element name="dimension" type="dimension" maxOccurs="1" minOccurs="0"/>

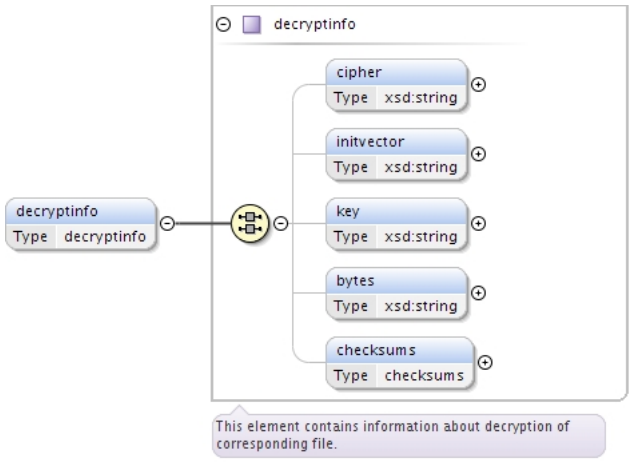
Element dimension / width

Namespace	No namespace
Diagram	
Type	xsd:integer
Properties	content: simple
Source	<xsd:element name="width" type="xsd:integer"/>

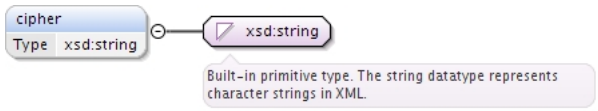
Element dimension / height

Namespace	No namespace
Diagram	
Type	xsd:integer
Properties	content: simple
Source	<xsd:element name="height" type="xsd:integer"/>

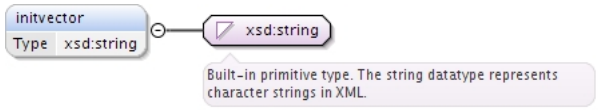
Element file / decryptinfo

Namespace	No namespace
Diagram	
Type	decryptinfo
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(cipher{0,1} initvector{0,1} key{0,1} bytes{0,1} checksums{0,1})
Children	bytes, checksums, cipher, initvector, key
Instance	<decryptinfo> <cipher>{0,1}</cipher> <initvector>{0,1}</initvector> <key>{0,1}</key> <bytes>{0,1}</bytes> <checksums>{0,1}</checksums> </decryptinfo>
Source	<xsd:element name="decryptinfo" type="decryptinfo" maxOccurs="1" minOccurs="0"/>

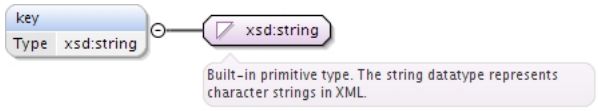
Element decryptinfo / cipher

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="cipher" type="xsd:string" minOccurs="0" maxOccurs="1"/>`

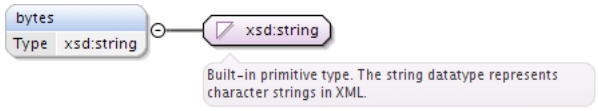
Element decryptinfo / initvector

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="initvector" type="xsd:string" minOccurs="0" maxOccurs="1"/>`

Element decryptinfo / key

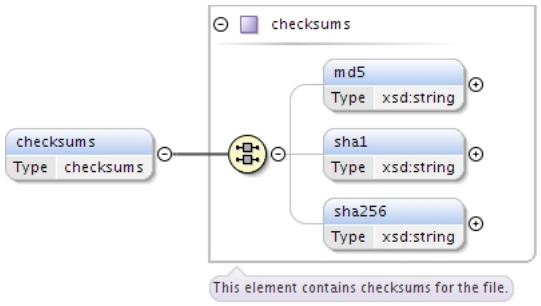
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="key" type="xsd:string" minOccurs="0" maxOccurs="1"/>`

Element decryptinfo / bytes

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	`<xsd:element name="bytes" type="xsd:string" minOccurs="0" maxOccurs="1"/>`

Element decryptinfo / checksums

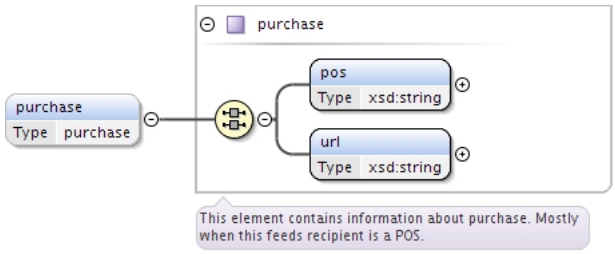
Namespace	No namespace

Diagram	
Type	checksums
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(md5{0,1} sha1{0,1} sha256{0,1})
Children	md5, sha1, sha256

Instance	`<checksums>`
	`  <md5>{0,1}</md5>`
	`  <sha1>{0,1}</sha1>`
	`  <sha256>{0,1}</sha256>`
	`</checksums>`

Source	`<xsd:element name="checksums" type="checksums" minOccurs="0" maxOccurs="1"/>`

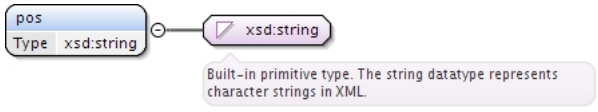
Element bundle / purchase

Namespace	No namespace
Diagram	
Type	purchase
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(pos url)
Children	pos, url

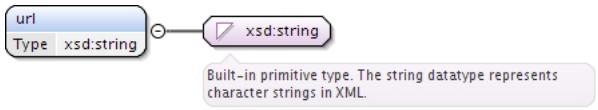
Instance	`<purchase>`
	`  <pos>{1,1}</pos>`
	`  <url>{1,1}</url>`
	`</purchase>`

Source	`<xsd:element name="purchase" type="purchase" maxOccurs="1" minOccurs="0"/>`

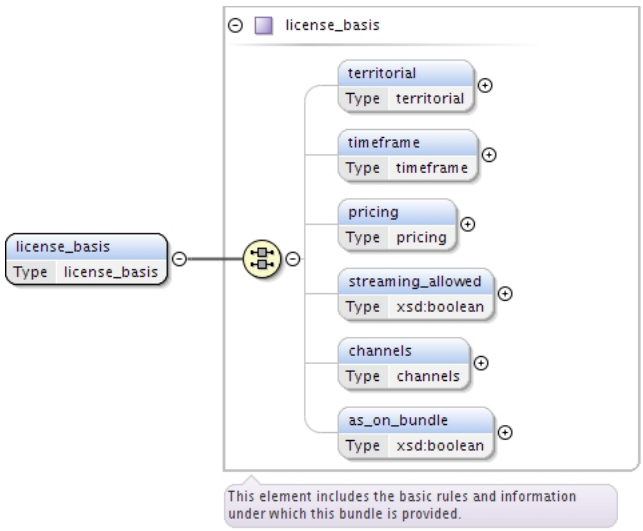
Element purchase / pos

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	`<xsd:element name="pos" type="xsd:string"/>`

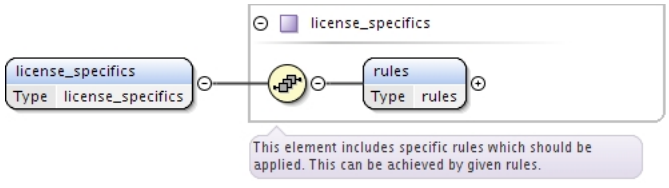
Element purchase / url

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="url" type="xsd:string"/>

Element item / license_basis

Namespace	No namespace
Diagram	
Type	license_basis
Properties	content: complex
Model	ALL(territorial{0,1} timeframe{0,1} pricing{0,1} streaming_allowed{0,1} channels{0,1} as_on_bundle{0,1})
Children	as_on_bundle, channels, pricing, streaming_allowed, territorial, timeframe
Instance	<license_basis> <territorial>{0,1}</territorial> <timeframe>{0,1}</timeframe> <pricing>{0,1}</pricing> <streaming_allowed>{0,1}</streaming_allowed> <channels>{0,1}</channels> <as_on_bundle>{0,1}</as_on_bundle> </license_basis>
Source	<xsd:element name="license_basis" type="license_basis"/>

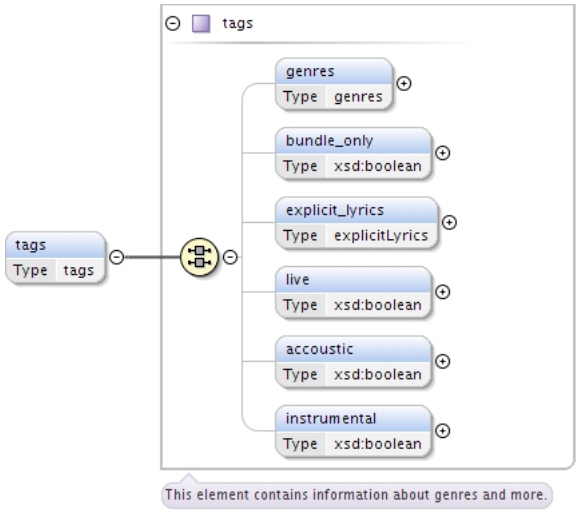
Element item / license_specifics

Namespace	No namespace
Diagram	
Type	license_specifics
Properties	content: complex
Model	rules
Children	rules

Instance	<pre><license_specifics>
	  <rules>{1,1}</rules>
	</license_specifics></pre>

Source	<pre><xsd:element name="license_specifics" type="license_specifics"/></pre>

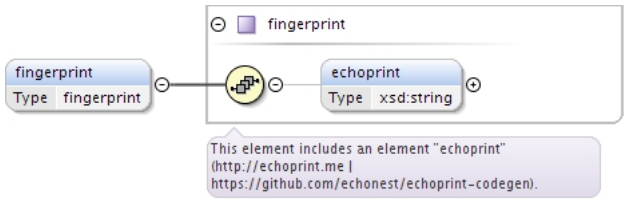
Element item / tags

Namespace	No namespace
Diagram	
Type	tags
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(genres{0,1} bundle_only{0,1} explicit_lyrics{0,1} live{0,1} accoustic{0,1} instrumental{0,1})
Children	accoustic, bundle_only, explicit_lyrics, genres, instrumental, live

Instance	<pre><tags>
	  <genres>{0,1}</genres>
	  <bundle_only>{0,1}</bundle_only>
	  <explicit_lyrics>{0,1}</explicit_lyrics>
	  <live>{0,1}</live>
	  <accoustic>{0,1}</accoustic>
	  <instrumental>{0,1}</instrumental>
	</tags></pre>

Source	<pre><xsd:element name="tags" type="tags" maxOccurs="1" minOccurs="0"/></pre>

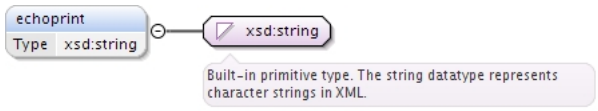
Element item / fingerprint

Namespace	No namespace
Diagram	
Type	fingerprint
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	echoprint{0,1}
Children	echoprint

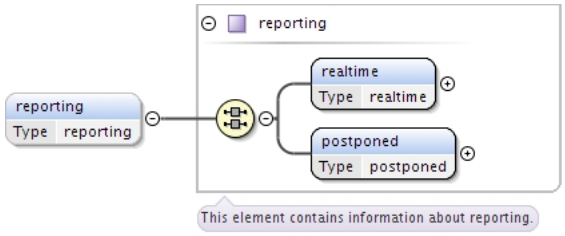
Instance	<pre><fingerprint>
	  <echoprint>{0,1}</echoprint>
	</fingerprint></pre>

Source	<xsd:element name="fingerprint" type="fingerprint" maxOccurs="1" minOccurs="0"/>

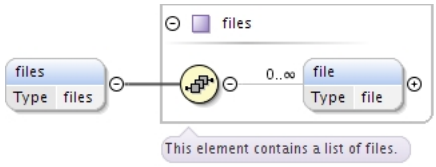
Element fingerprint / echoprint

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="echoprint" type="xsd:string" maxOccurs="1" minOccurs="0"/>

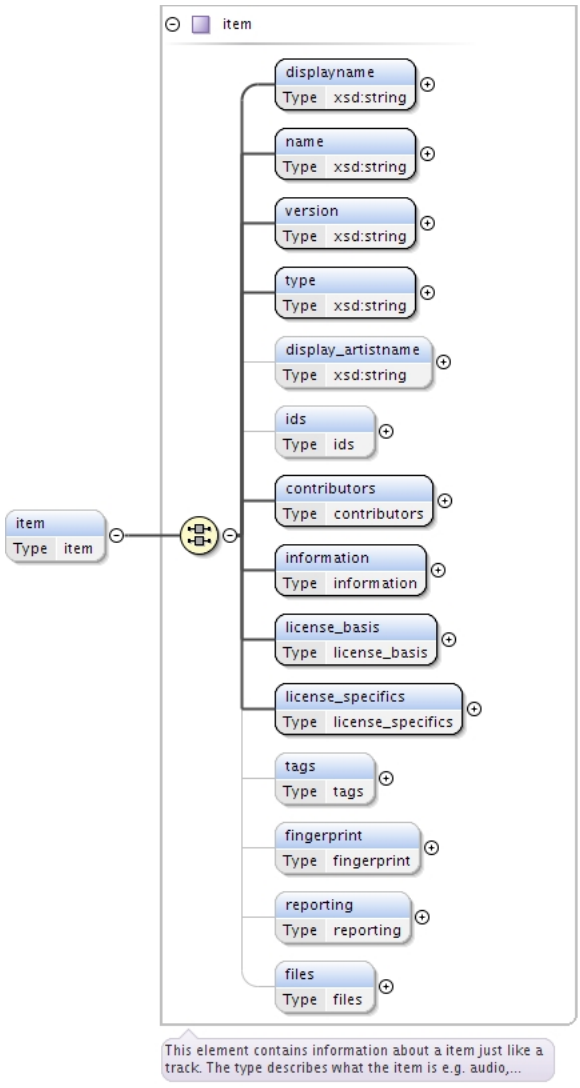
Element item / reporting

Namespace	No namespace
Diagram	
Type	reporting
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(realtime postponed)
Children	postponed, realtime
Instance	<reporting> <realtime>{1,1}</realtime> <postponed>{1,1}</postponed> </reporting>
Source	<xsd:element name="reporting" type="reporting" maxOccurs="1" minOccurs="0"/>

Element item / files

Namespace	No namespace
Diagram	
Type	files
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	file*
Children	file
Instance	<files> <file>{0,unbounded}</file> </files>
Source	<xsd:element name="files" type="files" maxOccurs="1" minOccurs="0"/>

Element `feed / item`

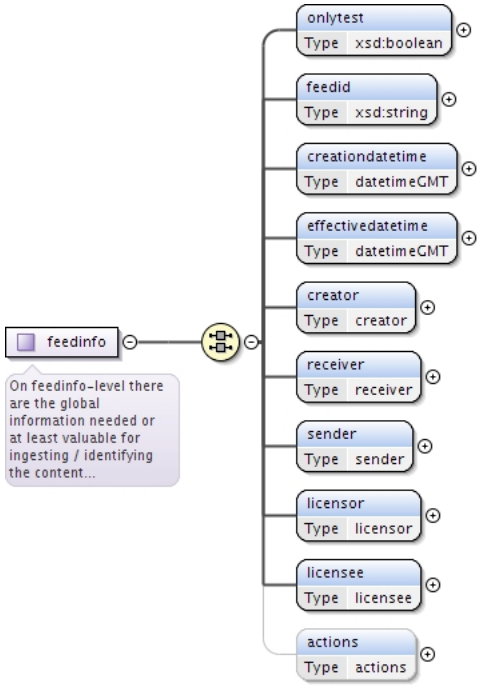
Namespace	No namespace
Diagram	
Type	item
Properties	content: `complex` minOccurs: `0` maxOccurs: `unbounded`
Model	ALL(displayname name version type display_artistname{0,1} ids{0,1} contributors information license_basis license_specifics tags{0,1} fingerprint{0,1} reporting{0,1} files{0,1})
Children	contributors, display_artistname, displayname, files, fingerprint, ids, information, license_basis, license_specifics, name, reporting, tags, type, version

`Instance`	`<item>`
	`  <displayname>{1,1}</displayname>`
	`  <name>{1,1}</name>`
	`  <version>{1,1}</version>`
	`  <type>{1,1}</type>`
	`  <display_artistname>{0,1}</display_artistname>`
	`  <ids>{0,1}</ids>`
	`  <contributors>{1,1}</contributors>`
	`  <information>{1,1}</information>`
	`  <license_basis>{1,1}</license_basis>`
	`  <license_specifics>{1,1}</license_specifics>`
	`  <tags>{0,1}</tags>`
	`  <fingerprint>{0,1}</fingerprint>`
	`  <reporting>{0,1}</reporting>`
	`  <files>{0,1}</files>`
	`</item>`

Source	<xsd:element name="item" type="item" maxOccurs="unbounded" minOccurs="0"/>

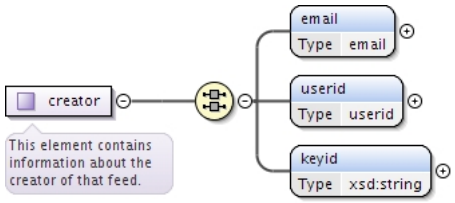
Complex Type(s)

Complex Type `feedinfo`

Namespace	No namespace
Annotations	On feedinfo-level there are the global information needed or at least valuable for ingesting / identifying the content sent. It is defined, when the feed was created, when it shall be come effective, who created the feed and who is the receiver of the feed. Also the sender (which can diverge from the creator) is to be stated. The licensor is also to be stated (which in turn can also diverge from the creator and/or the sender). There can be "actions" defined on the receiving-party's side which should be "done" when initially receiving this feed, or starting to process the feed for ingestion or finishing the feeds processing. Additionally when everything could be interpreted correctly (in the sense of the receiving party), a "full-success-action" could be issued; likewise if "some error" occurred while processing the feed, an "onerror-action" could be issued. Those actions are initially defined to be email-notifications or http-calls; we also included some action to have a "registered letter" and/or "fax" to be sent; wether this is accepted/handled by the receiving party is to be dealt with contractually (we included a field for stating how much the sending party will cover the fee max.).
Diagram	
Used by	Element feed/feedinfo
Model	ALL(onlytest feedid creationdatetime effectivedatetime creator receiver sender licensor licensee actions{0,1})
Children	actions, creationdatetime, creator, effectivedatetime, feedid, licensee, licensor, onlytest, receiver, sender
Source	<xsd:complexType name="feedinfo"> <xsd:annotation> <xsd:documentation xml:lang="en">On feedinfo-level there are the global information needed or at least valuable for ingesting / identifying the content sent. It is defined, when the feed was created, when it shall be come effective, who created the feed and who is the receiver of the feed. Also the sender (which can diverge from the creator) is to be stated. The licensor is also to be stated (which in turn can also diverge from the creator and/or the sender). There can be "actions" defined on the receiving-party's side which should be "done" when initially receiving this feed, or starting to process the feed for ingestion or finishing the feeds processing. Additionally when everything could be interpreted correctly (in the sense of the receiving party), a "full-success-action" could be issued; likewise if "some error" occurred while processing the feed,

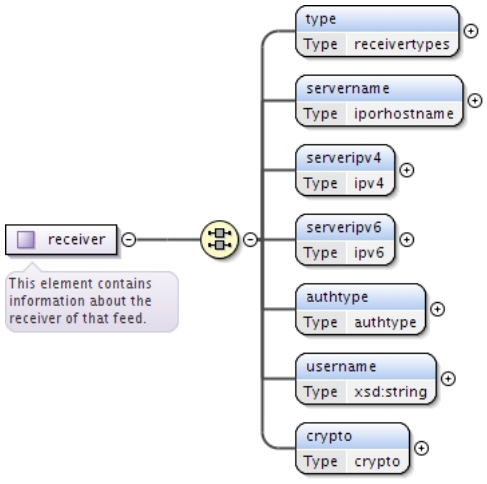
```
 an "onerror-action" could be issued. Those actions are initially defined to be email-
notifications or http-calls; we also included some action to have a "registered letter"
 and/or "fax" to be sent; wether this is accepted/handled by the receiving party is to be
 dealt with contractually (we included a field for stating how much the sending party will
 cover the fee max.).</xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="onlytest" type="xsd:boolean"/>
    <xsd:element name="feedid" type="xsd:string"/>
    <xsd:element name="creationdatetime" type="datetimeGMT"/>
    <xsd:element name="effectivedatetime" type="datetimeGMT"/>
    <xsd:element name="creator" type="creator"/>
    <xsd:element name="receiver" type="receiver"/>
    <xsd:element name="sender" type="sender"/>
    <xsd:element name="licensor" type="licensor"/>
    <xsd:element name="licensee" type="licensee"/>
    <xsd:element name="actions" type="actions" maxOccurs="1" minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

Complex Type `creator`

Namespace	No namespace
Annotations	This element contains information about the creator of that feed.
Diagram	
Used by	Element feedinfo/creator
Model	ALL(email userid keyid)
Children	email, keyid, userid
Source	

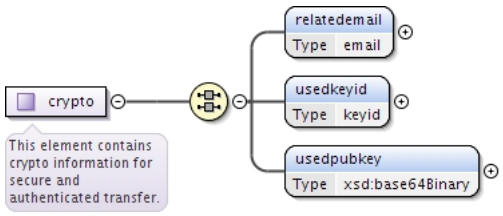
```
<xsd:complexType name="creator">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element contains information about the creator
 of that feed.</xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="email" type="email"/>
    <xsd:element name="userid" type="userid"/>
    <xsd:element name="keyid" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
```

Complex Type `receiver`

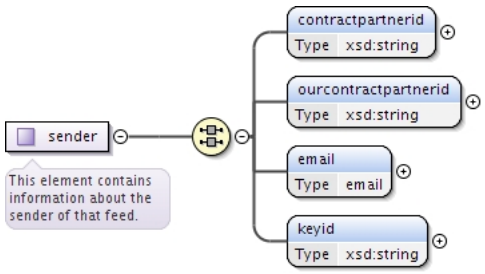
Namespace	No namespace
Annotations	This element contains information about the receiver of that feed.
Diagram	

Used by	Element feedinfo/receiver
Model	ALL(type servername serveripv4 serveripv6 authtype username crypto)
Children	authtype, crypto, serveripv4, serveripv6, servername, type, username
Source	<xsd:complexType name="receiver"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about the receiver of that feed.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="type" type="receivertypes"/> <xsd:element name="servername" type="iporhostname"/> <xsd:element name="serveripv4" type="ipv4"/> <xsd:element name="serveripv6" type="ipv6"/> <xsd:element name="authtype" type="authtype"/> <xsd:element name="username" type="xsd:string"/> <xsd:element name="crypto" type="crypto"/> </xsd:all> </xsd:complexType>

Complex Type crypto

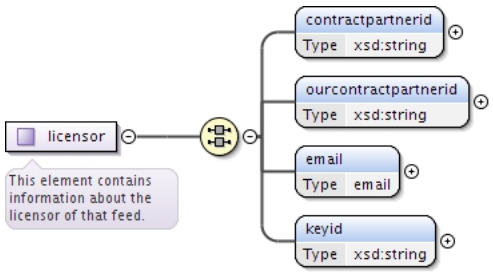
Namespace	No namespace
Annotations	This element contains crypto information for secure and authenticated transfer.
Diagram	
Used by	Element receiver/crypto
Model	ALL(relatedemail usedkeyid usedpubkey)
Children	relatedemail, usedkeyid, usedpubkey
Source	<xsd:complexType name="crypto"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains crypto information for secure and authenticated transfer.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="relatedemail" type="email"/> <xsd:element name="usedkeyid" type="keyid"/> <xsd:element name="usedpubkey" type="xsd:base64Binary"/> </xsd:all> </xsd:complexType>

Complex Type sender

Namespace	No namespace
Annotations	This element contains information about the sender of that feed.
Diagram	
Used by	Element feedinfo/sender
Model	ALL(contractpartnerid ourcontractpartnerid email keyid)
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Source	<xsd:complexType name="sender"> <xsd:annotation>

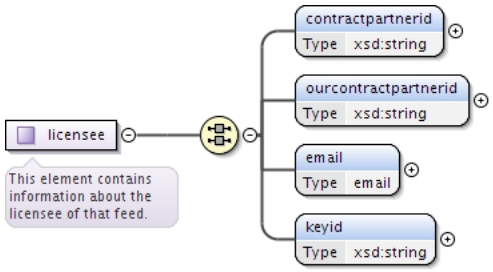
```
    <xsd:documentation xml:lang="en">This element contains information about the sender of
 that feed.</xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="contractpartnerid" type="xsd:string"/>
    <xsd:element name="ourcontractpartnerid" type="xsd:string"/>
    <xsd:element name="email" type="email"/>
    <xsd:element name="keyid" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
```

Complex Type `licensor`

Namespace	No namespace
Annotations	`This element contains information about the licensor of that feed.`
Diagram	
Used by	Element feedinfo/licensor
Model	ALL(contractpartnerid ourcontractpartnerid email keyid)
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Source	

```
<xsd:complexType name="licensor">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element contains information about the licensor
 of that feed.</xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="contractpartnerid" type="xsd:string"/>
    <xsd:element name="ourcontractpartnerid" type="xsd:string"/>
    <xsd:element name="email" type="email"/>
    <xsd:element name="keyid" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
```

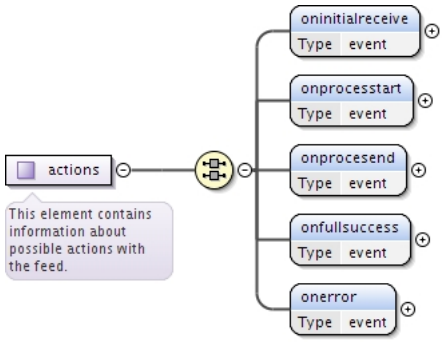
Complex Type `licensee`

Namespace	No namespace
Annotations	`This element contains information about the licensee of that feed.`
Diagram	
Used by	Element feedinfo/licensee
Model	ALL(contractpartnerid ourcontractpartnerid email keyid)
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Source	

```
<xsd:complexType name="licensee">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element contains information about the licensee
 of that feed.</xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="contractpartnerid" type="xsd:string"/>
    <xsd:element name="ourcontractpartnerid" type="xsd:string"/>
    <xsd:element name="email" type="email"/>
```

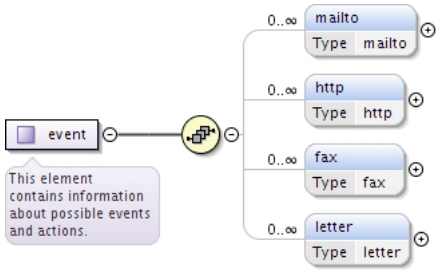
```
    <xsd:element name="keyid" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
```

Complex Type `actions`

Namespace	No namespace
Annotations	`This element contains information about possible actions with the feed.`
Diagram	
Used by	Element feedinfo/actions
Model	ALL(oninitialreceive onprocesstart onprocesend onfullsuccess onerror)
Children	onerror, onfullsuccess, oninitialreceive, onprocesend, onprocesstart
Source	

```
<xsd:complexType name="actions">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element contains information about possible
 actions with the feed.</xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="oninitialreceive" type="event"/>
    <xsd:element name="onprocesstart" type="event"/>
    <xsd:element name="onprocesend" type="event"/>
    <xsd:element name="onfullsuccess" type="event"/>
    <xsd:element name="onerror" type="event"/>
  </xsd:all>
</xsd:complexType>
```

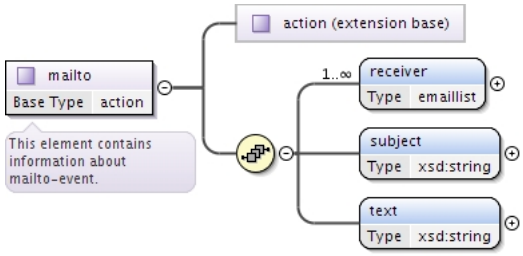
Complex Type `event`

Namespace	No namespace
Annotations	`This element contains information about possible events and actions.`
Diagram	
Used by	Elements: actions/onerror, actions/onfullsuccess, actions/oninitialreceive, actions/onprocesend, actions/onprocesstart Complex Types: onerror, onfullsuccess, oninitialreceive, onprocessend, onprocesstart
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto
Source	

```
<xsd:complexType name="event">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element contains information about possible
 events and actions.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="mailto" type="mailto" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="http" type="http" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="fax" type="fax" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="letter" type="letter" minOccurs="0" maxOccurs="unbounded"/>
```

```
  </xsd:sequence>
</xsd:complexType>
```

Complex Type `mailto`

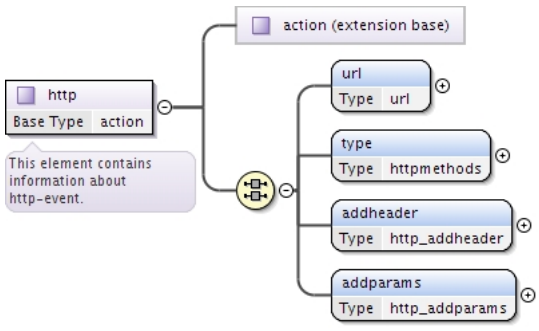
Namespace	No namespace
Annotations	`This element contains information about mailto-event.`
Diagram	
Type	extension of action
Type hierarchy	• action • mailto
Used by	Element event/mailto
Model	receiver+ , subject , text
Children	receiver, subject, text

`Source`	`<xsd:complexType name="mailto">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains information about mailto-event.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:complexContent>`
	`  <xsd:extension base="action">`
	`  <xsd:sequence>`
	`  <xsd:element name="receiver" type="emaillist" minOccurs="1" maxOccurs="unbounded"/>`
	`  <xsd:element name="subject" type="xsd:string"/>`
	`  <xsd:element name="text" type="xsd:string"/>`
	`  </xsd:sequence>`
	`  </xsd:extension>`
	`  </xsd:complexContent>`
	`</xsd:complexType>`

Complex Type `action`

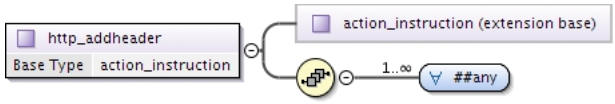
Namespace	No namespace
Diagram	
Used by	Complex Types fax, fileFtp, fileHttp, http, mailto
`Source`	`<xsd:complexType name="action"/>`

Complex Type `http`

Namespace	No namespace
Annotations	`This element contains information about http-event.`
Diagram	

Type	extension of action
Type hierarchy	• action • http
Used by	Elements event/http, realtime/http
Model	ALL(url type addheader addparams)
Children	addheader, addparams, type, url
Source	<xsd:complexType name="http"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about http-event.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="action"> <xsd:all> <xsd:element name="url" type="url"/> <xsd:element name="type" type="httpmethods"/> <xsd:element name="addheader" type="http_addheader"/> <xsd:element name="addparams" type="http_addparams"/> </xsd:all> </xsd:extension> </xsd:complexContent> </xsd:complexType>

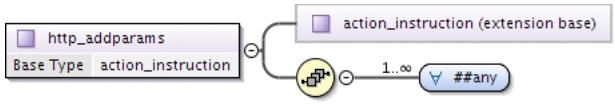
Complex Type `http_addheader`

Namespace	No namespace
Diagram	
Type	extension of action_instruction
Type hierarchy	• action_instruction • http_addheader
Used by	Element http/addheader
Model	ANY element from ANY namespace
Source	<xsd:complexType name="http_addheader"> <xsd:complexContent mixed="false"> <xsd:extension base="action_instruction"> <xsd:sequence> <xsd:any processContents="lax" maxOccurs="unbounded"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

Complex Type `action_instruction`

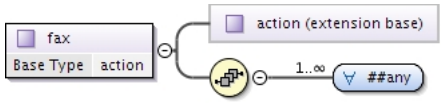
Namespace	No namespace
Diagram	
Used by	Complex Types http_addheader, http_addparams
Source	<xsd:complexType name="action_instruction"/>

Complex Type `http_addparams`

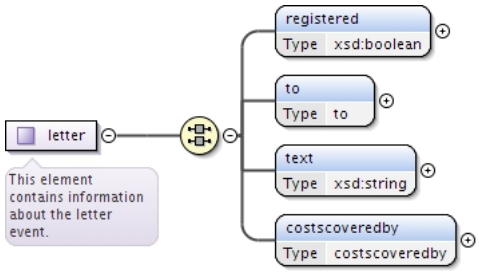
Namespace	No namespace
Diagram	
Type	extension of action_instruction
Type hierarchy	• action_instruction

	• http_addparams
Used by	Element http/addparams
Model	ANY element from ANY namespace
Source	<xsd:complexType name="http_addparams"> <xsd:complexContent> <xsd:extension base="action_instruction"> <xsd:sequence> <xsd:any processContents="lax" maxOccurs="unbounded"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

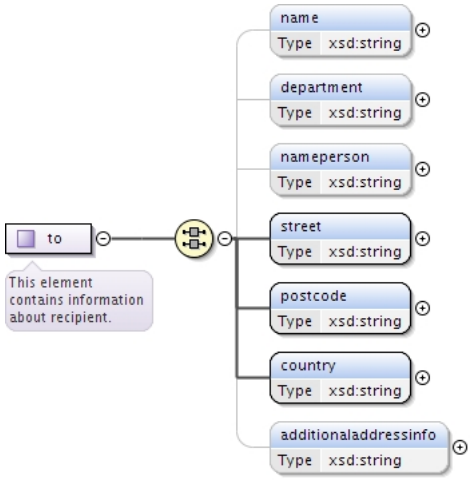
Complex Type fax

Namespace	No namespace
Diagram	
Type	extension of action
Type hierarchy	• action • fax
Used by	Element event/fax
Model	ANY element from ANY namespace
Source	<xsd:complexType name="fax"> <xsd:complexContent> <xsd:extension base="action"> <xsd:sequence> <xsd:any processContents="lax" maxOccurs="unbounded"/> </xsd:sequence> </xsd:extension> </xsd:complexContent> </xsd:complexType>

Complex Type letter

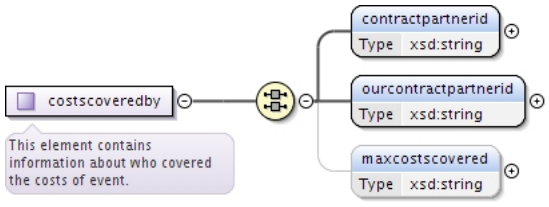
Namespace	No namespace
Annotations	This element contains information about the letter event.
Diagram	
Used by	Element event/letter
Model	ALL(registered to text costscoveredby)
Children	costscoveredby, registered, text, to
Source	<xsd:complexType name="letter"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about the letter event.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="registered" type="xsd:boolean"/> <xsd:element name="to" type="to"/> <xsd:element name="text" type="xsd:string"/> <xsd:element name="costscoveredby" type="costscoveredby"/> </xsd:all> </xsd:complexType>

Complex Type `to`

Namespace	No namespace
Annotations	`This element contains information about recipient.`
Diagram	
Used by	Element letter/to
Model	ALL(name{0,1} department{0,1} nameperson{0,1} street postcode country additionaladdressinfo{0,1})
Children	additionaladdressinfo, country, department, name, nameperson, postcode, street

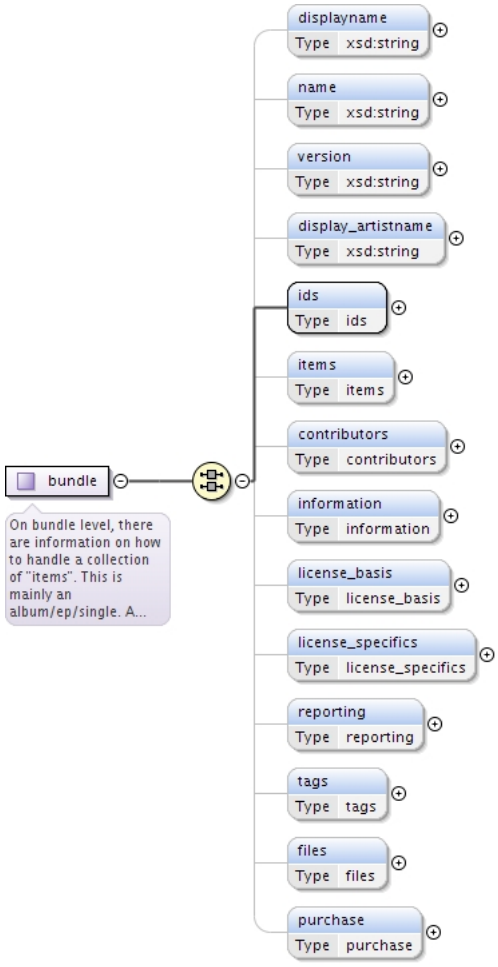
Source	`<xsd:complexType name="to">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains information about recipient.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:all>`
	`  <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>`
	`  <xsd:element name="department" type="xsd:string" minOccurs="0" maxOccurs="1"/>`
	`  <xsd:element name="nameperson" type="xsd:string" minOccurs="0" maxOccurs="1"/>`
	`  <xsd:element name="street" type="xsd:string"/>`
	`  <xsd:element name="postcode" type="xsd:string"/>`
	`  <xsd:element name="country" type="xsd:string"/>`
	`  <xsd:element name="additionaladdressinfo" type="xsd:string" minOccurs="0" maxOccurs="1"/>`
	`  </xsd:all>`
	`</xsd:complexType>`

Complex Type `costscoveredby`

Namespace	No namespace
Annotations	`This element contains information about who covered the costs of event.`
Diagram	
Used by	Element letter/costscoveredby
Model	ALL(contractpartnerid ourcontractpartnerid maxcostscovered{0,1})
Children	contractpartnerid, maxcostscovered, ourcontractpartnerid

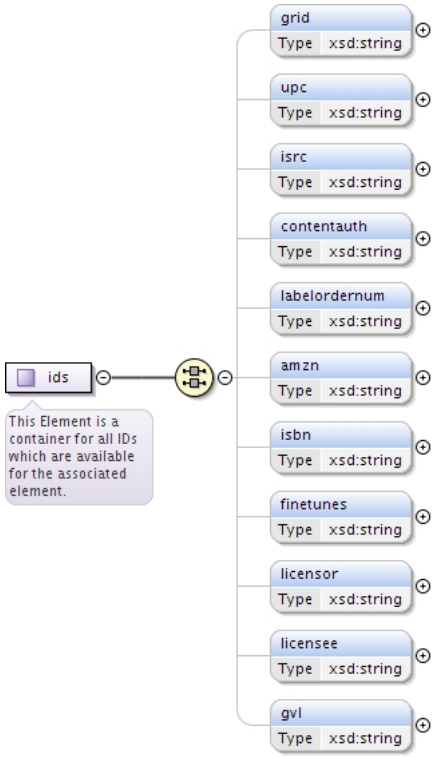
Source	`<xsd:complexType name="costscoveredby">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains information about who covered the costs of event.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:all>`
	`  <xsd:element name="contractpartnerid" type="xsd:string"/>`
	`  <xsd:element name="ourcontractpartnerid" type="xsd:string"/>`
	`  <xsd:element name="maxcostscovered" type="xsd:string" minOccurs="0" maxOccurs="1"/>`
	`  </xsd:all>`
	`</xsd:complexType>`

Complex Type bundle

Namespace	No namespace
Annotations	On bundle level, there are information on how to handle a collection of "items". This is mainly an album/ep/single. A bundle is identified by one unique identifier, but more unique identifiers could and should be transmitted as well (see below "ids"). Most notably on the bundle-level is the "bundle name" which is basically the conjunction of the "name"- and the "version"-field. Also to have this easy at hand, there should be the desired "display_artistname"-string be present on this level. Of course, the receiver of the feed can still calculate the "correct" display_artistname by evaluating the contributors (see below) for this.
Diagram	
Used by	Elements feed/bundle, related/bundle
Model	ALL(displayname{0,1} name{0,1} version{0,1} display_artistname{0,1} ids items{0,1} contributors{0,1} information{0,1} license_basis{0,1} license_specifics{0,1} reporting{0,1} tags{0,1} files{0,1} purchase{0,1})
Children	contributors, display_artistname, displayname, files, ids, information, items, license_basis, license_specifics, name, purchase, reporting, tags, version
Source	<xsd:complexType name="bundle"> <xsd:annotation> <xsd:documentation xml:lang="en">On bundle level, there are information on how to handle a collection of "items". This is mainly an album/ep/single. A bundle is identified by one unique identifier, but more unique identifiers could and should be transmitted as well (see below "ids"). Most notably on the bundle-level is the "bundle name" which is basically the conjunction of the "name"- and the "version"-field. Also to have this easy at hand, there should be the desired "display_artistname"-string be present on this level. Of course, the receiver of the feed can still calculate the "correct" display_artistname by evaluating the contributors (see below) for this.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="displayname" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="name" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="version" type="xsd:string" maxOccurs="1" minOccurs="0"/>

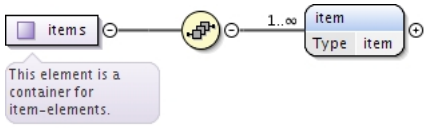
```
    <xsd:element name="display_artistname" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="ids" type="ids"/>
    <xsd:element name="items" type="items" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="contributors" type="contributors" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="information" type="information" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="license_basis" type="license_basis" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="license_specifics" type="license_specifics" maxOccurs="1"
 minOccurs="0"/>
    <xsd:element name="reporting" type="reporting" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="tags" type="tags" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="files" type="files" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="purchase" type="purchase" maxOccurs="1" minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

Complex Type `ids`

Namespace	No namespace
Annotations	`This Element is a container for all IDs which are available for the associated element.`
Diagram	
Used by	Elements bundle/ids, contributor/ids, item/ids
Model	ALL(grid{0,1} upc{0,1} isrc{0,1} contentauth{0,1} labelordernum{0,1} amzn{0,1} isbn{0,1} finetunes{0,1} licensor{0,1} licensee{0,1} gvl{0,1})
Children	amzn, contentauth, finetunes, grid, gvl, isbn, isrc, labelordernum, licensee, licensor, upc
`Source`	

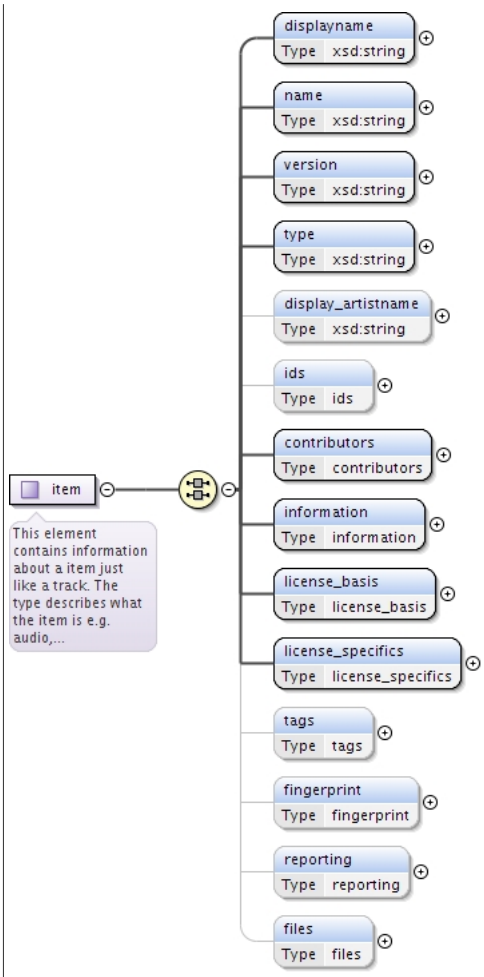
```
<xsd:complexType name="ids">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This Element is a container for all IDs which are
 available for the associated element.</xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="grid" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="upc" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="isrc" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="contentauth" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="labelordernum" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="amzn" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="isbn" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="finetunes" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="licensor" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="licensee" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="gvl" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

Complex Type `items`

Namespace	No namespace
Annotations	This element is a container for item-elements.
Diagram	
Used by	Element bundle/items
Model	item+
Children	item
Source	<xsd:complexType name="items"> <xsd:annotation> <xsd:documentation xml:lang="en">This element is a container for item-elements.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="item" type="item" maxOccurs="unbounded" minOccurs="1"/> </xsd:sequence> </xsd:complexType>

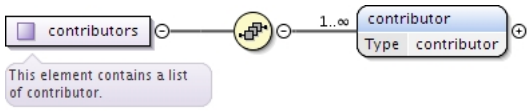
Complex Type `item`

Namespace	No namespace
Annotations	This element contains information about a item just like a track. The type describes what the item is e.g. audio, video, android-app et cetera. The entry "version" is important if different versions of the bundle exist. The licens_basic and license_specifics contains information and rules about pricing, allowed and disallowed territories, channels an so on. The child "files" hold information for the associated files for this item.

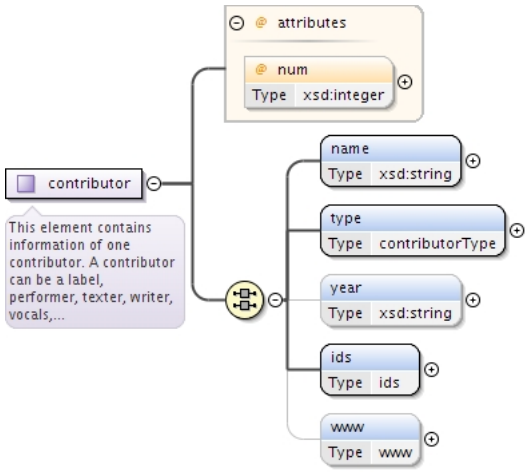
Diagram	
Used by	Elements feed/item, items/item
Model	ALL(displayname name version type display_artistname{0,1} ids{0,1} contributors information license_basis license_specifics tags{0,1} fingerprint{0,1} reporting{0,1} files{0,1})
Children	contributors, display_artistname, displayname, files, fingerprint, ids, information, license_basis, license_specifics, name, reporting, tags, type, version
Source	`<xsd:complexType name="item">` `<xsd:annotation>` `<xsd:documentation xml:lang="en">This element contains information about a item just like a track. The type describes what the item is e.g. audio, video, android-app et cetera. The entry "version" is important if different versions of the bundle exist. The licens_basic and license_specifics contains information and rules about pricing, allowed and disallowed territories, channels an so on. The child "files" hold information for the associated files for this item.</xsd:documentation>` `</xsd:annotation>` `<xsd:all>` `<xsd:element name="displayname" type="xsd:string"/>` `<xsd:element name="name" type="xsd:string"/>` `<xsd:element name="version" type="xsd:string"/>` `<xsd:element name="type" type="xsd:string"/>` `<xsd:element name="display_artistname" type="xsd:string" maxOccurs="1" minOccurs="0"/>` `<xsd:element name="ids" type="ids" maxOccurs="1" minOccurs="0"/>` `<xsd:element name="contributors" type="contributors"/>` `<xsd:element name="information" type="information"/>` `<xsd:element name="license_basis" type="license_basis"/>` `<xsd:element name="license_specifics" type="license_specifics"/>` `<xsd:element name="tags" type="tags" maxOccurs="1" minOccurs="0"/>` `<xsd:element name="fingerprint" type="fingerprint" maxOccurs="1" minOccurs="0"/>` `<xsd:element name="reporting" type="reporting" maxOccurs="1" minOccurs="0"/>` `<xsd:element name="files" type="files" maxOccurs="1" minOccurs="0"/>` `</xsd:all>` `</xsd:complexType>`

Complex Type `contributors`

Namespace	No namespace

Annotations	This element contains a list of contributor.
Diagram	
Used by	Elements bundle/contributors, item/contributors
Model	contributor+
Children	contributor
Source	<xsd:complexType name="contributors"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains a list of contributor.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="contributor" type="contributor" maxOccurs="unbounded" minOccurs="1"/> </xsd:sequence> </xsd:complexType>

Complex Type contributor

Namespace	No namespace
Annotations	This element contains information of one contributor. A contributor can be a label, performer, texter, writer, vocals, conductor, display_artist, compilator, copyright, production or clearinghouse. A Year should be provided in case the type equals copyright or production.
Diagram	
Used by	Element contributors/contributor
Model	ALL(name type year{0,1} ids www{0,1})
Children	ids, name, type, www, year

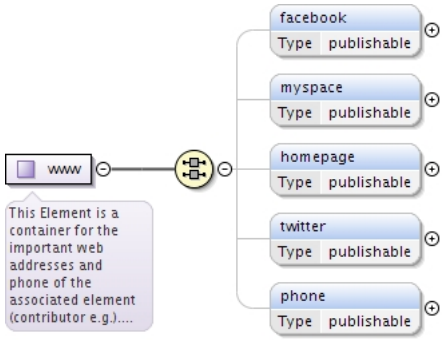
Attributes

QName	Type	Fixed	Default	Use
num	xsd:integer			optional

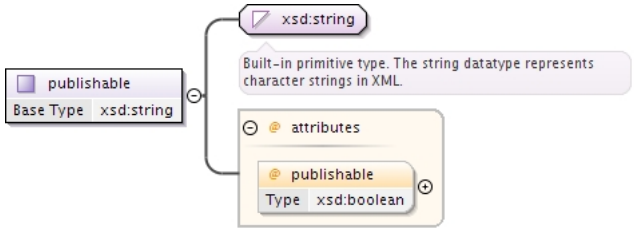
Source

```
<xsd:complexType name="contributor">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element contains information of one
 contributor. A contributor can be a label, performer, texter, writer, vocals, conductor,
 display_artist, compilator, copyright, production or clearinghouse. A Year should be
 provided in case the type equals copyright or production.</xsd:documentation>
  </xsd:annotation>
  <xsd:all>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="type" type="contributorType"/>
    <xsd:element name="year" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="ids" type="ids"/>
    <xsd:element name="www" type="www" maxOccurs="1" minOccurs="0"/>
  </xsd:all>
  <xsd:attribute name="num" type="xsd:integer"/>
</xsd:complexType>
```

Complex Type `www`

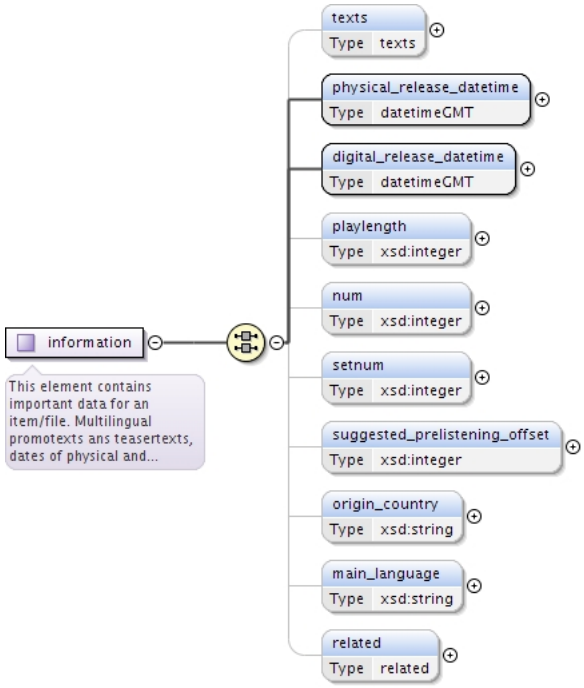
Namespace	No namespace
Annotations	This Element is a container for the important web addresses and phone of the associated element (contributor e.g.). Phone should be in international format. Every single information-entry cold be tagged "publishable" which would then mean wether customers of receiver are also allowed to be given this information. If publishable is not given, then this is granted.
Diagram	
Used by	Element contributor/www
Model	ALL(facebook{0,1} myspace{0,1} homepage{0,1} twitter{0,1} phone{0,1})
Children	facebook, homepage, myspace, phone, twitter
Source	`<xsd:complexType name="www">` `  <xsd:annotation>` ` <xsd:documentation xml:lang="en">This Element is a container for the important web addresses and phone of the associated element (contributor e.g.). Phone should be in international format. Every single information-entry cold be tagged "publishable" which would then mean wether customers of receiver are also allowed to be given this information. If publishable is not given, then this is granted.</xsd:documentation>` ` </xsd:annotation>` ` <xsd:all>` ` <xsd:element name="facebook" type="publishable" maxOccurs="1" minOccurs="0"/>` ` <xsd:element name="myspace" type="publishable" maxOccurs="1" minOccurs="0"/>` ` <xsd:element name="homepage" type="publishable" maxOccurs="1" minOccurs="0"/>` ` <xsd:element name="twitter" type="publishable" maxOccurs="1" minOccurs="0"/>` ` <xsd:element name="phone" type="publishable" maxOccurs="1" minOccurs="0"/>` ` </xsd:all>` `</xsd:complexType>`

Complex Type `publishable`

Namespace	No namespace
Diagram	
Type	extension of xsd:string
Used by	Elements www/facebook, www/homepage, www/myspace, www/phone, www/twitter
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **publishable** | xsd:boolean | | | optional

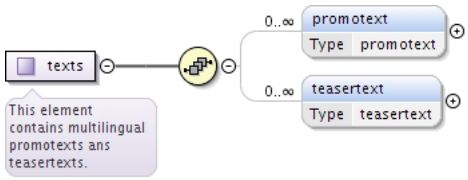
Source	`<xsd:complexType name="publishable">`
	`  <xsd:simpleContent>`
	`  <xsd:extension base="xsd:string">`
	`  <xsd:attribute name="publishable" type="xsd:boolean"/>`
	`  </xsd:extension>`
	`  </xsd:simpleContent>`
	`</xsd:complexType>`

Complex Type `information`

Namespace	No namespace
Annotations	This element contains important data for an item/file. Multilingual promotexts ans teasertexts, dates of physical and digital release, playlength of file, position of the file in relation to other file of bundle, number of set (e.g. 2 for cd 2), the suggested prelistining offset if the file not starts e.g. with significant content, origin country and main language of file and information about related bundles.
Diagram	
Used by	Elements bundle/information, item/information
Model	ALL(texts{0,1} physical_release_datetime digital_release_datetime playlength{0,1} num{0,1} setnum{0,1} suggested_prelistening_offset{0,1} origin_country{0,1} main_language{0,1} related{0,1})
Children	digital_release_datetime, main_language, num, origin_country, physical_release_datetime, playlength, related, setnum, suggested_prelistening_offset, texts
Source	<xsd:complexType name="information"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains important data for an item/file. Multilingual promotexts ans teasertexts, dates of physical and digital release, playlength of file, position of the file in relation to other file of bundle, number of set (e.g. 2 for cd 2), the suggested prelistining offset if the file not starts e.g. with significant content, origin country and main language of file and information about related bundles.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="texts" type="texts" maxOccurs="1" minOccurs="0"/> <xsd:element name="physical_release_datetime" type="datetimeGMT"/> <xsd:element name="digital_release_datetime" type="datetimeGMT"/> <xsd:element name="playlength" type="xsd:integer" maxOccurs="1" minOccurs="0"/> <xsd:element name="num" type="xsd:integer" maxOccurs="1" minOccurs="0"/> <xsd:element name="setnum" type="xsd:integer" maxOccurs="1" minOccurs="0"/> <xsd:element name="suggested_prelistening_offset" type="xsd:integer" maxOccurs="1" minOccurs="0"/> <xsd:element name="origin_country" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="main_language" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="related" type="related" maxOccurs="1" minOccurs="0"/> </xsd:all> </xsd:complexType>

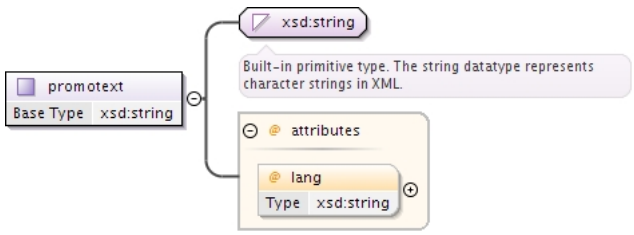
Complex Type `texts`

Namespace	No namespace
Annotations	This element contains multilingual promotexts ans teasertexts.

Diagram	
Used by	Element information/texts
Model	promotext* , teasertext*
Children	promotext, teasertext

Source	`<xsd:complexType name="texts">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains multilingual promotexts ans teasertexts.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:sequence>`
	`  <xsd:element name="promotext" type="promotext" maxOccurs="unbounded" minOccurs="0"/>`
	`  <xsd:element name="teasertext" type="teasertext" maxOccurs="unbounded" minOccurs="0"/>`
	`  </xsd:sequence>`
	`</xsd:complexType>`

Complex Type `promotext`

Namespace	No namespace
Diagram	
Type	extension of xsd:string
Used by	Element texts/promotext

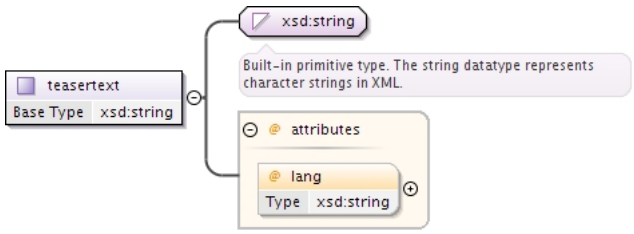
Attributes

QName	Type	Fixed	Default	Use
lang	xsd:string			optional

Source


```
<xsd:complexType name="promotext">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="lang" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```


Complex Type `teasertext`

Namespace	No namespace
Diagram	
Type	extension of xsd:string
Used by	Element texts/teasertext

Attributes

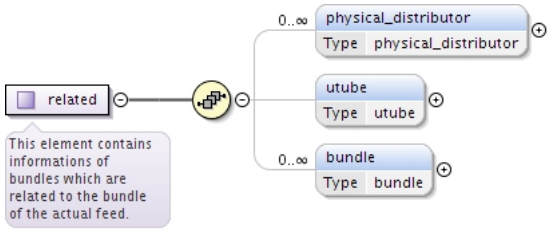
QName	Type	Fixed	Default	Use
lang	xsd:string			optional

Source

`<xsd:complexType name="teasertext">`

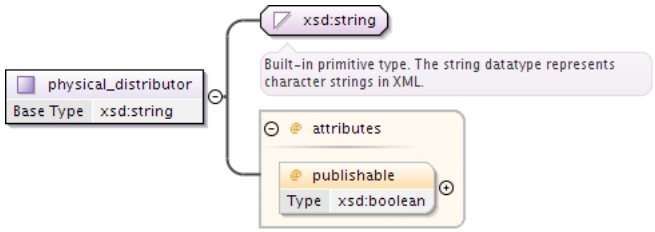
```
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="lang" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

Complex Type `related`

Namespace	No namespace
Annotations	This element contains informations of bundles which are related to the bundle of the actual feed.
Diagram	
Used by	Element information/related
Model	physical_distributor* , utube{0,1} , bundle*
Children	bundle, physical_distributor, utube
Source	

```
<xsd:complexType name="related">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element contains informations of bundles which
 are related to the bundle of the actual feed.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="physical_distributor" type="physical_distributor"
 maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="utube" type="utube" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="bundle" type="bundle" maxOccurs="unbounded" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Complex Type `physical_distributor`

Namespace	No namespace
Diagram	
Type	extension of xsd:string
Used by	Element related/physical_distributor

Attributes

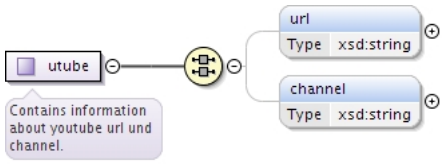
QName	Type	Fixed	Default	Use
publishable	xsd:boolean			optional

Source

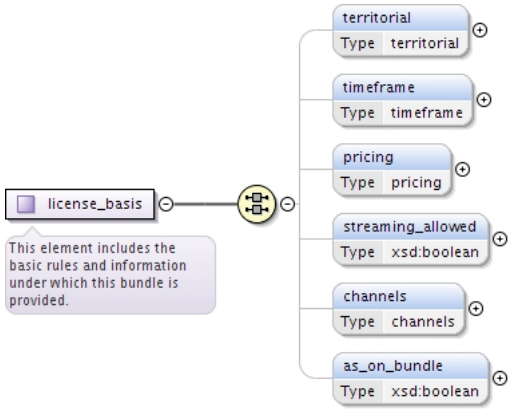
```
<xsd:complexType name="physical_distributor">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="publishable" type="xsd:boolean"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

Complex Type `utube`

Namespace	No namespace
Annotations	Contains information about youtube url und channel.

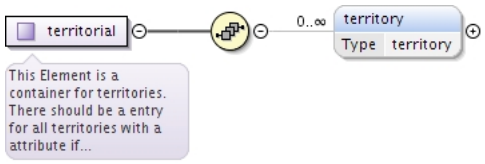
Diagram	
Used by	Element related/utube
Model	ALL(url{0,1} channel{0,1})
Children	channel, url
Source	<xsd:complexType name="utube"> <xsd:annotation> <xsd:documentation xml:lang="en">Contains information about youtube url und channel.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="url" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="channel" type="xsd:string" maxOccurs="1" minOccurs="0"/> </xsd:all> </xsd:complexType>

Complex Type `license_basis`

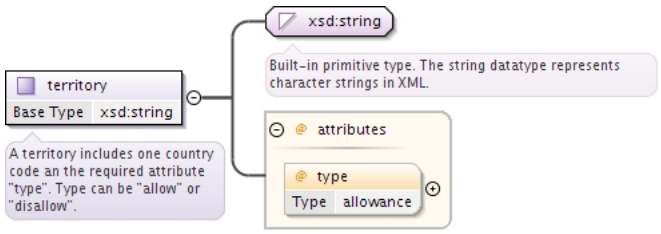
Namespace	No namespace
Annotations	This element includes the basic rules and information under which this bundle is provided.
Diagram	
Used by	Elements bundle/license_basis, item/license_basis
Model	ALL(territorial{0,1} timeframe{0,1} pricing{0,1} streaming_allowed{0,1} channels{0,1} as_on_bundle{0,1})
Children	as_on_bundle, channels, pricing, streaming_allowed, territorial, timeframe
Source	<xsd:complexType name="license_basis"> <xsd:annotation> <xsd:documentation xml:lang="en">This element includes the basic rules and information under which this bundle is provided.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="territorial" type="territorial" maxOccurs="1" minOccurs="0"/> <xsd:element name="timeframe" type="timeframe" maxOccurs="1" minOccurs="0"/> <xsd:element name="pricing" type="pricing" maxOccurs="1" minOccurs="0"/> <xsd:element name="streaming_allowed" type="xsd:boolean" maxOccurs="1" minOccurs="0"/> <xsd:element name="channels" type="channels" maxOccurs="1" minOccurs="0"/> <xsd:element name="as_on_bundle" type="xsd:boolean" maxOccurs="1" minOccurs="0"/> </xsd:all> </xsd:complexType>

Complex Type `territorial`

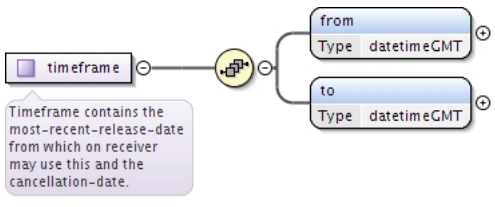
Namespace	No namespace
Annotations	This Element is a container for territories. There should be a entry for all territories with a attribute if distribution is allowed or not.

Diagram	
Used by	Element license_basis/territorial
Model	territory*
Children	territory
Source	<xsd:complexType name="territorial"> <xsd:annotation> <xsd:documentation xml:lang="en">This Element is a container for territories. There should be a entry for all territories with a attribute if distribution is allowed or not.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="territory" type="territory" maxOccurs="unbounded" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

Complex Type `territory`

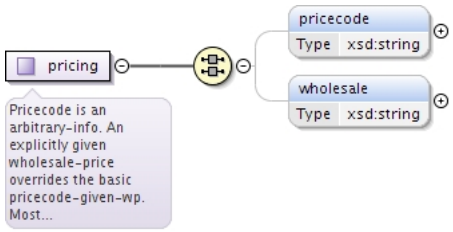
Namespace	No namespace
Annotations	A territory includes one country code an the required attribute "type". Type can be "allow" or "disallow".
Diagram	
Type	extension of xsd:string
Used by	Element territorial/territory
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **type** | allowance | | | optional
Source	<xsd:complexType name="territory"> <xsd:annotation> <xsd:documentation xml:lang="en">A territory includes one country code an the required attribute "type". Type can be "allow" or "disallow".</xsd:documentation> </xsd:annotation> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="type" type="allowance" use="optional"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType>

Complex Type `timeframe`

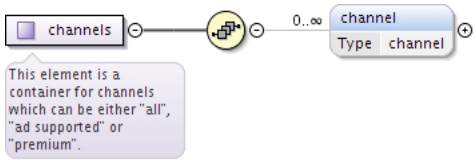
Namespace	No namespace
Annotations	Timeframe contains the most-recent-release-date from which on receiver may use this and the cancellation-date.
Diagram	
Used by	Element license_basis/timeframe

Model	from , to
Children	from, to
Source	<xsd:complexType name="timeframe"> <xsd:annotation> <xsd:documentation xml:lang="en">Timeframe contains the most-recent-release-date from which on receiver may use this and the cancellation-date.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="from" type="datetimeGMT"/> <xsd:element name="to" type="datetimeGMT"/> </xsd:sequence> </xsd:complexType>

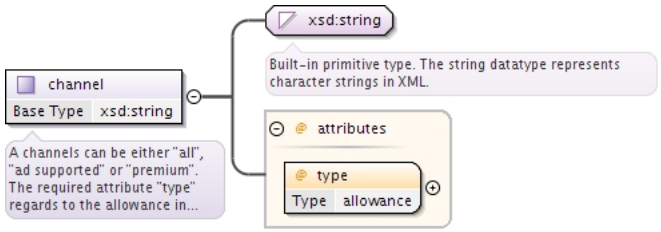
Complex Type pricing

Namespace	No namespace
Annotations	Pricecode is an arbitrary-info. An explicitly given wholesale-price overrides the basic pricecode-given-wp. Most probably either one of pricecode OR wholesaleprice should be given.
Diagram	
Used by	Element license_basis/pricing
Model	ALL(pricecode{0,1} wholesale{0,1})
Children	pricecode, wholesale
Source	<xsd:complexType name="pricing"> <xsd:annotation> <xsd:documentation xml:lang="en">Pricecode is an arbitrary-info. An explicitly given wholesale-price overrides the basic pricecode-given-wp. Most probably either one of pricecode OR wholesaleprice should be given.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="pricecode" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="wholesale" type="xsd:string" maxOccurs="1" minOccurs="0"/> </xsd:all> </xsd:complexType>

Complex Type channels

Namespace	No namespace
Annotations	This element is a container for channels which can be either "all", "ad supported" or "premium".
Diagram	
Used by	Element license_basis/channels
Model	channel*
Children	channel
Source	<xsd:complexType name="channels"> <xsd:annotation> <xsd:documentation xml:lang="en">This element is a container for channels which can be either "all", "ad supported" or "premium".</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="channel" type="channel" maxOccurs="unbounded" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

Complex Type `channel`

Namespace	No namespace
Annotations	A channels can be either "all", "ad supported" or "premium". The required attribute "type" regards to the allowance in reference to the channel. Type can be "allow" or "disallow".
Diagram	
Type	extension of xsd:string
Used by	Element channels/channel

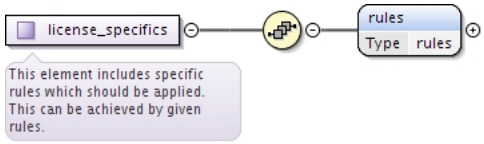
Attributes

QName	Type	Fixed	Default	Use
type	allowance			required

Source

```
<xsd:complexType name="channel">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">A channels can be either "all", "ad supported" or
 "premium". The required attribute "type" regards to the allowance in reference to the
 channel. Type can be "allow" or "disallow".</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="type" type="allowance" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

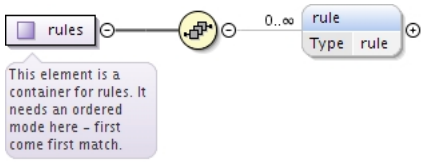
Complex Type `license_specifics`

Namespace	No namespace
Annotations	This element includes specific rules which should be applied. This can be achieved by given rules.
Diagram	
Used by	Elements bundle/license_specifics, item/license_specifics
Model	rules
Children	rules

Source

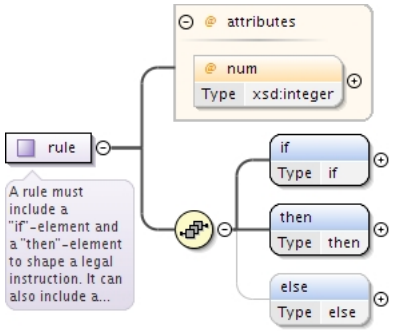
```
<xsd:complexType name="license_specifics">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element includes specific rules which should be
 applied. This can be achieved by given rules.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="rules" type="rules"/>
  </xsd:sequence>
</xsd:complexType>
```

Complex Type `rules`

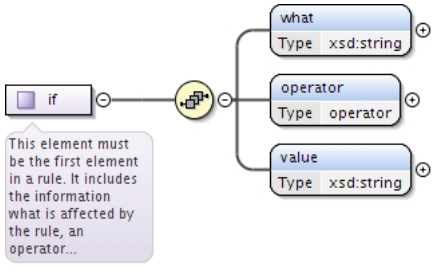
Namespace	No namespace
Annotations	This element is a container for rules. It needs an ordered mode here - first come first match.
Diagram	

Used by	Element license_specifics/rules
Model	rule*
Children	rule
Source	<xsd:complexType name="rules"> <xsd:annotation> <xsd:documentation xml:lang="en">This element is a container for rules. It needs an ordered mode here - first come first match.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="rule" type="rule" maxOccurs="unbounded" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

Complex Type `rule`

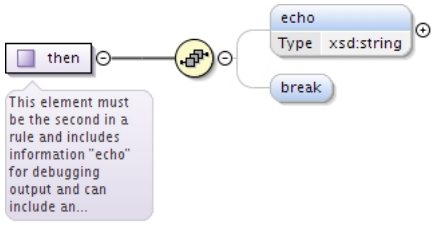
Namespace	No namespace
Annotations	A rule must include a "if"-element and a "then"-element to shape a legal instruction. It can also include a "else"-element.
Diagram	
Used by	Element rules/rule
Model	if , then , else{0,1}
Children	else, if, then
Attributes	**QName** | **Type** | **Fixed** | **Default** | **Use** **num** | xsd:integer | | | optional
Source	<xsd:complexType name="rule"> <xsd:annotation> <xsd:documentation xml:lang="en">A rule must include a "if"-element and a "then"-element to shape a legal instruction. It can also include a "else"-element.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="if" type="if"/> <xsd:element name="then" type="then"/> <xsd:element name="else" type="else" maxOccurs="1" minOccurs="0"/> </xsd:sequence> <xsd:attribute name="num" type="xsd:integer"/> </xsd:complexType>

Complex Type `if`

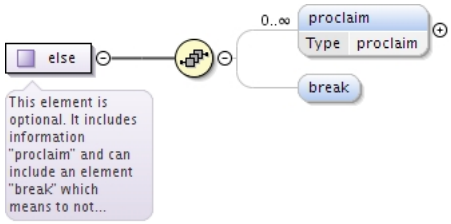
Namespace	No namespace
Annotations	This element must be the first element in a rule. It includes the information what is affected by the rule, an operator like "equals", "before","after","contains" or "containedin" and a value which will be compared.
Diagram	

Used by	Element rule/if
Model	what , operator , value
Children	operator, value, what
Source	<xsd:complexType name="if"> <xsd:annotation> <xsd:documentation xml:lang="en">This element must be the first element in a rule. It includes the information what is affected by the rule, an operator like "equals", "before","after","contains" or "containedin" and a value which will be compared.</ xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="what" type="xsd:string"/> <xsd:element name="operator" type="operator"/> <xsd:element name="value" type="xsd:string"/> </xsd:sequence> </xsd:complexType>

Complex Type then

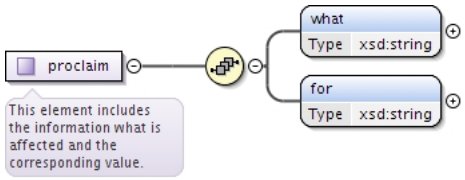
Namespace	No namespace
Annotations	This element must be the second in a rule and includes information "echo" for debugging output and can include an element "break" which means to not process any more rules.
Diagram	
Used by	Element rule/then
Model	echo{0,1} , break{0,1}
Children	break, echo
Source	<xsd:complexType name="then"> <xsd:annotation> <xsd:documentation xml:lang="en">This element must be the second in a rule and includes information "echo" for debugging output and can include an element "break" which means to not process any more rules.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="echo" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="break" maxOccurs="1" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

Complex Type else

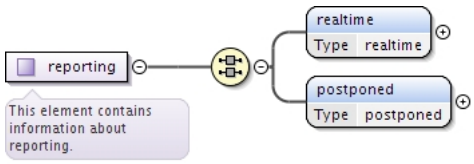
Namespace	No namespace
Annotations	This element is optional. It includes information "proclaim" and can include an element "break" which means to not process any more rules.
Diagram	
Used by	Element rule/else
Model	proclaim* , break{0,1}
Children	break, proclaim
Source	<xsd:complexType name="else">

```
  <xsd:annotation>
    <xsd:documentation xml:lang="en">This element is optional. It includes information
 "proclaim" and can include an element "break" which means to not process any more
 rules.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="proclaim" type="proclaim" maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="break" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Complex Type `proclaim`

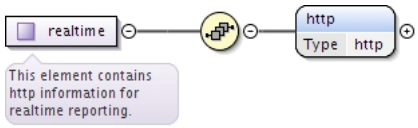
Namespace	No namespace
Annotations	This element includes the information what is affected and the corresponding value.
Diagram	
Used by	Element else/proclaim
Model	what , for
Children	for, what
Source	<xsd:complexType name="proclaim"> <xsd:annotation> <xsd:documentation xml:lang="en">This element includes the information what is affected and the corresponding value.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="what" type="xsd:string"/> <xsd:element name="for" type="xsd:string"/> </xsd:sequence> </xsd:complexType>

Complex Type `reporting`

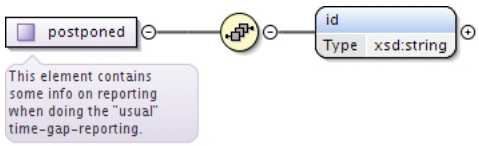
Namespace	No namespace
Annotations	This element contains information about reporting.
Diagram	
Used by	Elements bundle/reporting, item/reporting
Model	ALL(realtime postponed)
Children	postponed, realtime
Source	<xsd:complexType name="reporting"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about reporting.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="realtime" type="realtime"/> <xsd:element name="postponed" type="postponed"/> </xsd:all> </xsd:complexType>

Complex Type `realtime`

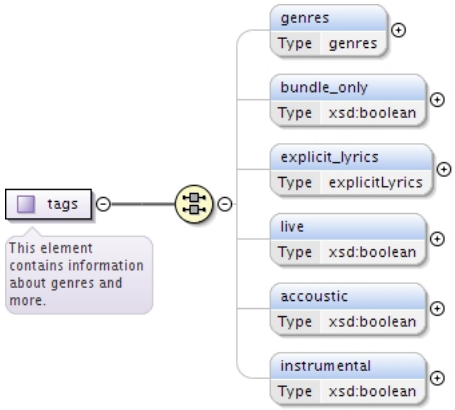
Namespace	No namespace
Annotations	This element contains http information for realtime reporting.

Diagram	
Used by	Element reporting/realtime
Model	http
Children	http
Source	<xsd:complexType name="realtime"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains http information for realtime reporting.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="http" type="http"/> </xsd:sequence> </xsd:complexType>

Complex Type postponed

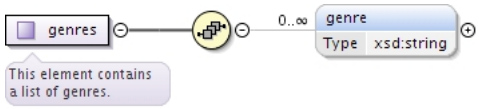
Namespace	No namespace
Annotations	This element contains some info on reporting when doing the "usual" time-gap-reporting.
Diagram	
Used by	Element reporting/postponed
Model	id
Children	id
Source	<xsd:complexType name="postponed"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains some info on reporting when doing the "usual" time-gap-reporting.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="id" type="xsd:string"/> </xsd:sequence> </xsd:complexType>

Complex Type tags

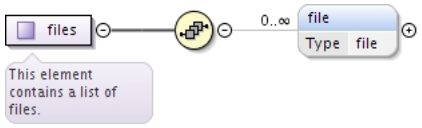
Namespace	No namespace
Annotations	This element contains information about genres and more.
Diagram	
Used by	Elements bundle/tags, item/tags
Model	ALL(genres{0,1} bundle_only{0,1} explicit_lyrics{0,1} live{0,1} accoustic{0,1} instrumental{0,1})

Children	accoustic, bundle_only, explicit_lyrics, genres, instrumental, live
Source	<xsd:complexType name="tags"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about genres and more.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="genres" type="genres" maxOccurs="1" minOccurs="0"/> <xsd:element name="bundle_only" type="xsd:boolean" maxOccurs="1" minOccurs="0"/> <xsd:element name="explicit_lyrics" type="explicitLyrics" maxOccurs="1" minOccurs="0"/> <xsd:element name="live" type="xsd:boolean" maxOccurs="1" minOccurs="0"/> <xsd:element name="accoustic" type="xsd:boolean" maxOccurs="1" minOccurs="0"/> <xsd:element name="instrumental" type="xsd:boolean" maxOccurs="1" minOccurs="0"/> </xsd:all> </xsd:complexType>

Complex Type genres

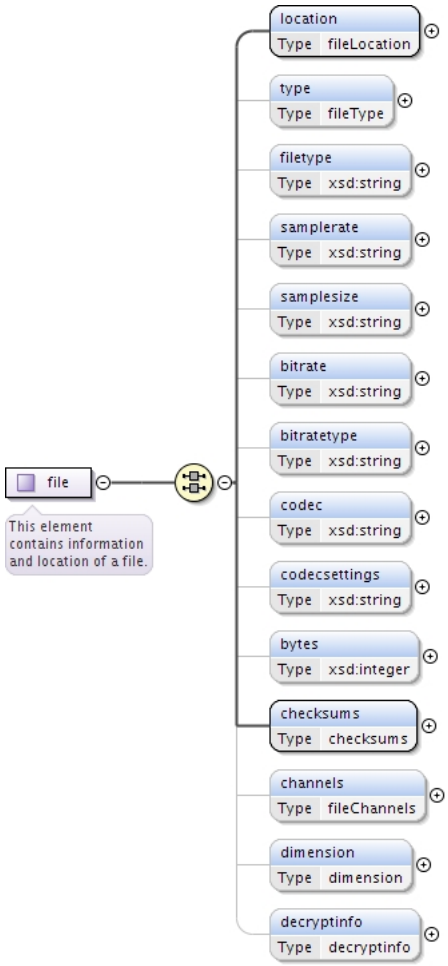
Namespace	No namespace
Annotations	This element contains a list of genres.
Diagram	
Used by	Element tags/genres
Model	genre*
Children	genre
Source	<xsd:complexType name="genres"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains a list of genres.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="genre" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

Complex Type files

Namespace	No namespace
Annotations	This element contains a list of files.
Diagram	
Used by	Elements bundle/files, item/files
Model	file*
Children	file
Source	<xsd:complexType name="files"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains a list of files.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="file" type="file" maxOccurs="unbounded" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

Complex Type file

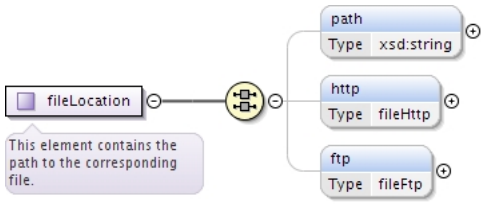
Namespace	No namespace
Annotations	This element contains information and location of a file.

Diagram	
Used by	Element files/file
Model	ALL(location type{0,1} filetype{0,1} samplerate{0,1} samplesize{0,1} bitrate{0,1} bitratetype{0,1} codec{0,1} codecsettings{0,1} bytes{0,1} checksums channels{0,1} dimension{0,1} decryptinfo{0,1})
Children	bitrate, bitratetype, bytes, channels, checksums, codec, codecsettings, decryptinfo, dimension, filetype, location, samplerate, samplesize, type

Source	`<xsd:complexType name="file">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains information and location of a file.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:all>`
	`  <xsd:element name="location" type="fileLocation"/>`
	`  <xsd:element name="type" type="fileType" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="filetype" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="samplerate" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="samplesize" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="bitrate" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="bitratetype" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="codec" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="codecsettings" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="bytes" type="xsd:integer" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="checksums" type="checksums"/>`
	`  <xsd:element name="channels" type="fileChannels" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="dimension" type="dimension" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="decryptinfo" type="decryptinfo" maxOccurs="1" minOccurs="0"/>`
	`  </xsd:all>`
	`</xsd:complexType>`

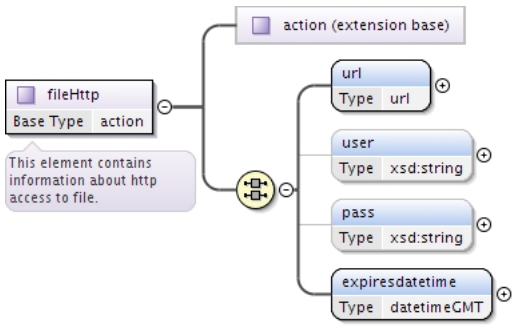
Complex Type `fileLocation`

Namespace	No namespace
Annotations	`This element contains the path to the corresponding file.`

Diagram	
Used by	Element file/location
Model	ALL(path{0,1} http{0,1} ftp{0,1})
Children	ftp, http, path

Source	`<xsd:complexType name="fileLocation">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains the path to the corresponding file.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:all>`
	`  <xsd:element name="path" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="http" type="fileHttp" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="ftp" type="fileFtp" maxOccurs="1" minOccurs="0"/>`
	`  </xsd:all>`
	`</xsd:complexType>`

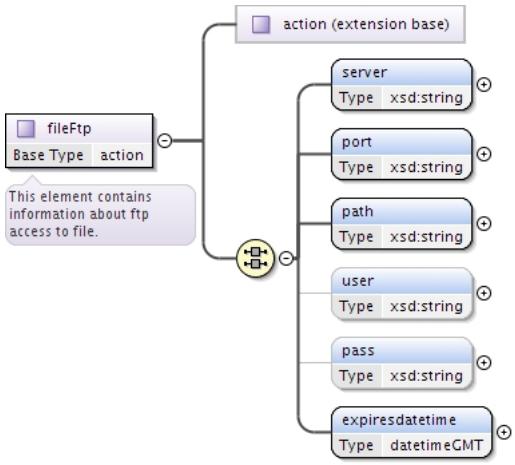
Complex Type `fileHttp`

Namespace	No namespace
Annotations	This element contains information about http access to file.
Diagram	
Type	extension of action
Type hierarchy	• action • fileHttp
Used by	Element fileLocation/http
Model	ALL(url user{0,1} pass{0,1} expiresdatetime)
Children	expiresdatetime, pass, url, user

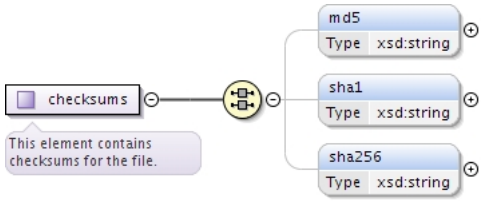
Source	`<xsd:complexType name="fileHttp">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains information about http access to file.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:complexContent>`
	`  <xsd:extension base="action">`
	`  <xsd:all>`
	`  <xsd:element name="url" type="url"/>`
	`  <xsd:element name="user" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="pass" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  <xsd:element name="expiresdatetime" type="datetimeGMT"/>`
	`  </xsd:all>`
	`  </xsd:extension>`
	`  </xsd:complexContent>`
	`</xsd:complexType>`

Complex Type `fileFtp`

Namespace	No namespace
Annotations	This element contains information about ftp access to file.

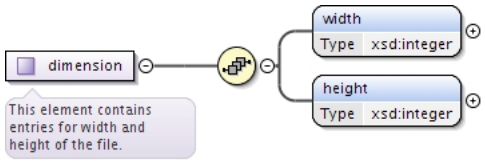
Diagram	
Type	extension of action
Type hierarchy	• action • fileFtp
Used by	Element fileLocation/ftp
Model	ALL(server port path user{0,1} pass{0,1} expiresdatetime)
Children	expiresdatetime, pass, path, port, server, user
Source	<xsd:complexType name="fileFtp"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about ftp access to file.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="action"> <xsd:all> <xsd:element name="server" type="xsd:string"/> <xsd:element name="port" type="xsd:string"/> <xsd:element name="path" type="xsd:string"/> <xsd:element name="user" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="pass" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="expiresdatetime" type="datetimeGMT"/> </xsd:all> </xsd:extension> </xsd:complexContent> </xsd:complexType>

Complex Type checksums

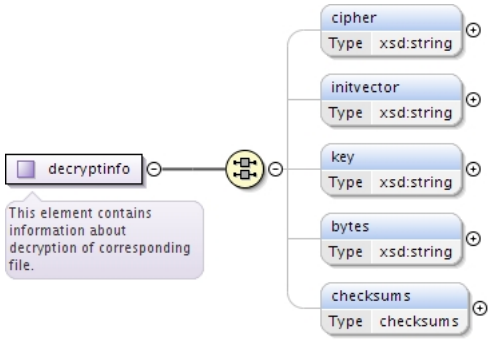
Namespace	No namespace
Annotations	This element contains checksums for the file.
Diagram	
Used by	Elements decryptinfo/checksums, file/checksums
Model	ALL(md5{0,1} sha1{0,1} sha256{0,1})
Children	md5, sha1, sha256
Source	<xsd:complexType name="checksums"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains checksums for the file.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="md5" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="sha1" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="sha256" type="xsd:string" maxOccurs="1" minOccurs="0"/> </xsd:all>

```
</xsd:complexType>
```

Complex Type dimension

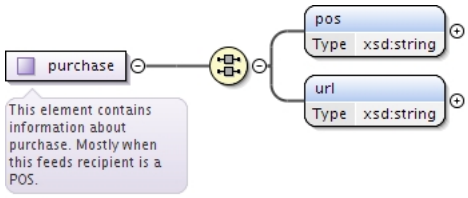
Namespace	No namespace
Annotations	This element contains entries for width and height of the file.
Diagram	
Used by	Element file/dimension
Model	width , height
Children	height, width
Source	<xsd:complexType name="dimension"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains entries for width and height of the file.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="width" type="xsd:integer"/> <xsd:element name="height" type="xsd:integer"/> </xsd:sequence> </xsd:complexType>

Complex Type decryptinfo

Namespace	No namespace
Annotations	This element contains information about decryption of corresponding file.
Diagram	
Used by	Element file/decryptinfo
Model	ALL(cipher{0,1} initvector{0,1} key{0,1} bytes{0,1} checksums{0,1})
Children	bytes, checksums, cipher, initvector, key
Source	<xsd:complexType name="decryptinfo"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about decryption of corresponding file.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="cipher" type="xsd:string" minOccurs="0" maxOccurs="1"/> <xsd:element name="initvector" type="xsd:string" minOccurs="0" maxOccurs="1"/> <xsd:element name="key" type="xsd:string" minOccurs="0" maxOccurs="1"/> <xsd:element name="bytes" type="xsd:string" minOccurs="0" maxOccurs="1"/> <xsd:element name="checksums" type="checksums" minOccurs="0" maxOccurs="1"/> </xsd:all> </xsd:complexType>

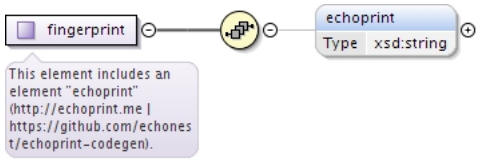
Complex Type purchase

Namespace	No namespace
Annotations	This element contains information about purchase. Mostly when this feeds recipient is a POS.

Diagram	
Used by	Element bundle/purchase
Model	ALL(pos url)
Children	pos, url

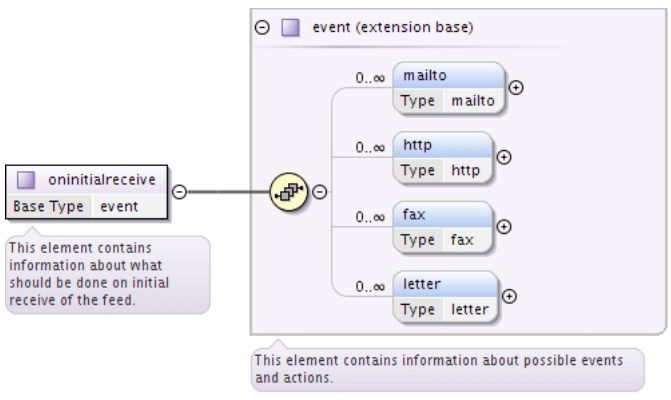
Source	`<xsd:complexType name="purchase">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains information about purchase.`
	`Mostly when this feeds recipient is a POS.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:all>`
	`  <xsd:element name="pos" type="xsd:string"/>`
	`  <xsd:element name="url" type="xsd:string"/>`
	`  </xsd:all>`
	`</xsd:complexType>`

Complex Type `fingerprint`

Namespace	No namespace
Annotations	This element includes an element "echoprint" (http://echoprint.me | https://github.com/echonest/echoprint-codegen).
Diagram	
Used by	Element item/fingerprint
Model	echoprint{0,1}
Children	echoprint

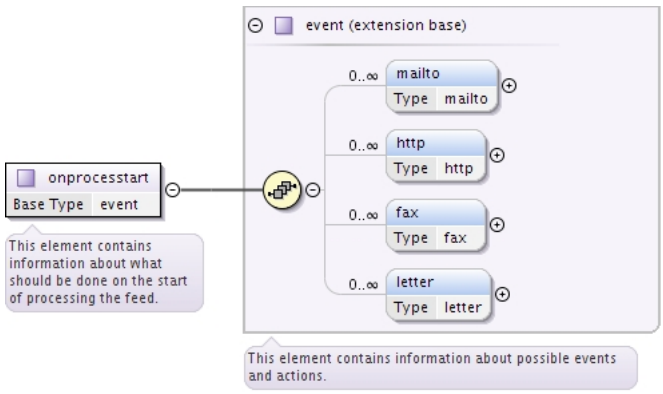
Source	`<xsd:complexType name="fingerprint">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element includes an element "echoprint" (http://`
	`echoprint.me | https://github.com/echonest/echoprint-codegen).</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:sequence>`
	`  <xsd:element name="echoprint" type="xsd:string" maxOccurs="1" minOccurs="0"/>`
	`  </xsd:sequence>`
	`</xsd:complexType>`

Complex Type `oninitialreceive`

Namespace	No namespace
Annotations	This element contains information about what should be done on initial receive of the feed.
Diagram	

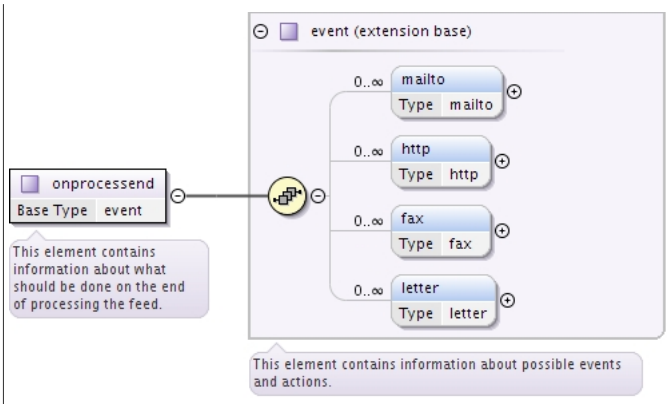
Type	extension of event
Type hierarchy	• event • oninitialreceive
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto
Source	<xsd:complexType name="oninitialreceive"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about what should be done on initial receive of the feed.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="event"/> </xsd:complexContent> </xsd:complexType>

Complex Type onprocesstart

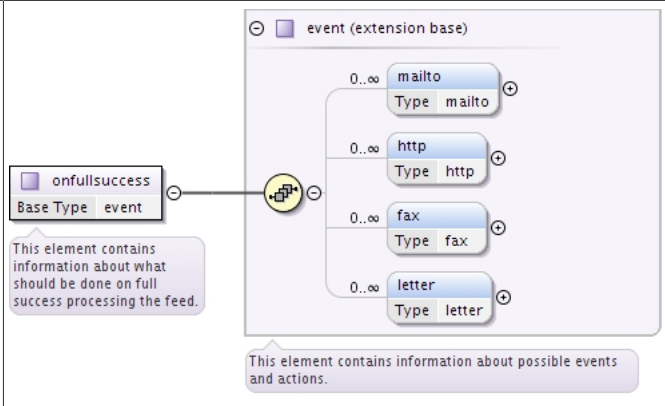
Namespace	No namespace
Annotations	This element contains information about what should be done on the start of processing the feed.
Diagram	
Type	extension of event
Type hierarchy	• event • onprocesstart
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto
Source	<xsd:complexType name="onprocesstart"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about what should be done on the start of processing the feed.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="event"/> </xsd:complexContent> </xsd:complexType>

Complex Type onprocessend

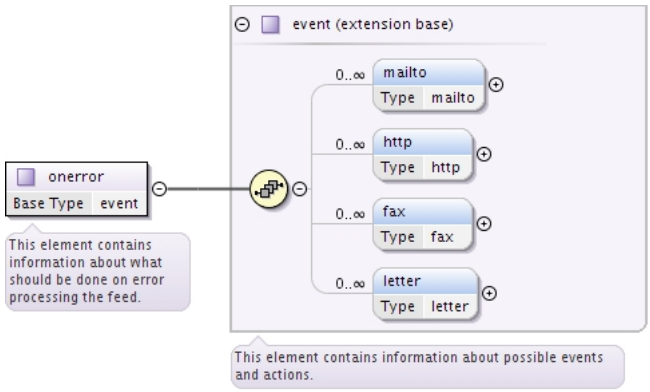
Namespace	No namespace
Annotations	This element contains information about what should be done on the end of processing the feed.

Diagram	
Type	extension of event
Type hierarchy	• event • onprocessend
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto
Source	<xsd:complexType name="onprocessend"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about what should be done on the end of processing the feed.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="event"/> </xsd:complexContent> </xsd:complexType>

Complex Type onfullsuccess

Namespace	No namespace
Annotations	This element contains information about what should be done on full success processing the feed.
Diagram	
Type	extension of event
Type hierarchy	• event • onfullsuccess
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto
Source	<xsd:complexType name="onfullsuccess"> <xsd:annotation> <xsd:documentation xml:lang="en">This element contains information about what should be done on full success processing the feed.</xsd:documentation> </xsd:annotation> <xsd:complexContent> <xsd:extension base="event"/> </xsd:complexContent> </xsd:complexType>

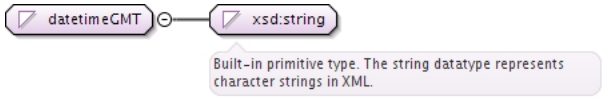
Complex Type `onerror`

Namespace	No namespace
Annotations	`This element contains information about what should be done on error processing the feed.`
Diagram	
Type	extension of event
Type hierarchy	• event • onerror
Model	mailto* , http* , fax* , letter*
Children	fax, http, letter, mailto

Source	`<xsd:complexType name="onerror">`
	`  <xsd:annotation>`
	`  <xsd:documentation xml:lang="en">This element contains information about what should be done on error processing the feed.</xsd:documentation>`
	`  </xsd:annotation>`
	`  <xsd:complexContent>`
	`  <xsd:extension base="event"/>`
	`  </xsd:complexContent>`
	`</xsd:complexType>`

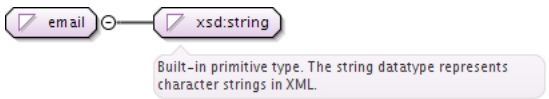
Simple Type(s)

Simple Type `datetimeGMT`

Namespace	No namespace	
Diagram		
Type	restriction of xsd:string	
Facets	pattern	`\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\ +\d{2}:\d{2}`
Used by	Elements	feedinfo/creationdatetime, feedinfo/effectivedatetime, fileFtp/expiresdatetime, fileHttp/expiresdatetime, information/digital_release_datetime, information/physical_release_datetime, timeframe/from, timeframe/to

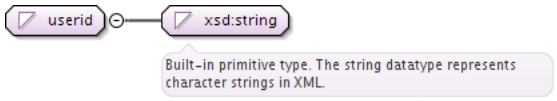
Source	`<xsd:simpleType name="datetimeGMT">`
	`  <xsd:restriction base="xsd:string">`
	`  <xsd:pattern value="\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2} GMT\+\d{2}:\d{2}"/>`
	`  <!-- "2010-01-31 00:00:00 GMT+00:00" - should be altered to some NMTOKENS or such ... -->`
	`  </xsd:restriction>`
	`</xsd:simpleType>`	

Simple Type `email`

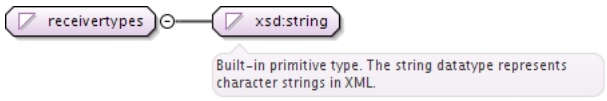
Namespace	No namespace
Diagram	

Type	xsd:string
Used by	Elements creator/email, crypto/relatedemail, licensee/email, licensor/email, sender/email
Source	<xsd:simpleType name="email"> <xsd:restriction base="xsd:string"/> </xsd:simpleType>

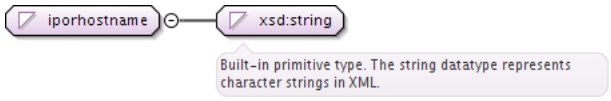
Simple Type `userid`

Namespace	No namespace
Diagram	
Type	xsd:string
Used by	Element creator/userid
Source	<xsd:simpleType name="userid"> <xsd:restriction base="xsd:string"/> </xsd:simpleType>

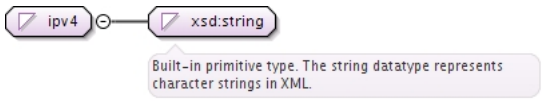
Simple Type `receivertypes`

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	enumeration `ftp` enumeration `ftps` enumeration `sftp` enumeration `openSDX-Beam`
Used by	Element receiver/type
Source	<xsd:simpleType name="receivertypes"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="ftp"/> <xsd:enumeration value="ftps"/> <xsd:enumeration value="sftp"/> <xsd:enumeration value="openSDX-Beam"/> </xsd:restriction> </xsd:simpleType>

Simple Type `iporhostname`

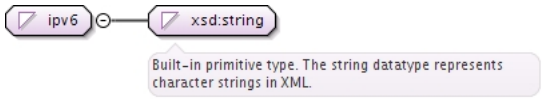
Namespace	No namespace
Diagram	
Type	xsd:string
Used by	Element receiver/servername
Source	<xsd:simpleType name="iporhostname"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType>

Simple Type `ipv4`

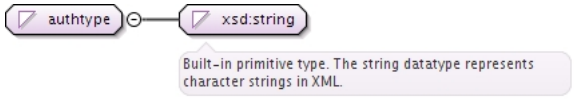
Namespace	No namespace
Diagram	

Type	restriction of xsd:string
Facets	pattern `\d{3}\.\d{3}\.\d{3}\.\d{3}`
Used by	Element receiver/serveripv4
Source	<xsd:simpleType name="ipv4"> <xsd:restriction base="xsd:string"> <xsd:pattern value="\d{3}\.\d{3}\.\d{3}\.\d{3}"/> <!-- should be checked for 0-255 etc. --> </xsd:restriction> </xsd:simpleType>

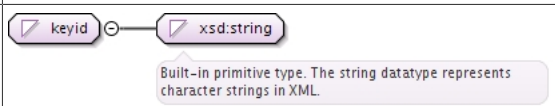
Simple Type ipv6

Namespace	No namespace
Diagram	
Type	xsd:string
Used by	Element receiver/serveripv6
Source	<xsd:simpleType name="ipv6"> <xsd:restriction base="xsd:string"> <!-- not pattern defined yet... --> </xsd:restriction> </xsd:simpleType>

Simple Type authtype

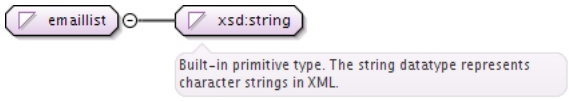
Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	enumeration `login` enumeration `keyfile` enumeration `kerberos` enumeration `keyfile+login` enumeration `keyfile+username`
Used by	Element receiver/authtype
Source	<xsd:simpleType name="authtype"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="login"/> <xsd:enumeration value="keyfile"/> <xsd:enumeration value="kerberos"/> <xsd:enumeration value="keyfile+login"/> <xsd:enumeration value="keyfile+username"/> </xsd:restriction> </xsd:simpleType>

Simple Type keyid

Namespace	No namespace
Diagram	
Type	xsd:string
Used by	Element crypto/usedkeyid
Source	<xsd:simpleType name="keyid"> <xsd:restriction base="xsd:string"> </xsd:restriction>

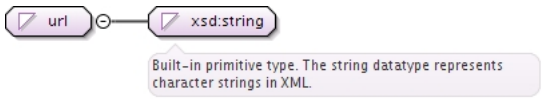
```
</xsd:simpleType>
```

Simple Type `emaillist`

Namespace	No namespace
Diagram	
Type	xsd:string
Used by	Element mailto/receiver

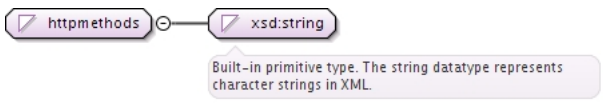
Source	`<xsd:simpleType name="emaillist">`
	`  <xsd:restriction base="xsd:string"/>`
	`  <!-- make to NMTOKENS or such... -->`
	`</xsd:simpleType>`

Simple Type `url`

Namespace	No namespace
Diagram	
Type	xsd:string
Used by	Elements fileHttp/url, http/url

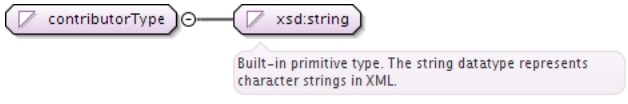
Source	`<xsd:simpleType name="url">`
	`  <xsd:restriction base="xsd:string">`
	`  <!-- not pattern defined yet... -->`
	`  </xsd:restriction>`
	`</xsd:simpleType>`

Simple Type `httpmethods`

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	enumeration `GET` enumeration `POST` enumeration `HEAD`
Used by	Element http/type

Source	`<xsd:simpleType name="httpmethods">`
	`  <xsd:restriction base="xsd:string">`
	`  <xsd:enumeration value="GET"/>`
	`  <xsd:enumeration value="POST"/>`
	`  <xsd:enumeration value="HEAD"/>`
	`  </xsd:restriction>`
	`</xsd:simpleType>`

Simple Type `contributorType`

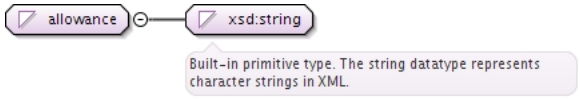
Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	enumeration `label` enumeration `performer`

	enumeration	texter
	enumeration	writer
	enumeration	vocals
	enumeration	conductor
	enumeration	display_artist
	enumeration	compilator
	enumeration	copyright
	enumeration	production
	enumeration	clearinghouse
Used by	Element	contributor/type

Source

```
<xsd:simpleType name="contributorType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="label"/>
    <xsd:enumeration value="performer"/>
    <xsd:enumeration value="texter"/>
    <xsd:enumeration value="writer"/>
    <xsd:enumeration value="vocals"/>
    <xsd:enumeration value="conductor"/>
    <xsd:enumeration value="display_artist"/>
    <xsd:enumeration value="compilator"/>
    <xsd:enumeration value="copyright"/>
    <xsd:enumeration value="production"/>
    <xsd:enumeration value="clearinghouse"/>
  </xsd:restriction>
</xsd:simpleType>
```

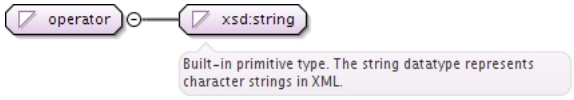
Simple Type `allowance`

Namespace	No namespace	
Diagram		
Type	restriction of xsd:string	
Facets	enumeration	allow
	enumeration	disallow
Used by	Attributes	channel/@type, territory/@type

Source

```
<xsd:simpleType name="allowance">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="allow"/>
    <xsd:enumeration value="disallow"/>
  </xsd:restriction>
</xsd:simpleType>
```

Simple Type `operator`

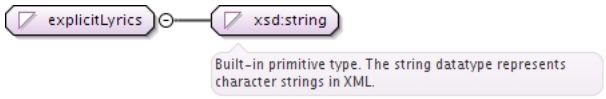
Namespace	No namespace	
Diagram		
Type	restriction of xsd:string	
Facets	enumeration	equals
	enumeration	before
	enumeration	after
	enumeration	contains
	enumeration	containedin
Used by	Element	if/operator

Source

```
<xsd:simpleType name="operator">
  <xsd:restriction base="xsd:string">
```

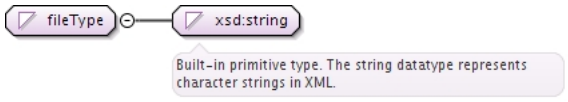
```
    <xsd:enumeration value="equals"/>
    <xsd:enumeration value="before"/>
    <xsd:enumeration value="after"/>
    <xsd:enumeration value="contains"/>
    <xsd:enumeration value="containedin"/>
  </xsd:restriction>
</xsd:simpleType>
```

Simple Type `explicitLyrics`

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	enumeration `true` enumeration `false` enumeration `cleaned`
Used by	Element tags/explicit_lyrics
Source	

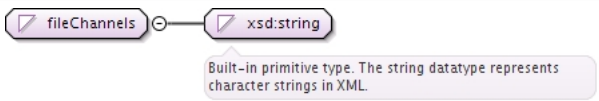
```
<xsd:simpleType name="explicitLyrics">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="true"/>
    <xsd:enumeration value="false"/>
    <xsd:enumeration value="cleaned"/>
  </xsd:restriction>
</xsd:simpleType>
```

Simple Type `fileType`

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	enumeration `full` enumeration `prelistening` enumeration `cover` enumeration `booklet`
Used by	Element file/type
Source	

```
<xsd:simpleType name="fileType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="full"/>
    <xsd:enumeration value="prelistening"/>
    <xsd:enumeration value="cover"/>
    <xsd:enumeration value="booklet"/>
  </xsd:restriction>
</xsd:simpleType>
```

Simple Type `fileChannels`

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	enumeration `mono` enumeration `stereo` enumeration `joint-stereo` enumeration `5.1`

Used by	Element file/channels
Source	<xsd:simpleType name="fileChannels"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="mono"/> <xsd:enumeration value="stereo"/> <xsd:enumeration value="joint-stereo"/> <xsd:enumeration value="5.1"/> </xsd:restriction> </xsd:simpleType>

Attribute(s)

Attribute publishable / @publishable

Namespace	No namespace
Type	xsd:boolean
Properties	content: simple
Used by	Complex Type publishable
Source	<xsd:attribute name="publishable" type="xsd:boolean"/>

Attribute contributor / @num

Namespace	No namespace
Type	xsd:integer
Properties	content: simple
Used by	Complex Type contributor
Source	<xsd:attribute name="num" type="xsd:integer"/>

Attribute promotext / @lang

Namespace	No namespace
Type	xsd:string
Properties	content: simple
Used by	Complex Type promotext
Source	<xsd:attribute name="lang" type="xsd:string"/>

Attribute teasertext / @lang

Namespace	No namespace
Type	xsd:string
Properties	content: simple
Used by	Complex Type teasertext
Source	<xsd:attribute name="lang" type="xsd:string"/>

Attribute physical_distributor / @publishable

Namespace	No namespace
Type	xsd:boolean
Properties	content: simple
Used by	Complex Type physical_distributor
Source	<xsd:attribute name="publishable" type="xsd:boolean"/>

Attribute territory / @type

Namespace	No namespace

Type	allowance	
Properties	use:	optional
Facets	enumeration	allow
	enumeration	disallow
Used by	Complex Type	territory
Source	<xsd:attribute name="type" type="allowance" use="optional"/>	

Attribute channel / @type

Namespace	No namespace	
Type	allowance	
Properties	use:	required
Facets	enumeration	allow
	enumeration	disallow
Used by	Complex Type	channel
Source	<xsd:attribute name="type" type="allowance" use="required"/>	

Attribute rule / @num

Namespace	No namespace	
Type	xsd:integer	
Properties	content:	simple
Used by	Complex Type	rule
Source	<xsd:attribute name="num" type="xsd:integer"/>	