

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

august 22, 2011

Table of Contents

Resource hierarchy:	5
Namespace: ""	6
Schema(s)	6
Main schema openSDX_00-00-00-01.xsd	6
Element(s)	6
Element feed	6
Element feed / feedinfo	6
Element feedinfo / onlytest	7
Element feedinfo / feedid	7
Element feedinfo / creationdatetime	8
Element feedinfo / effectivedatetime	8
Element feedinfo / creator	8
Element creator / email	9
Element creator / userid	9
Element creator / keyid	9
Element feedinfo / receiver	9
Element receiver / type	10
Element receiver / servername	10
Element receiver / serveripv4	10
Element receiver / serveripv6	10
Element receiver / authtype	11
Element receiver / username	11
Element receiver / crypto	11
Element crypto / relatedemail	12
Element crypto / usedkeyid	12
Element crypto / usedpubkey	12
Element feedinfo / sender	12
Element sender / contractpartnerid	13
Element sender / ourcontractpartnerid	13
Element sender / email	13
Element sender / keyid	13
Element feedinfo / licensor	13
Element licensor / contractpartnerid	14
Element licensor / ourcontractpartnerid	14
Element licensor / email	14
Element licensor / keyid	15
Element feedinfo / licensee	15
Element licensee / contractpartnerid	15
Element licensee / ourcontractpartnerid	15
Element licensee / email	16
Element licensee / keyid	16
Element feedinfo / actions	16
Element actions / oninitialreceive	17
Element event / mailto	17
Element mailto / receiver	18
Element mailto / subject	18
Element mailto / text	18
Element event / http	18
Element http / url	19
Element http / type	19
Element http / addheader	19
Element http / addparams	20
Element event / fax	20
Element event / letter	20
Element letter / registered	21
Element letter / to	21
Element to / name	21
Element to / department	22
Element to / nameperson	22
Element to / street	22

Element to / postcode	22
Element to / country	23
Element to / additionaladdressinfo	23
Element letter / text	23
Element letter / costscoveredby	23
Element costscoveredby / contractpartnerid	24
Element costscoveredby / ourcontractpartnerid	24
Element costscoveredby / maxcostscovered	24
Element actions / onprocesstart	24
Element actions / onprocesend	25
Element actions / onfullsuccess	25
Element actions / onerror	26
Element feed / bundle	26
Element bundle / displayname	28
Element bundle / name	28
Element bundle / version	28
Element bundle / display_artistname	28
Element bundle / ids	28
Element ids / grid	29
Element ids / upc	30
Element ids / isrc	30
Element ids / contentauth	30
Element ids / labelordernum	30
Element ids / amzn	30
Element ids / isbn	31
Element ids / finetunes	31
Element ids / licensor	31
Element ids / licensee	31
Element ids / gvl	32
Element bundle / items	32
Element items / item	32
Element item / displayname	34
Element item / name	34
Element item / version	34
Element item / type	34
Element item / display_artistname	34
Element item / ids	35
Element item / contributors	35
Element contributors / contributor	36
Element contributor / name	36
Element contributor / type	37
Element contributor / year	37
Element contributor / ids	38
Element contributor / www	38
Element www / facebook	39
Element www / myspace	39
Element www / homepage	40
Element www / twitter	40
Element www / phone	41
Element item / information	41
Element information / texts	42
Element texts / promotext	43
Element texts / teasertext	43
Element information / physical_release_datetime	43
Element information / digital_release_datetime	44
Element information / playlength	44
Element information / num	44
Element information / setnum	44
Element information / suggested_prelistening_offset	45
Element information / origin_country	45
Element information / main_language	50
Element information / related	53
Element related / physical_distributor	54
Element related / utube	54
Element utube / url	54
Element utube / channel	55
Element related / bundle	55
Element bundle / contributors	56
Element bundle / information	56
Element bundle / license_basis	57
Element license_basis / territorial	58
Element territorial / territory	58

Element license_basis / timeframe	59
Element timeframe / from	59
Element timeframe / to	60
Element license_basis / pricing	60
Element pricing / pricecode	60
Element pricing / wholesale	60
Element license_basis / streaming_allowed	61
Element license_basis / channels	61
Element channels / channel	61
Element license_basis / as_on_bundle	62
Element bundle / license_specifics	62
Element license_specifics / rules	62
Element rules / rule	63
Element rule / if	63
Element if / what	64
Element if / operator	64
Element if / value	64
Element rule / then	64
Element then / echo	65
Element then / break	65
Element rule / else	65
Element else / proclaim	65
Element proclaim / what	66
Element proclaim / for	66
Element else / break	66
Element bundle / reporting	66
Element reporting / realtime	67
Element realtime / http	67
Element reporting / postponed	68
Element postponed / id	68
Element bundle / tags	68
Element tags / genres	69
Element genres / genre	69
Element tags / bundle_only	85
Element tags / explicit_lyrics	85
Element tags / live	85
Element tags / accoustic	85
Element tags / instrumental	86
Element bundle / files	86
Element files / file	86
Element file / location	88
Element fileLocation / path	88
Element fileLocation / http	88
Element fileHttp / url	89
Element fileHttp / user	89
Element fileHttp / pass	89
Element fileHttp / expiresdatetime	90
Element fileLocation / ftp	90
Element fileFtp / server	91
Element fileFtp / port	91
Element fileFtp / path	91
Element fileFtp / user	91
Element fileFtp / pass	91
Element fileFtp / expiresdatetime	92
Element file / type	92
Element file / filetype	92
Element file / samplerate	92
Element file / prelistening_offset	93
Element file / prelistening_length	93
Element file / samplesize	93
Element file / bitrate	94
Element file / bitratetype	94
Element file / codec	94
Element file / codecsettings	94
Element file / bytes	95
Element file / checksums	95
Element checksums / md5	95
Element checksums / sha1	95
Element checksums / sha256	96
Element file / channels	96
Element file / dimension	96
Element dimension / width	97

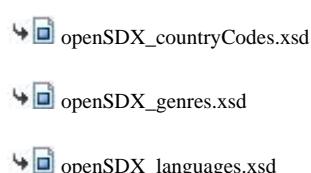
Element dimension / height	97
Element file / decryptinfo	97
Element decryptinfo / cipher	98
Element decryptinfo / initvector	98
Element decryptinfo / key	98
Element decryptinfo / bytes	98
Element decryptinfo / checksums	98
Element bundle / purchase	99
Element purchase / pos	99
Element purchase / url	100
Element item / license_basis	100
Element item / license_specifics	100
Element item / tags	101
Element item / fingerprint	101
Element fingerprint / echoprint	102
Element item / reporting	102
Element item / files	102
Element feed / item	103
Complex Type(s)	104
Complex Type feedinfo	104
Complex Type creator	105
Complex Type receiver	105
Complex Type crypto	106
Complex Type sender	106
Complex Type licensor	107
Complex Type licensee	107
Complex Type actions	108
Complex Type event	108
Complex Type mailto	109
Complex Type action	109
Complex Type http	109
Complex Type http_addheader	110
Complex Type action_instruction	110
Complex Type http_addparams	110
Complex Type fax	111
Complex Type letter	111
Complex Type to	112
Complex Type costscoveredby	112
Complex Type bundle	113
Complex Type ids	114
Complex Type items	115
Complex Type item	115
Complex Type contributors	116
Complex Type contributor	117
Complex Type www	118
Complex Type publishable	118
Complex Type information	119
Complex Type texts	119
Complex Type promotext	120
Complex Type teasertext	120
Complex Type related	121
Complex Type physical_distributor	121
Complex Type utube	121
Complex Type license_basis	122
Complex Type territorial	122
Complex Type territory	123
Complex Type timeframe	123
Complex Type pricing	124
Complex Type channels	124
Complex Type channel	125
Complex Type license_specifics	125
Complex Type rules	125
Complex Type rule	126
Complex Type if	126
Complex Type then	127
Complex Type else	127
Complex Type proclaim	128
Complex Type reporting	128
Complex Type realtime	128
Complex Type postponed	129
Complex Type tags	129
Complex Type genres	130

Complex Type files	130
Complex Type file	130
Complex Type fileLocation	132
Complex Type fileHttp	132
Complex Type fileFtp	133
Complex Type checksums	133
Complex Type dimension	134
Complex Type decryptinfo	134
Complex Type purchase	135
Complex Type fingerprint	135
Complex Type oninitialreceive	136
Complex Type onprocessstart	136
Complex Type onprocessend	137
Complex Type onfullsuccess	137
Complex Type onerror	138
Simple Type(s)	138
Simple Type datetimeGMT	138
Simple Type email	139
Simple Type userid	139
Simple Type receivertypes	139
Simple Type iporhostname	139
Simple Type ipv4	140
Simple Type ipv6	140
Simple Type authtype	140
Simple Type keyid	141
Simple Type emaillist	141
Simple Type url	141
Simple Type httpmethods	141
Simple Type contributorType	142
Simple Type allowance	142
Simple Type operator	143
Simple Type explicitLyrics	143
Simple Type fileType	143
Simple Type fileChannels	144
Attribute(s)	144
Attribute publishable / @publishable	144
Attribute contributor / @num	144
Attribute promotext / @lang	145
Attribute teasertext / @lang	145
Attribute physical_distributor / @publishable	145
Attribute territory / @type	145
Attribute channel / @type	145
Attribute rule / @num	145
Namespace: "http://fnppl.org/opensdx/countrycodes"	146
Schema(s)	146
Imported schema openSDX_countryCodes.xsd	146
Simple Type(s)	146
Simple Type countryCode	146
Namespace: "http://fnppl.org/opensdx/genres"	166
Schema(s)	166
Imported schema openSDX_genres.xsd	166
Simple Type(s)	166
Simple Type genre	166
Namespace: "http://fnppl.org/opensdx/languages"	190
Schema(s)	190
Imported schema openSDX_languages.xsd	190
Simple Type(s)	191
Simple Type language	191

Resource hierarchy:

Legend: Import, Include, Redefine, Cycle detected

openSDX_00-00-00-01.xsd



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	<pre> classDiagram class feed { <<General Element for the whole XML-Doc (root)>> } class feedinfo { <<Type feedinfo>> } class bundle { <<Type bundle>> } class item { <<Type item>> } feed "0..1" --> feedinfo feed "0..>" --> bundle feed "0..>" --> item </pre>
Properties	content: complex
Model	feedinfo , bundle* , item*
Children	bundle, feedinfo, item
Instance	<pre> <feed> <feedinfo>{1,1}</feedinfo> <bundle>{0,unbounded}</bundle> <item>{0,unbounded}</item> </feed> </pre>
Source	<pre> <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> </pre>

Element feed / feedinfo

Namespace	No namespace
-----------	--------------

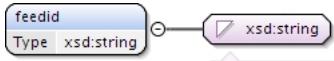
Diagram	<pre> classDiagram class feedinfo { onlytest feedid creationdatetime effectivedatetime creator receiver sender licensor licensee actions } feedinfo "2" --> "2" feedinfo </pre>
Type	feedinfo
Properties	content: complex
Model	ALL(onlytest feedid creationdatetime effectivedatetime creator receiver sender licensor licensee actions{0,1})
Children	actions, creationdatetime, creator, effectivedatetime, feedid, licensee, licensor, onlytest, receiver, sender
Instance	<pre> <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> </pre>
Source	<code><xsd:element name="feedinfo" type="feedinfo"/></code>

Element feedinfo / onlytest

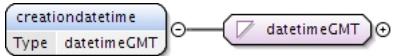
Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple
Source	<code><xsd:element name="onlytest" type="xsd:boolean"/></code>

Element feedinfo / feedid

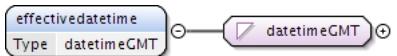
Namespace	No namespace
-----------	--------------

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

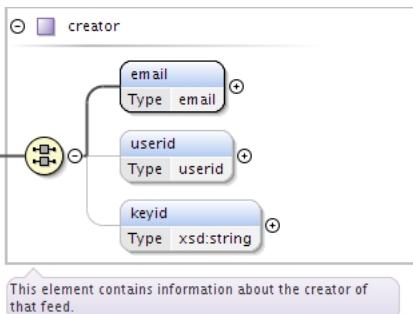
Element feedinfo / creationdatetime

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

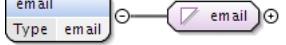
Element feedinfo / effectivedatetime

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

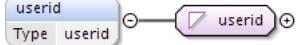
Element feedinfo / creator

Namespace	No namespace
Diagram	
Type	creator
Properties	content: complex
Model	ALL(email userid{0,1} keyid{0,1})
Children	email, keyid, userid
Instance	<creator> <email>{1,1}</email> <userid>{0,1}</userid> <keyid>{0,1}</keyid> </creator>
Source	<xsd:element name="creator" type="creator" />

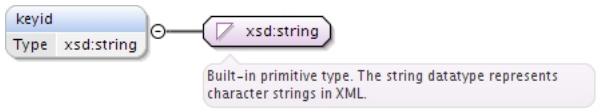
Element creator / email

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

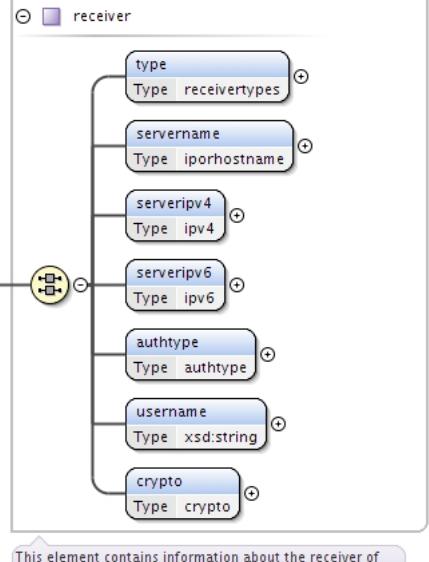
Element creator / userid

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

Element creator / keyid

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

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

Element receiver / type

Namespace	No namespace										
Diagram	<pre> graph LR type[Type type] --> receivertypes[receivertypes] </pre>										
Type	receivertypes										
Properties	content: simple										
Facets	<table border="1"> <tr><td>enumeration</td><td>ftp</td></tr> <tr><td>enumeration</td><td>ftps</td></tr> <tr><td>enumeration</td><td>sftp</td></tr> <tr><td>enumeration</td><td>webdav</td></tr> <tr><td>enumeration</td><td>openSDX fileserver</td></tr> </table>	enumeration	ftp	enumeration	ftps	enumeration	sftp	enumeration	webdav	enumeration	openSDX fileserver
enumeration	ftp										
enumeration	ftps										
enumeration	sftp										
enumeration	webdav										
enumeration	openSDX fileserver										
Source	<xsd:element name="type" type="receivertypes" />										

Element receiver / servername

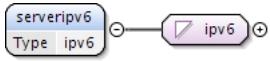
Namespace	No namespace
Diagram	<pre> graph LR servername[servername] --> iporhostname[iporhostname] </pre>
Type	iporhostname
Properties	content: simple
Source	<xsd:element name="servername" type="iporhostname" />

Element receiver / serveripv4

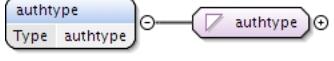
Namespace	No namespace				
Diagram	<pre> graph LR serveripv4[serveripv4] --> ipv4[ipv4] </pre>				
Type	ipv4				
Properties	content: simple				
Facets	<table border="1"> <tr><td>pattern</td><td>((25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1[1-9][0-9] 1[0-9][0-9])\.)\{3\}</td></tr> <tr><td></td><td>(25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1[1-9][0-9])\{3\}</td></tr> </table>	pattern	((25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1[1-9][0-9] 1[0-9][0-9])\.)\{3\}		(25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1[1-9][0-9])\{3\}
pattern	((25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1[1-9][0-9] 1[0-9][0-9])\.)\{3\}				
	(25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1[1-9][0-9])\{3\}				
Source	<xsd:element name="serveripv4" type="ipv4" />				

Element receiver / serveripv6

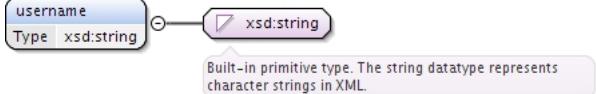
Namespace	No namespace
-----------	--------------

Diagram	
Type	ipv6
Properties	content: simple
Source	<xsd:element name="serveripv6" type="ipv6" />

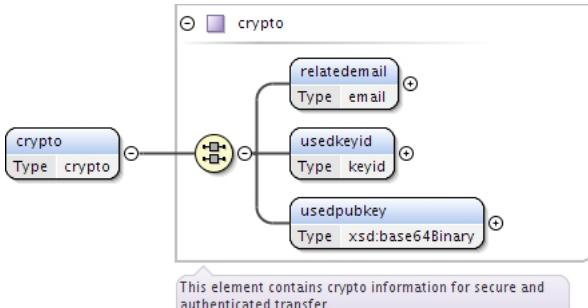
Element receiver / authtype

Namespace	No namespace										
Diagram											
Type	authtype										
Properties	content: simple										
Facets	<table border="1"> <tr><td>enumeration</td><td>login</td></tr> <tr><td>enumeration</td><td>keyfile</td></tr> <tr><td>enumeration</td><td>kerberos</td></tr> <tr><td>enumeration</td><td>keyfile+login</td></tr> <tr><td>enumeration</td><td>keyfile+username</td></tr> </table>	enumeration	login	enumeration	keyfile	enumeration	kerberos	enumeration	keyfile+login	enumeration	keyfile+username
enumeration	login										
enumeration	keyfile										
enumeration	kerberos										
enumeration	keyfile+login										
enumeration	keyfile+username										
Source	<xsd:element name="authtype" type="authtype" />										

Element receiver / username

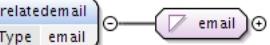
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="username" type="xsd:string" />

Element receiver / crypto

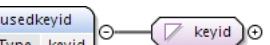
Namespace	No namespace
Diagram	
Type	crypto
Properties	content: complex
Model	ALL(relatedemail usedkeyid usedpubkey)
Children	relatedemail, usedkeyid, usedpubkey
Instance	<pre><crypto> <relatedemail>{1,1}</relatedemail> <usedkeyid>{1,1}</usedkeyid> <usedpubkey>{1,1}</usedpubkey> </crypto></pre>

Source	<code><xsd:element name="crypto" type="crypto"/></code>
--------	---

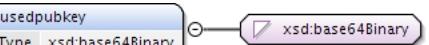
Element crypto / relatedemail

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

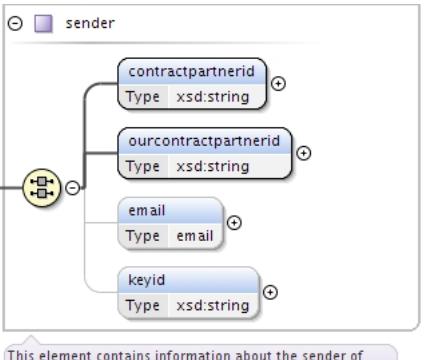
Element crypto / usedkeyid

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

Element crypto / usedpubkey

Namespace	No namespace
Diagram	 <small>Built-in primitive type. The base64Binary datatype represents Base64-encoded arbitrary binary data.</small>
Type	xsd:base64Binary
Properties	content: simple
Source	<code><xsd:element name="usedpubkey" type="xsd:base64Binary"/></code>

Element feedinfo / sender

Namespace	No namespace
Diagram	 <small>This element contains information about the sender of that feed.</small>
Type	sender
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid email{0,1} keyid{0,1})
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Instance	<code><sender> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid></code>

	<pre><email>{0,1}</email> <keyid>{0,1}</keyid> </sender></pre>
Source	<code><xsd:element name="sender" type="sender" /></code>

Element sender / contractpartnerid

Namespace	No namespace
Diagram	<p>The diagram shows the element <code>contractpartnerid</code> which is of type <code>xsd:string</code>. A callout box indicates that <code>xsd:string</code> is a built-in primitive type representing character strings in XML.</p>
Type	<code>xsd:string</code>
Properties	content: simple
Source	<code><xsd:element name="contractpartnerid" type="xsd:string" /></code>

Element sender / ourcontractpartnerid

Namespace	No namespace
Diagram	<p>The diagram shows the element <code>ourcontractpartnerid</code> which is of type <code>xsd:string</code>. A callout box indicates that <code>xsd:string</code> is a built-in primitive type representing character strings in XML.</p>
Type	<code>xsd:string</code>
Properties	content: simple
Source	<code><xsd:element name="ourcontractpartnerid" type="xsd:string" /></code>

Element sender / email

Namespace	No namespace
Diagram	<p>The diagram shows the element <code>email</code> which is of type <code>xsd:string</code>. A callout box indicates that <code>xsd:string</code> is a built-in primitive type representing character strings in XML.</p>
Type	<code>email</code>
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<code><xsd:element name="email" type="email" maxOccurs="1" minOccurs="0" /></code>

Element sender / keyid

Namespace	No namespace
Diagram	<p>The diagram shows the element <code>keyid</code> which is of type <code>xsd:string</code>. A callout box indicates that <code>xsd:string</code> is a built-in primitive type representing character strings in XML.</p>
Type	<code>xsd:string</code>
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<code><xsd:element name="keyid" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>

Element feedinfo / licensor

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram class licensor { contractpartnerid : xsd:string ourcontractpartnerid : xsd:string email : email keyid : xsd:string } licensor < -- licensor note over licensor: This element contains information about the licensor of that feed. </pre>
Type	licensor
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid email{0,1} keyid{0,1})
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Instance	<pre> <licensor> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid> <email>{0,1}</email> <keyid>{0,1}</keyid> </licensor> </pre>
Source	<xsd:element name="licensor" type="licensor"/>

Element licensor / contractpartnerid

Namespace	No namespace
Diagram	<pre> classDiagram class licensor { contractpartnerid : xsd:string } note over contractpartnerid: Built-in primitive type. The string datatype represents character strings in XML. </pre>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="contractpartnerid" type="xsd:string"/>

Element licensor / ourcontractpartnerid

Namespace	No namespace
Diagram	<pre> classDiagram class licensor { ourcontractpartnerid : xsd:string } note over ourcontractpartnerid: Built-in primitive type. The string datatype represents character strings in XML. </pre>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="ourcontractpartnerid" type="xsd:string"/>

Element licensor / email

Namespace	No namespace
Diagram	<pre> classDiagram class licensor { email : email } note over email: image of an envelope icon </pre>
Type	email
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="email" type="email" maxOccurs="1" minOccurs="0"/>

Element licensor / keyid

Namespace	No namespace
Diagram	<p>The diagram shows a box labeled "keyid" with "Type xsd:string". A line connects it to a "xsd:string" icon. A callout bubble states: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="keyid" type="xsd:string" maxOccurs="1" minOccurs="0" />

Element feedinfo / licensee

Namespace	No namespace
Diagram	<p>The diagram shows a box labeled "licensee" with "Type licensee". A line connects it to a "licensee" icon. Inside the "licensee" box are four children: "contractpartnerid", "ourcontractpartnerid", "email", and "keyid", each with its own "xsd:string" type indicator. A callout bubble states: "This element contains information about the licensee of that feed."</p>
Type	licensee
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid email{0,1} keyid{0,1})
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Instance	<pre><licensee> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid> <email>{0,1}</email> <keyid>{0,1}</keyid> </licensee></pre>
Source	<xsd:element name="licensee" type="licensee" />

Element licensee / contractpartnerid

Namespace	No namespace
Diagram	<p>The diagram shows a box labeled "contractpartnerid" with "Type xsd:string". A line connects it to a "xsd:string" icon. A callout bubble states: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="contractpartnerid" type="xsd:string" />

Element licensee / ourcontractpartnerid

Namespace	No namespace
Diagram	<p>The diagram shows a box labeled "ourcontractpartnerid" with "Type xsd:string". A line connects it to a "xsd:string" icon. A callout bubble states: "Built-in primitive type. The string datatype represents character strings in XML."</p>

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

Element licensee / email

Namespace	No namespace						
Diagram	<pre> graph LR A[Type email] --> B[Type email] </pre>						
Type	email						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<xsd:element name="email" type="email" maxOccurs="1" minOccurs="0"/>						

Element licensee / keyid

Namespace	No namespace						
Diagram	<pre> graph LR A[keyid] --> B[xsd:string] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>						
Type	xsd:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<xsd:element name="keyid" type="xsd:string" maxOccurs="1" minOccurs="0"/>						

Element feedinfo / actions

Namespace	No namespace						
Diagram	<pre> graph LR A[actions] --> B[oninitialreceive] A --> C[onprocessstart] A --> D[onprocessend] A --> E[onfullsuccess] A --> F[onerror] </pre> <p>This element contains information about possible actions with the feed.</p>						
Type	actions						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	ALL(oninitialreceive onprocessstart onprocessend onfullsuccess onerror)						
Children	onerror, onfullsuccess, oninitialreceive, onprocessend, onprocessstart						
Instance	<pre> <actions> <oninitialreceive>{1,1}</oninitialreceive> <onprocessstart>{1,1}</onprocessstart> </pre>						

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

Element actions / oninitialreceive

Namespace	No namespace
Diagram	<p>This element contains information about possible events and actions.</p>
Type	event
Properties	content: complex
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Instance	<pre> <oninitialreceive> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </oninitialreceive> </pre>
Source	<pre><xsd:element name="oninitialreceive" type="event" /></pre>

Element event / mailto

Namespace	No namespace
Diagram	<p>This element contains information about mailto-event.</p>
Type	mailto
Type hierarchy	<ul style="list-style-type: none"> • action • mailto
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	receiver+, subject, text
Children	receiver, subject, text
Instance	<pre><mailto></pre>

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

Element mailto / receiver

Namespace	No namespace						
Diagram							
Type	emaillist						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>1</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	1	maxOccurs:	unbounded
content:	simple						
minOccurs:	1						
maxOccurs:	unbounded						
Source	<pre><xsd:element name="mailto" type="mailto" minOccurs="0" maxOccurs="unbounded" /></pre>						

Element mailto / subject

Namespace	No namespace		
Diagram			
Type	xsd:string		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Source	<pre><xsd:element name="subject" type="xsd:string" /></pre>		

Element mailto / text

Namespace	No namespace		
Diagram			
Type	xsd:string		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Source	<pre><xsd:element name="text" type="xsd:string" /></pre>		

Element event / http

Namespace	No namespace
Diagram	

Type	http
Type hierarchy	<ul style="list-style-type: none"> • action <ul style="list-style-type: none"> • http
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(url type addheader addparams)
Children	addheader, addparams, type, url
Instance	<pre><http> <url>{1,1}</url> <type>{1,1}</type> <addheader>{1,1}</addheader> <addparams>{1,1}</addparams> </http></pre>
Source	<code><xsd:element name="http" type="http" minOccurs="0" maxOccurs="unbounded"/></code>

Element http / url

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

Element http / type

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

Element http / addheader

Namespace	No namespace
Diagram	
Type	http_addheader
Type hierarchy	<ul style="list-style-type: none"> • action_instruction <ul style="list-style-type: none"> • http_addheader
Properties	content: complex
Model	ANY element from ANY namespace
Source	<code><xsd:element name="addheader" type="http_addheader"/></code>

Element http / addparams

Namespace	No namespace
Diagram	<pre> classDiagram class http_addparams { <<Base Type action_instruction>> } class addparams { <<Type http_addparams>> } http_addparams < -- addparams http_addparams "1..*" --> "#any" </pre>
Type	http_addparams
Type hierarchy	<ul style="list-style-type: none"> • 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	<pre> classDiagram class fax { <<Base Type action>> } class fax { <<Type fax>> } fax < -- fax fax "1..*" --> "#any" </pre>
Type	fax
Type hierarchy	<ul style="list-style-type: none"> • action • fax
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	ANY element from ANY namespace
Source	<xsd:element name="fax" type="fax" minOccurs="0" maxOccurs="unbounded" />

Element event / letter

Namespace	No namespace
Diagram	<pre> classDiagram class letter { <<Base Type>> } class letter { <<Type letter>> } letter < -- registered letter < -- to letter < -- text letter < -- costscoveredby registered <--> letter to <--> letter text <--> letter costscoveredby <--> letter </pre> <p>This element contains information about the letter event.</p>
Type	letter
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>

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

Element letter / registered

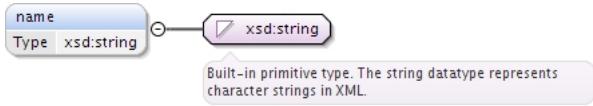
Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple
Source	<code><xsd:element name="registered" type="xsd:boolean" /></code>

Element letter / to

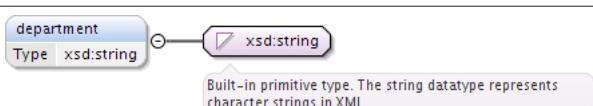
Namespace	No namespace
Diagram	
Type	to
Properties	content: complex
Model	ALL(name{0,1} department{0,1} nameperson{0,1} street postcode country additionaladdressinfo{0,1})
Children	additionaladdressinfo, country, department, name, nameperson, postcode, street
Instance	<pre><to> <name>{0,1}</name> <department>{0,1}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

Element to / name

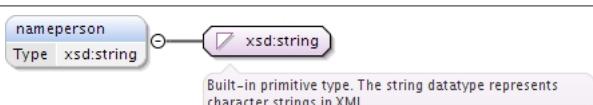
Namespace	No namespace
-----------	--------------

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

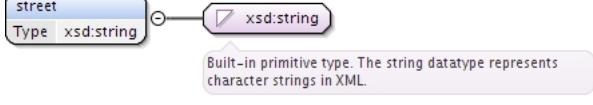
Element to / department

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

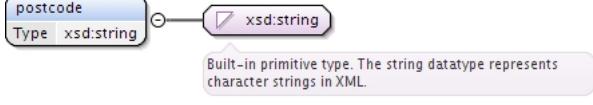
Element to / nameperson

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

Element to / street

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	<p>content: simple</p>
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	<pre><costscoveredby> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid> <maxcostscovered>{0,1}</maxcostscovered> </costscoveredby></pre>
Source	<pre><xsd:element name="costscoveredby" type="costscoveredby" /></pre>

Element costscoveredby / contractpartnerid

Namespace	No namespace
Diagram	<p>The diagram shows a class named "contractpartnerid" with a multiplicity of 0..1. It has a directed association labeled "xsd:string" pointing to a string icon. A callout bubble indicates: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<pre><xsd:element name="contractpartnerid" type="xsd:string" /></pre>

Element costscoveredby / ourcontractpartnerid

Namespace	No namespace
Diagram	<p>The diagram shows a class named "ourcontractpartnerid" with a multiplicity of 0..1. It has a directed association labeled "xsd:string" pointing to a string icon. A callout bubble indicates: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<pre><xsd:element name="ourcontractpartnerid" type="xsd:string" /></pre>

Element costscoveredby / maxcostscovered

Namespace	No namespace
Diagram	<p>The diagram shows a class named "maxcostscovered" with a multiplicity of 0..1. It has a directed association labeled "xsd:string" pointing to a string icon. A callout bubble indicates: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Source	<pre><xsd:element name="maxcostscovered" type="xsd:string" minOccurs="0" maxOccurs="1" /></pre>

Element actions / onprocessstart

Namespace	No namespace
Diagram	<p>The diagram shows a class named "onprocessstart" with a multiplicity of 0..1. It has a directed association labeled "event" pointing to an event icon. The event icon is expanded to show four possible events: "mailto" (multiplicity 0..infinity), "http" (multiplicity 0..infinity), "fax" (multiplicity 0..infinity), and "letter" (multiplicity 0..infinity). Each event is associated with its own "Type" (mailto, http, fax, letter) and a plus sign icon.</p> <p>A callout bubble indicates: "This element contains information about possible events and actions."</p>

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

Element actions / onprocesend

Namespace	No namespace
Diagram	<p>The diagram shows an event class with four associations named mailto, http, fax, and letter. Each association is multiplicity 0..∞ and has a Type of mailto, http, fax, or letter respectively. A callout box below the associations states: "This element contains information about possible events and actions."</p>
Type	event
Properties	content: complex
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Instance	<pre><onprocesend> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </onprocesend></pre>
Source	<xsd:element name="onprocesend" type="event"/>

Element actions / onfullsuccess

Namespace	No namespace
Diagram	<p>The diagram shows an event class with four associations named mailto, http, fax, and letter. Each association is multiplicity 0..∞ and has a Type of mailto, http, fax, or letter respectively. A callout box below the associations states: "This element contains information about possible events and actions."</p>
Type	event
Properties	content: complex

Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Instance	<pre><onfullsuccess> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </onfullsuccess></pre>
Source	<code><xsd:element name="onfullsuccess" type="event"/></code>

Element actions / onerror

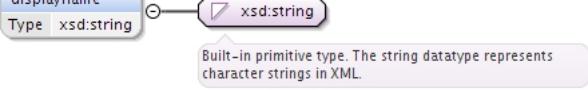
Namespace	No namespace
Diagram	<p>This element contains information about possible events and actions.</p>
Type	event
Properties	content: complex
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Instance	<pre><onerror> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </onerror></pre>
Source	<code><xsd:element name="onerror" type="event"/></code>

Element feed / bundle

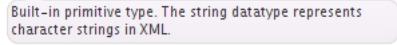
Namespace	No namespace
-----------	--------------

Diagram	<p>On bundle level, there are information on how to handle a collection of "items". This is mainly an album/ep/single. A...</p>						
Type	bundle						
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> <tr> <td style="padding: 2px;">maxOccurs:</td><td style="padding: 2px;">unbounded</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
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	<pre style="font-family: monospace; padding: 10px;"> <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></pre>						
Source	<xsd:element name="bundle" type="bundle" maxOccurs="unbounded" minOccurs="0" />						

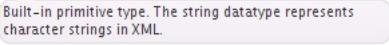
Element bundle / displayname

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

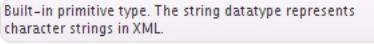
Element bundle / name

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

Element bundle / version

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

Element bundle / display_artistname

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

Element bundle / ids

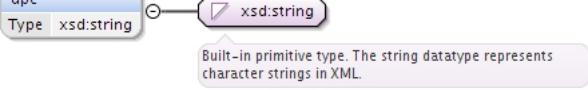
Namespace	No namespace
-----------	--------------

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

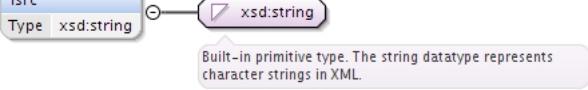
Element ids / grid

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

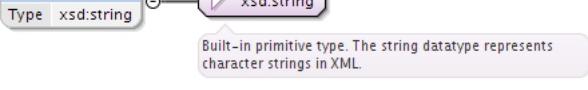
Element `ids / upc`

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

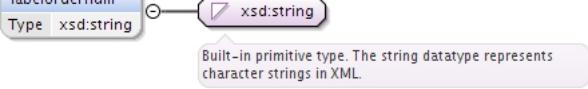
Element `ids / isrc`

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

Element `ids / contentauth`

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

Element `ids / labelordernum`

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

Element `ids / amzn`

Namespace	No namespace
-----------	--------------

Diagram	The diagram shows a blue rounded rectangle labeled "amzn" with a small "Type" label below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A tooltip below the connection states: "Built-in primitive type. The string datatype represents character strings in XML."/>						
Type	xsd:string						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
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	The diagram shows a blue rounded rectangle labeled "isbn" with a small "Type" label below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A tooltip below the connection states: "Built-in primitive type. The string datatype represents character strings in XML."/>						
Type	xsd:string						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
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	The diagram shows a blue rounded rectangle labeled "finetunes" with a small "Type" label below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A tooltip below the connection states: "Built-in primitive type. The string datatype represents character strings in XML."/>						
Type	xsd:string						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
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	The diagram shows a blue rounded rectangle labeled "licensor" with a small "Type" label below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A tooltip below the connection states: "Built-in primitive type. The string datatype represents character strings in XML."/>						
Type	xsd:string						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
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	A diagram showing the 'licensee' element. It has a rounded rectangle labeled 'licensee' with a 'Type' tab showing 'xsd:string'. A line connects it to a purple rounded rectangle labeled 'xsd:string'. A tooltip below says: 'Built-in primitive type. The string datatype represents character strings in XML.'						
Type	xsd:string						
Properties	<table> <tr><td>content:</td><td>simple</td></tr> <tr><td>minOccurs:</td><td>0</td></tr> <tr><td>maxOccurs:</td><td>1</td></tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
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	A diagram showing the 'gvl' element. It has a rounded rectangle labeled 'gvl' with a 'Type' tab showing 'xsd:string'. A line connects it to a purple rounded rectangle labeled 'xsd:string'. A tooltip below says: 'Built-in primitive type. The string datatype represents character strings in XML.'						
Type	xsd:string						
Properties	<table> <tr><td>content:</td><td>simple</td></tr> <tr><td>minOccurs:</td><td>0</td></tr> <tr><td>maxOccurs:</td><td>1</td></tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
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	A diagram showing the 'items' element. It has a rounded rectangle labeled 'items' with a 'Type' tab showing 'items'. A line connects it to another rounded rectangle labeled 'items' with a 'Type' tab showing 'item'. A tooltip below says: 'This element is a container for item-elements.'						
Type	items						
Properties	<table> <tr><td>content:</td><td>complex</td></tr> <tr><td>minOccurs:</td><td>0</td></tr> <tr><td>maxOccurs:</td><td>1</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
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	
	<pre> classDiagram class item { displayname name version type display_artistname ids contributors information license_basis license_specifics tags fingerprint reporting files } item < -- item note over item: This element contains information about a item just like a track. The type describes what the item is e.g. audio, ... </pre>
Type	item
Properties	<p>content: complex</p> <p>minOccurs: 1</p> <p>maxOccurs: unbounded</p>
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	<pre> <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> </pre>
Source	<xsd:element name="item" type="item" maxOccurs="unbounded" minOccurs="1"/>

Element item / displayname

Namespace	No namespace
Diagram	<p>The diagram shows the element 'displayname' with its type 'xsd:string'. A tooltip below the type indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="displayname" type="xsd:string"/>

Element item / name

Namespace	No namespace
Diagram	<p>The diagram shows the element 'name' with its type 'xsd:string'. A tooltip below the type indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="name" type="xsd:string"/>

Element item / version

Namespace	No namespace
Diagram	<p>The diagram shows the element 'version' with its type 'xsd:string'. A tooltip below the type indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="version" type="xsd:string"/>

Element item / type

Namespace	No namespace
Diagram	<p>The diagram shows the element 'type' with its type 'xsd:string'. A tooltip below the type indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="type" type="xsd:string"/>

Element item / display_artistname

Namespace	No namespace
Diagram	<p>The diagram shows the element 'display_artistname' with its type 'xsd:string'. A tooltip below the type indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple

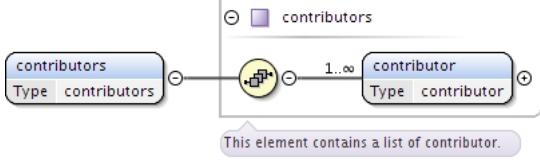
	minOccurs: 0
	maxOccurs: 1
Source	<xsd:element name="display_artistname" type="xsd:string" maxOccurs="1" minOccurs="0" />

Element item / ids

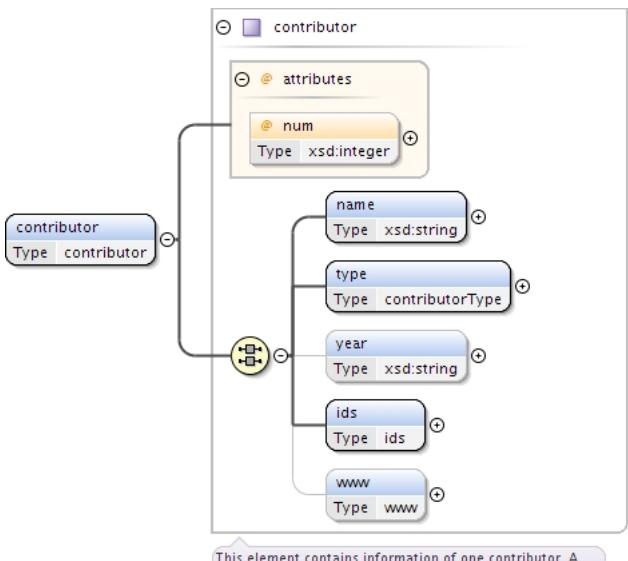
Namespace	No namespace
Diagram	<p>This Element is a container for all IDs which are available for the associated element.</p>
Type	ids
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	ALL(grid{0,1} upc{0,1} isrc{0,1} contentauth{0,1} labelordernum{0,1} amzn{0,1} isbn{0,1} 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	<pre><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></pre>
Source	<xsd:element name="ids" type="ids" maxOccurs="1" minOccurs="0" />

Element item / contributors

Namespace	No namespace
-----------	--------------

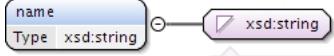
Diagram	 This element contains a list of contributor.
Type	contributors
Properties	content: complex
Model	contributor+
Children	contributor
Instance	<contributors> <contributor num="">{1,unbounded}</contributor> </contributors>
Source	<xsd:element name="contributors" type="contributors"/>

Element contributors / contributor

Namespace	No namespace										
Diagram	 This element contains information of one contributor. A contributor can be a label, performer, texter, editor,...										
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	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>num</td> <td>xsd:integer</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	num	xsd:integer			optional
QName	Type	Fixed	Default	Use							
num	xsd:integer			optional							
Source	<xsd:element name="contributor" type="contributor" maxOccurs="unbounded" minOccurs="1"/>										

Element contributor / name

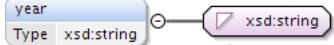
Namespace	No namespace
-----------	--------------

Diagram	 Built-in primitive type. The string datatype represents character strings in XML.
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="name" type="xsd:string" />

Element contributor / type

Namespace	No namespace
Diagram	 contributorType
Type	contributorType
Properties	content: simple
Facets	enumeration label enumeration performer enumeration texter enumeration editor enumeration conductor enumeration artist enumeration singer enumeration composer enumeration mixer enumeration remixer enumeration producer enumeration featuring enumeration with enumeration DJ enumeration versus enumeration meets enumeration presents enumeration compilator enumeration copyright enumeration production enumeration publisher enumeration clearinghouse
Source	<xsd:element name="type" type="contributorType" />

Element contributor / year

Namespace	No namespace
Diagram	 Built-in primitive type. The string datatype represents character strings in XML.
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	<p>This Element is a container for all IDs which are available for the associated element.</p>
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	<pre><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></pre>
Source	<pre><xsd:element name="ids" type="ids" /></pre>

Element contributor / www

Namespace	No namespace
-----------	--------------

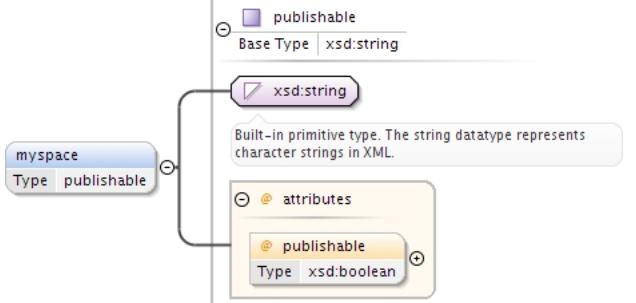
Diagram	<p>This Element is a container for the important web addresses and phone of the associated element (contributor e.g....)</p>
Type	www
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	ALL/facebook{0,1} myspace{0,1} homepage{0,1} twitter{0,1} phone{0,1})
Children	facebook, homepage, myspace, phone, twitter
Instance	<pre><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></pre>
Source	<xsd:element name="www" type="www" maxOccurs="1" minOccurs="0" />

Element www / facebook

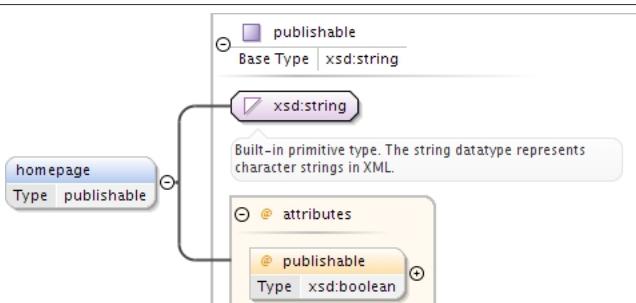
Namespace	No namespace										
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>										
Type	publishable										
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>publishable</td> <td>xsd:boolean</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	publishable	xsd:boolean			optional
QName	Type	Fixed	Default	Use							
publishable	xsd:boolean			optional							
Source	<xsd:element name="facebook" type="publishable" maxOccurs="1" minOccurs="0" />										

Element www / myspace

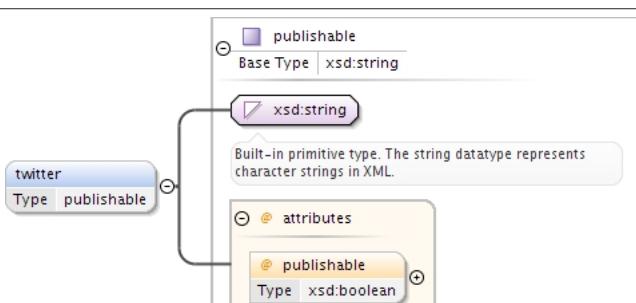
Namespace	No namespace
-----------	--------------

Diagram											
Type	publishable										
Properties	<table border="1"> <tr> <td>content:</td><td>complex</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> <tr> <td>maxOccurs:</td><td>1</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1				
content:	complex										
minOccurs:	0										
maxOccurs:	1										
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr> </thead> <tbody> <tr> <td>publishable</td><td>xsd:boolean</td><td></td><td></td><td>optional</td></tr> </tbody> </table>	QName	Type	Fixed	Default	Use	publishable	xsd:boolean			optional
QName	Type	Fixed	Default	Use							
publishable	xsd:boolean			optional							
Source	<xsd:element name="myspace" type="publishable" maxOccurs="1" minOccurs="0" />										

Element www / homepage

Namespace	No namespace										
Diagram											
Type	publishable										
Properties	<table border="1"> <tr> <td>content:</td><td>complex</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> <tr> <td>maxOccurs:</td><td>1</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1				
content:	complex										
minOccurs:	0										
maxOccurs:	1										
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr> </thead> <tbody> <tr> <td>publishable</td><td>xsd:boolean</td><td></td><td></td><td>optional</td></tr> </tbody> </table>	QName	Type	Fixed	Default	Use	publishable	xsd:boolean			optional
QName	Type	Fixed	Default	Use							
publishable	xsd:boolean			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	<pre> classDiagram class phone { <<publishable>> } class publishable { <<xsd:string>> <<Built-in primitive type. The string datatype represents character strings in XML.>> <<attributes>> attribute publishable { <<xsd:boolean>> } } phone < -- publishable </pre>										
Type	publishable										
Properties	content: complex minOccurs: 0 maxOccurs: 1										
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr> </thead> <tbody> <tr> <td>publishable</td><td>xsd:boolean</td><td></td><td></td><td>optional</td></tr> </tbody> </table>	QName	Type	Fixed	Default	Use	publishable	xsd:boolean			optional
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	<pre> classDiagram information < -- texts texts < -- physical_release_datetime texts < -- digital_release_datetime texts < -- playlength texts < -- num texts < -- setnum texts < -- suggested_prelistening_offset texts < -- origin_country texts < -- main_language texts < -- related </pre> <p>This element contains important data for an item/file. Multilingual promotexts ans teasertexts, dates of physical and...</p>
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	<pre> <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> </pre>
Source	<xsd:element name="information" type="information" />

Element information / texts

Namespace	No namespace
Diagram	<pre> classDiagram texts < -- promotext texts < -- teasertext </pre> <p>This element contains multilingual promotexts ans teasertexts.</p>
Type	texts
Properties	content: complex

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

Element texts / promotext

Namespace	No namespace										
Diagram	<p>The diagram illustrates the schema definition for the <code>promotext</code> element. It is defined as a complex type (<code>promotext</code>) derived from the built-in primitive type <code>xsd:string</code>. The <code>xsd:string</code> type is described as a "Built-in primitive type. The string datatype represents character strings in XML." The <code>promotext</code> type includes an attribute <code>lang</code>, which is also of type <code>xsd:string</code>.</p>										
Type	<code>promotext</code>										
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded				
content:	complex										
minOccurs:	0										
maxOccurs:	unbounded										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><code>lang</code></td> <td><code>xsd:string</code></td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	<code>lang</code>	<code>xsd:string</code>			optional
QName	Type	Fixed	Default	Use							
<code>lang</code>	<code>xsd:string</code>			optional							
Source	<code><xsd:element name="promotext" type="promotext" maxOccurs="unbounded" minOccurs="0" /></code>										

Element texts / teasertext

Namespace	No namespace										
Diagram	<p>The diagram illustrates the schema definition for the <code>teasertext</code> element. It is defined as a complex type (<code>teasertext</code>) derived from the built-in primitive type <code>xsd:string</code>. The <code>xsd:string</code> type is described as a "Built-in primitive type. The string datatype represents character strings in XML." The <code>teasertext</code> type includes an attribute <code>lang</code>, which is also of type <code>xsd:string</code>.</p>										
Type	<code>teasertext</code>										
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded				
content:	complex										
minOccurs:	0										
maxOccurs:	unbounded										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><code>lang</code></td> <td><code>xsd:string</code></td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	<code>lang</code>	<code>xsd:string</code>			optional
QName	Type	Fixed	Default	Use							
<code>lang</code>	<code>xsd:string</code>			optional							
Source	<code><xsd:element name="teasertext" type="teasertext" maxOccurs="unbounded" minOccurs="0" /></code>										

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	 <small>Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...</small>
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

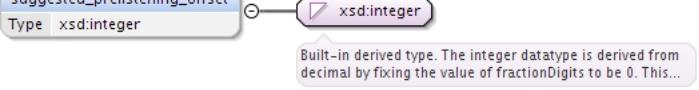
Namespace	No namespace
Diagram	 <small>Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...</small>
Type	xsd:integer
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="num" type="xsd:integer" maxOccurs="1" minOccurs="0" />

Element information / setnum

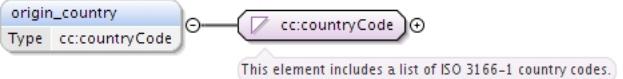
Namespace	No namespace
Diagram	 <small>Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...</small>
Type	xsd:integer

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	 <p>The diagram shows the element 'suggested_prelistening_offset' highlighted in blue. A line connects it to the type 'xsd:integer', which is shown in a purple rounded rectangle. A tooltip below the type says: 'Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...'.</p>
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	 <p>The diagram shows the element 'origin_country' highlighted in blue. A line connects it to the type 'cc:countryCode', which is shown in a purple rounded rectangle. A tooltip below the type says: 'This element includes a list of ISO 3166-1 country codes.'</p>																																																																					
Type	countryCode																																																																					
Properties	content: simple minOccurs: 0 maxOccurs: 1																																																																					
Facets	<table> <tbody> <tr><td>enumeration</td><td>AD</td><td>ANDORRA</td></tr> <tr><td>enumeration</td><td>AE</td><td>UNITED ARAB EMIRATES</td></tr> <tr><td>enumeration</td><td>AF</td><td>AFGHANISTAN</td></tr> <tr><td>enumeration</td><td>AG</td><td>ANTIGUA AND BARBUDA</td></tr> <tr><td>enumeration</td><td>AI</td><td>ANGUILLA</td></tr> <tr><td>enumeration</td><td>AL</td><td>ALBANIA</td></tr> <tr><td>enumeration</td><td>AM</td><td>ARMENIA</td></tr> <tr><td>enumeration</td><td>AN</td><td>NETHERLANDS ANTILLES</td></tr> <tr><td>enumeration</td><td>AO</td><td>ANGOLA</td></tr> <tr><td>enumeration</td><td>AQ</td><td>ANTARCTICA</td></tr> <tr><td>enumeration</td><td>AR</td><td>ARGENTINA</td></tr> <tr><td>enumeration</td><td>AS</td><td>AMERICAN SAMOA</td></tr> <tr><td>enumeration</td><td>AT</td><td>AUSTRIA</td></tr> <tr><td>enumeration</td><td>AU</td><td>AUSTRALIA</td></tr> <tr><td>enumeration</td><td>AW</td><td>ARUBA</td></tr> <tr><td>enumeration</td><td>AX</td><td>ÅLAND ISLANDS</td></tr> <tr><td>enumeration</td><td>AZ</td><td>AZERBAIJAN</td></tr> <tr><td>enumeration</td><td>BA</td><td>BOSNIA AND HERZEGOVINA</td></tr> <tr><td>enumeration</td><td>BB</td><td>BARBADOS</td></tr> <tr><td>enumeration</td><td>BD</td><td>BANGLADESH</td></tr> <tr><td>enumeration</td><td>BE</td><td>BELGIUM</td></tr> <tr><td>enumeration</td><td>BF</td><td>BURKINA FASO</td></tr> <tr><td>enumeration</td><td>BG</td><td>BULGARIA</td></tr> </tbody> </table>	enumeration	AD	ANDORRA	enumeration	AE	UNITED ARAB EMIRATES	enumeration	AF	AFGHANISTAN	enumeration	AG	ANTIGUA AND BARBUDA	enumeration	AI	ANGUILLA	enumeration	AL	ALBANIA	enumeration	AM	ARMENIA	enumeration	AN	NETHERLANDS ANTILLES	enumeration	AO	ANGOLA	enumeration	AQ	ANTARCTICA	enumeration	AR	ARGENTINA	enumeration	AS	AMERICAN SAMOA	enumeration	AT	AUSTRIA	enumeration	AU	AUSTRALIA	enumeration	AW	ARUBA	enumeration	AX	ÅLAND ISLANDS	enumeration	AZ	AZERBAIJAN	enumeration	BA	BOSNIA AND HERZEGOVINA	enumeration	BB	BARBADOS	enumeration	BD	BANGLADESH	enumeration	BE	BELGIUM	enumeration	BF	BURKINA FASO	enumeration	BG	BULGARIA
enumeration	AD	ANDORRA																																																																				
enumeration	AE	UNITED ARAB EMIRATES																																																																				
enumeration	AF	AFGHANISTAN																																																																				
enumeration	AG	ANTIGUA AND BARBUDA																																																																				
enumeration	AI	ANGUILLA																																																																				
enumeration	AL	ALBANIA																																																																				
enumeration	AM	ARMENIA																																																																				
enumeration	AN	NETHERLANDS ANTILLES																																																																				
enumeration	AO	ANGOLA																																																																				
enumeration	AQ	ANTARCTICA																																																																				
enumeration	AR	ARGENTINA																																																																				
enumeration	AS	AMERICAN SAMOA																																																																				
enumeration	AT	AUSTRIA																																																																				
enumeration	AU	AUSTRALIA																																																																				
enumeration	AW	ARUBA																																																																				
enumeration	AX	ÅLAND ISLANDS																																																																				
enumeration	AZ	AZERBAIJAN																																																																				
enumeration	BA	BOSNIA AND HERZEGOVINA																																																																				
enumeration	BB	BARBADOS																																																																				
enumeration	BD	BANGLADESH																																																																				
enumeration	BE	BELGIUM																																																																				
enumeration	BF	BURKINA FASO																																																																				
enumeration	BG	BULGARIA																																																																				

enumeration	BH	BAHRAIN
enumeration	BI	BURUNDI
enumeration	BJ	BENIN
enumeration	BL	SAINT BARTH#LEMY
enumeration	BM	BERMUDA
enumeration	BN	BRUNEI DARUSSALAM
enumeration	BO	BOLIVIA
enumeration	BR	BRAZIL
enumeration	BS	BAHAMAS
enumeration	BT	BHUTAN
enumeration	BV	BOUVET ISLAND
enumeration	BW	BOTSWANA
enumeration	BY	BELARUS
enumeration	BZ	BELIZE
enumeration	CA	CANADA
enumeration	CC	COCOS (KEELING) ISLANDS
enumeration	CD	CONGO, THE DEMOCRATIC REPUBLIC OF THE
enumeration	CF	CENTRAL AFRICAN REPUBLIC
enumeration	CG	CONGO
enumeration	CH	SWITZERLAND
enumeration	CI	COTE D'IVOIRE
enumeration	CK	COOK ISLANDS
enumeration	CL	CHILE
enumeration	CM	CAMEROON
enumeration	CN	CHINA
enumeration	CO	COLOMBIA
enumeration	CR	COSTA RICA
enumeration	CU	CUBA
enumeration	CV	CAPE VERDE
enumeration	CX	CHRISTMAS ISLAND
enumeration	CY	CYPRUS
enumeration	CZ	CZECH REPUBLIC
enumeration	DE	GERMANY
enumeration	DJ	DJIBOUTI
enumeration	DK	DENMARK
enumeration	DM	DOMINICA
enumeration	DO	DOMINICAN REPUBLIC
enumeration	DZ	ALGERIA
enumeration	EC	ECUADOR
enumeration	EE	ESTONIA
enumeration	EG	EGYPT
enumeration	EH	WESTERN SAHARA
enumeration	ER	ERITREA
enumeration	ES	SPAIN
enumeration	ET	ETHIOPIA
enumeration	FI	FINLAND
enumeration	FJ	FIJI
enumeration	FK	FALKLAND ISLANDS (MALVINAS)
enumeration	FM	MICRONESIA, FEDERATED STATES OF

enumeration	FO	FAROE ISLANDS
enumeration	FR	FRANCE
enumeration	GA	GABON
enumeration	GB	UNITED KINGDOM
enumeration	GD	GRENADA
enumeration	GE	GEORGIA
enumeration	GF	FRENCH GUIANA
enumeration	GG	GUERNSEY
enumeration	GH	GHANA
enumeration	GI	GIBRALTAR
enumeration	GL	GREENLAND
enumeration	GM	GAMBIA
enumeration	GN	GUINEA
enumeration	GP	GUADELOUPE
enumeration	GQ	EQUATORIAL GUINEA
enumeration	GR	GREECE
enumeration	GS	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS
enumeration	GT	GUATEMALA
enumeration	GU	GUAM
enumeration	GW	GUINEA-BISSAU
enumeration	GY	GUYANA
enumeration	HK	HONG KONG
enumeration	HM	HEARD ISLAND AND MCDONALD ISLANDS
enumeration	HN	HONDURAS
enumeration	HR	CROATIA
enumeration	HT	HAITI
enumeration	HU	HUNGARY
enumeration	ID	INDONESIA
enumeration	IE	IRELAND
enumeration	IL	ISRAEL
enumeration	IM	ISLE OF MAN
enumeration	IN	INDIA
enumeration	IO	BRITISH INDIAN OCEAN TERRITORY
enumeration	IQ	IRAQ
enumeration	IR	IRAN, ISLAMIC REPUBLIC OF
enumeration	IS	ICELAND
enumeration	IT	ITALY
enumeration	JE	JERSEY
enumeration	JM	JAMAICA
enumeration	JO	JORDAN
enumeration	JP	JAPAN
enumeration	KE	KENYA
enumeration	KG	KYRGYZSTAN
enumeration	KH	CAMBODIA
enumeration	KI	KIRIBATI
enumeration	KM	COMOROS
enumeration	KN	SAINT KITTS AND NEVIS
enumeration	KP	KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF
enumeration	KR	KOREA, REPUBLIC OF

enumeration	KW	KUWAIT
enumeration	KY	CAYMAN ISLANDS
enumeration	KZ	KAZAKHSTAN
enumeration	LA	LAO PEOPLE'S DEMOCRATIC REPUBLIC
enumeration	LB	LEBANON
enumeration	LC	SAINT LUCIA
enumeration	LI	LIECHTENSTEIN
enumeration	LK	SRI LANKA
enumeration	LR	LIBERIA
enumeration	LS	LESOTHO
enumeration	LT	LITHUANIA
enumeration	LU	LUXEMBOURG
enumeration	LV	LATVIA
enumeration	LY	LIBYAN ARAB JAMAHIRIYA
enumeration	MA	MOROCCO
enumeration	MC	MONACO
enumeration	MD	MOLDOVA, REPUBLIC OF
enumeration	ME	MONTENEGRO
enumeration	MF	SAINT MARTIN (FRENCH PART)
enumeration	MG	MADAGASCAR
enumeration	MH	MARSHALL ISLANDS
enumeration	MK	MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF
enumeration	ML	MALI
enumeration	MM	MYANMAR
enumeration	MN	MONGOLIA
enumeration	MO	MACAO
enumeration	MP	NORTHERN MARIANA ISLANDS
enumeration	MQ	MARTINIQUE
enumeration	MR	MAURITANIA
enumeration	MS	MONTSERRAT
enumeration	MT	MALTA
enumeration	MU	MAURITIUS
enumeration	MV	MALDIVES
enumeration	MW	MALAWI
enumeration	MX	MEXICO
enumeration	MY	MALAYSIA
enumeration	MZ	MOZAMBIQUE
enumeration	NA	NAMIBIA
enumeration	NC	NEW CALEDONIA
enumeration	NE	NIGER
enumeration	NF	NORFOLK ISLAND
enumeration	NG	NIGERIA
enumeration	NI	NICARAGUA
enumeration	NL	NETHERLANDS
enumeration	NO	NORWAY
enumeration	NP	NEPAL
enumeration	NR	NAURU
enumeration	NU	NIUE
enumeration	NZ	NEW ZEALAND

enumeration	OM	OMAN
enumeration	PA	PANAMA
enumeration	PE	PERU
enumeration	PF	FRENCH POLYNESIA
enumeration	PG	PAPUA NEW GUINEA
enumeration	PH	PHILIPPINES
enumeration	PK	PAKISTAN
enumeration	PL	POLAND
enumeration	PM	SAINT PIERRE AND MIQUELON
enumeration	PN	PITCAIRN
enumeration	PR	PUERTO RICO
enumeration	PS	PALESTINIAN TERRITORY, OCCUPIED
enumeration	PT	PORTUGAL
enumeration	PW	PALAU
enumeration	PY	PARAGUAY
enumeration	QA	QATAR
enumeration	RE	REUNION
enumeration	RO	ROMANIA
enumeration	RS	SERBIA
enumeration	RU	RUSSIAN FEDERATION
enumeration	RW	RWANDA
enumeration	SA	SAUDI ARABIA
enumeration	SB	SOLOMON ISLANDS
enumeration	SC	SEYCHELLES
enumeration	SD	SUDAN
enumeration	SE	SWEDEN
enumeration	SG	SINGAPORE
enumeration	SH	SAINT HELENA
enumeration	SI	SLOVENIA
enumeration	SJ	SVALBARD AND JAN MAYEN
enumeration	SK	SLOVAKIA
enumeration	SL	SIERRA LEONE
enumeration	SM	SAN MARINO
enumeration	SN	SENEGAL
enumeration	SO	SOMALIA
enumeration	SR	SURINAME
enumeration	ST	SAO TOME AND PRINCIPE
enumeration	SV	EL SALVADOR
enumeration	SY	SYRIAN ARAB REPUBLIC
enumeration	SZ	SWAZILAND
enumeration	TC	TURKS AND CAICOS ISLANDS
enumeration	TD	CHAD
enumeration	TF	FRENCH SOUTHERN TERRITORIES
enumeration	TG	TOGO
enumeration	TH	THAILAND
enumeration	TJ	TAJIKISTAN
enumeration	TK	TOKELAU
enumeration	TL	TIMOR-LESTE
enumeration	TM	TURKMENISTAN

enumeration	TN	TUNISIA
enumeration	TO	TONGA
enumeration	TR	TURKEY
enumeration	TT	TRINIDAD AND TOBAGO
enumeration	TV	TUVALU
enumeration	TW	TAIWAN, PROVINCE OF CHINA
enumeration	TZ	TANZANIA, UNITED REPUBLIC OF
enumeration	UA	UKRAINE
enumeration	UG	UGANDA
enumeration	UM	UNITED STATES MINOR OUTLYING ISLANDS
enumeration	US	UNITED STATES
enumeration	UY	URUGUAY
enumeration	UZ	UZBEKISTAN
enumeration	VA	HOLY SEE (VATICAN CITY STATE)
enumeration	VC	SAINT VINCENT AND THE GRENADINES
enumeration	VE	VENEZUELA, PLURINATIONAL STATE OF
enumeration	VG	VIRGIN ISLANDS, BRITISH
enumeration	VI	VIRGIN ISLANDS, U.S.
enumeration	VN	VIET NAM
enumeration	VU	VANUATU
enumeration	WF	WALLIS AND FUTUNA
enumeration	WS	SAMOA
enumeration	YE	YEMEN
enumeration	YT	MAYOTTE
enumeration	ZA	SOUTH AFRICA
enumeration	ZM	ZAMBIA
enumeration	ZW	ZIMBABWE
Source	<xsd:element name="origin_country" type="cc:countryCode" maxOccurs="1" minOccurs="0" />	

Element information / main_language

Namespace	No namespace	
Diagram		
Type	language	
Properties	content: simple minOccurs: 0 maxOccurs: 1	
Facets	enumeration aa Afar enumeration ab Abkhazian enumeration af Afrikaans enumeration am Amharic enumeration ar Arabic enumeration as Assamese enumeration ay Aymara enumeration az Azerbaijani enumeration ba Bashkir enumeration be Byelorussian enumeration bg Bulgarian	

enumeration	bh	Bihari
enumeration	bi	Bislama
enumeration	bn	Bengali; Bangla
enumeration	bo	Tibetan
enumeration	br	Breton
enumeration	ca	Catalan
enumeration	co	Corsican
enumeration	cs	Czech
enumeration	cy	Welsh
enumeration	da	Danish
enumeration	de	German
enumeration	dz	Bhutani
enumeration	el	Greek
enumeration	en	English
enumeration	eo	Esperanto
enumeration	es	Spanish
enumeration	et	Estonian
enumeration	eu	Basque
enumeration	fa	Persian
enumeration	fi	Finnish
enumeration	fj	Fiji
enumeration	fo	Faroese
enumeration	fr	French
enumeration	fy	Frisian
enumeration	ga	Irish
enumeration	gd	Scots Gaelic
enumeration	gl	Galician
enumeration	gn	Guarani
enumeration	gu	Gujarati
enumeration	ha	Hausa
enumeration	he	Hebrew
enumeration	hi	Hindi
enumeration	hr	Croatian
enumeration	hu	Hungarian
enumeration	hy	Armenian
enumeration	ia	Interlingua
enumeration	id	Indonesian
enumeration	ie	Interlingue
enumeration	ik	Inupiak
enumeration	is	Icelandic
enumeration	it	Italian
enumeration	iu	Inuktitut
enumeration	ja	Japanese
enumeration	jw	Javanese
enumeration	ka	Georgian
enumeration	kk	Kazakh
enumeration	kl	Greenlandic
enumeration	km	Cambodian
enumeration	kn	Kannada

enumeration	ko	Korean
enumeration	ks	Kashmiri
enumeration	ku	Kurdish
enumeration	ky	Kirghiz
enumeration	la	Latin
enumeration	ln	Lingala
enumeration	lo	Laothian
enumeration	lt	Lithuanian
enumeration	lv	Latvian; Lettish
enumeration	mg	Malagasy
enumeration	mi	Maori
enumeration	mk	Macedonian
enumeration	ml	Malayalam
enumeration	mn	Mongolian
enumeration	mo	Moldavian
enumeration	mr	Marathi
enumeration	ms	Malay
enumeration	mt	Maltese
enumeration	my	Burmese
enumeration	na	Nauru
enumeration	ne	Nepali
enumeration	nl	Dutch
enumeration	no	Norwegian
enumeration	oc	Occitan
enumeration	om	(Afan) Oromo
enumeration	or	Oriya
enumeration	pa	Punjabi
enumeration	pl	Polish
enumeration	ps	Pashto, Pushto
enumeration	pt	Portuguese
enumeration	qu	Quechua
enumeration	rm	Rhaeto-Romance
enumeration	rn	Kirundi
enumeration	ro	Romanian
enumeration	ru	Russian
enumeration	rw	Kinyarwanda
enumeration	sa	Sanskrit
enumeration	sd	Sindhi
enumeration	sg	Sangho
enumeration	sh	Serbo-Croatian
enumeration	si	Singhalese
enumeration	sk	Slovak
enumeration	sl	Slovenian
enumeration	sm	Samoan
enumeration	sn	Shona
enumeration	so	Somali
enumeration	sq	Albanian
enumeration	sr	Serbian
enumeration	ss	Siswati

	enumeration	st	Sesotho
	enumeration	su	Sundanese
	enumeration	sv	Swedish
	enumeration	sw	Swahili
	enumeration	ta	Tamil
	enumeration	te	Telugu
	enumeration	tg	Tajik
	enumeration	th	Thai
	enumeration	ti	Tigrinya
	enumeration	tk	Turkmen
	enumeration	tl	Tagalog
	enumeration	tn	Setswana
	enumeration	to	Tonga
	enumeration	tr	Turkish
	enumeration	ts	Tsonga
	enumeration	tt	Tatar
	enumeration	tw	Twi
	enumeration	ug	Uigur
	enumeration	uk	Ukrainian
	enumeration	ur	Urdu
	enumeration	uz	Uzbek
	enumeration	vi	Vietnamese
	enumeration	vo	Volapuk
	enumeration	wo	Wolof
	enumeration	xh	Xhosa
	enumeration	yi	Yiddish
	enumeration	yo	Yoruba
	enumeration	za	Zhuang
	enumeration	zh	Chinese
	enumeration	zu	Zulu
Source	<xsd:element name="main_language" type="l:language" maxOccurs="1" minOccurs="0" />		

Element information / related

Namespace	No namespace
Diagram	<p>This element contains informations of bundles which are related to the bundle of the actual feed.</p>
Type	related
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	physical_distributor*, utube{0,1}, bundle*
Children	bundle, physical_distributor, utube

Instance	<pre><related> <physical_distributor publishable="">{0,unbounded}</physical_distributor> <utube>{0,1}</utube> <bundle>{0,unbounded}</bundle> </related></pre>
Source	<pre><xsd:element name="related" type="related" maxOccurs="1" minOccurs="0"/></pre>

Element related / physical_distributor

Namespace	No namespace										
Diagram	<p>The diagram shows the <code>physical_distributor</code> element as a class. It is derived from the <code>xsd:string</code> base type. It has one attribute, <code>publishable</code>, which is of type <code>xsd:boolean</code>. A callout box provides a detailed description of the <code>xsd:string</code> type.</p>										
Type	physical_distributor										
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded				
content:	complex										
minOccurs:	0										
maxOccurs:	unbounded										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>publishable</td> <td>xsd:boolean</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	publishable	xsd:boolean			optional
QName	Type	Fixed	Default	Use							
publishable	xsd:boolean			optional							
Source	<pre><xsd:element name="physical_distributor" type="physical_distributor" maxOccurs="unbounded" minOccurs="0"/></pre>										

Element related / utube

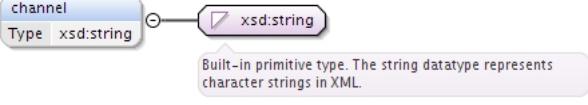
Namespace	No namespace						
Diagram	<p>The diagram shows the <code>utube</code> element as a class. It contains two child elements: <code>url</code> and <code>channel</code>. A callout box provides a general description of the <code>utube</code> element.</p>						
Type	utube						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	ALL(url{0,1} channel{0,1})						
Children	channel, url						
Instance	<pre><utube> <url>{0,1}</url> <channel>{0,1}</channel> </utube></pre>						
Source	<pre><xsd:element name="utube" type="utube" maxOccurs="1" minOccurs="0"/></pre>						

Element utube / url

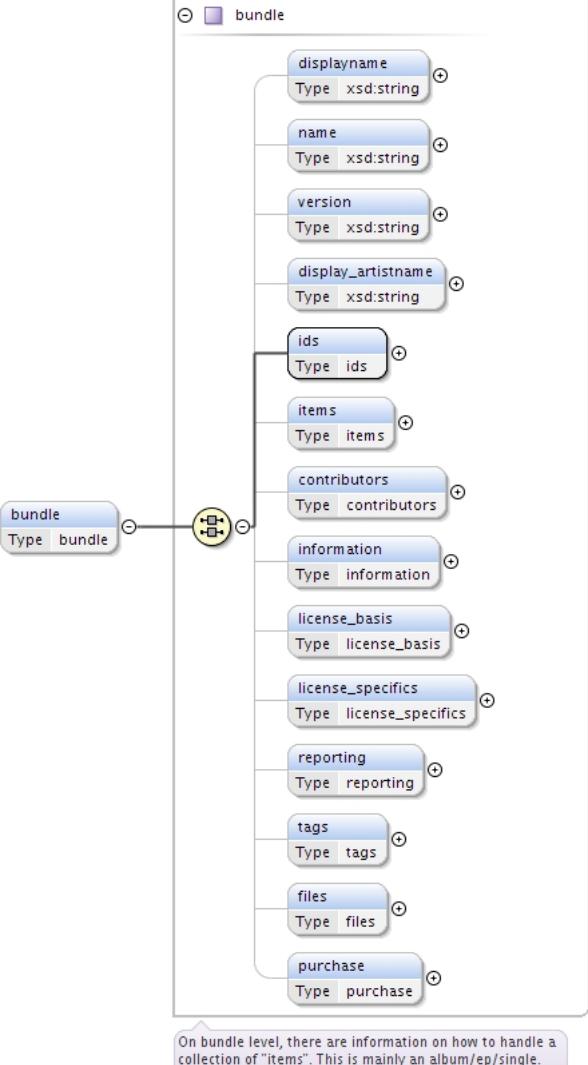
Namespace	No namespace
Diagram	<p>The diagram shows the <code>url</code> element as a class. It is derived from the <code>xsd:string</code> base type. A callout box provides a detailed description of the <code>xsd:string</code> type.</p>

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	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
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	<pre><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></pre>
Source	<code><xsd:element name="bundle" type="bundle" maxOccurs="unbounded" minOccurs="0" /></code>

Element bundle / contributors

Namespace	No namespace
Diagram	<pre> classDiagram class contributors { Type contributors } class contributor { Type contributor } contributors "1..∞" -- "1" contributor note over contributor: This element contains a list of contributor. </pre>
Type	contributors
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	contributor+
Children	contributor
Instance	<pre><contributors> <contributor num="">{1,unbounded}</contributor> </contributors></pre>
Source	<code><xsd:element name="contributors" type="contributors" maxOccurs="1" minOccurs="0" /></code>

Element bundle / information

Namespace	No namespace
-----------	--------------

Diagram	<p>This element contains important data for an item/file. Multilingual promotexts and teasertexts, dates of physical and...</p>						
Type	information						
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> <tr> <td style="padding: 2px;">maxOccurs:</td><td style="padding: 2px;">1</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
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	<pre><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></pre>						
Source	<pre><xsd:element name="information" type="information" maxOccurs="1" minOccurs="0"/></pre>						

Element bundle / license_basis

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram class license_basis { territorial timeframe pricing streaming_allowed channels as_on_bundle } </pre> <p>This element includes the basic rules and information under which this bundle is provided.</p>
Type	license_basis
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
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	<pre> classDiagram class territorial { <> territory } </pre> <p>This Element is a container for territories. There should be a entry for all territories with a attribute if...</p>
Type	territorial
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
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	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>										
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr> </thead> <tbody> <tr> <td>type</td><td>allowance</td><td></td><td></td><td>optional</td></tr> </tbody> </table>	QName	Type	Fixed	Default	Use	type	allowance			optional
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	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
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	<p>pattern</p> $\begin{aligned} &\backslash d\{4\}-\backslash d\{2\}-\backslash d\{2\} \\ &\backslash d\{2\}:\backslash d\{2\}:\backslash d\{2\} \text{ GMT} \\ &+\backslash d\{2\}:\backslash d\{2\} \end{aligned}$
Source	<xsd:element name="from" type="datetimeGMT"/>

Element `timeframe / to`

Namespace	No namespace
Diagram	<pre> graph LR to["to
Type: datetimeGMT"] --> datetimeGMT["datetimeGMT"] </pre>
Type	datetimeGMT
Properties	content: simple
Facets	pattern $\text{\d\{4\}}-\text{\d\{2\}}-\text{\d\{2\}}$ $\text{\d\{2\}}:\text{\d\{2\}}:\text{\d\{2\}} \text{ GMT}\backslash$ $+ \text{\d\{2\}}:\text{\d\{2\}}$
Source	<code><xsd:element name="to" type="datetimeGMT" /></code>

Element `license_basis / pricing`

Namespace	No namespace
Diagram	<pre> graph LR pricing["pricing
Type: pricing"] --> pricecode["pricecode
Type: xsd:string"] pricing --> wholesale["wholesale
Type: xsd:string"] </pre> <p>Pricecode is an arbitrary-info. An explicitly given wholesale-price overrides the basic pricecode-given-wp. Most...</p>
Type	pricing
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(pricecode{0,1} wholesale{0,1})
Children	pricecode, wholesale
Instance	<code><pricing> <pricecode>{0,1}</pricecode> <wholesale>{0,1}</wholesale> </pricing></code>
Source	<code><xsd:element name="pricing" type="pricing" maxOccurs="1" minOccurs="0" /></code>

Element `pricing / pricecode`

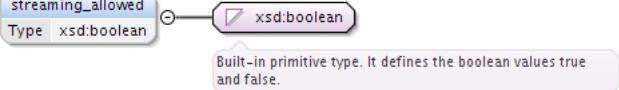
Namespace	No namespace
Diagram	<pre> graph LR pricecode["pricecode
Type: xsd:string"] --> xsdString["xsd:string"] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<code><xsd:element name="pricecode" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>

Element `pricing / wholesale`

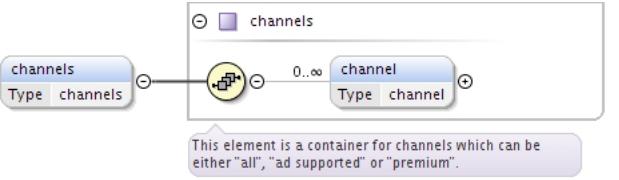
Namespace	No namespace
Diagram	<pre> graph LR wholesale["wholesale
Type: xsd:string"] --> xsdString["xsd:string"] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string

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

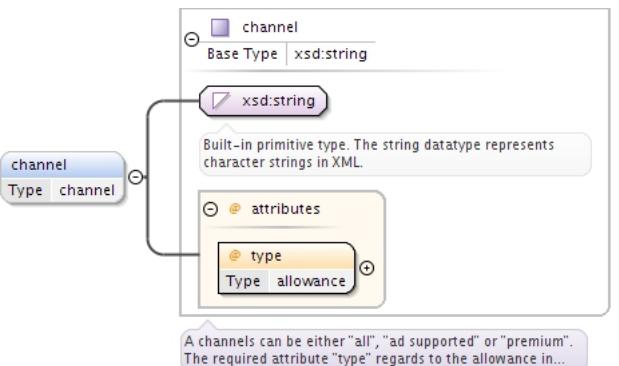
Element license_basis / streaming_allowed

Namespace	No namespace
Diagram	 <pre> graph LR SA[streaming_allowed] --> XB[xsd:boolean] subgraph Info [] XB XB -- "Built-in primitive type. It defines the boolean values true and false." --- NoteXB end </pre>
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	 <pre> graph LR CH[channels] --> CH CH -- "0..∞" --> CH subgraph Info [] CH CH -- "This element is a container for channels which can be either \"all\", \"ad supported\" or \"premium\"." --- NoteCH end </pre>
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	 <pre> graph LR CH[channel] --> XB[xsd:string] subgraph Info [] XB XB -- "Built-in primitive type. The string datatype represents character strings in XML." --- NoteXB XB -- "@ attributes" --- Attrib Attrib -- "@ type" --- Type Type -- "Type allowance" --- Allowance end subgraph Note [] Note --- NoteCH end </pre> <p>A channels can be either "all", "ad supported" or "premium". The required attribute "type" regards to the allowance in...</p>
Type	channel
Properties	content: complex minOccurs: 0

	maxOccurs:	unbounded
Attributes	QName	Type
	type	allowance
Source	<xsd:element name="channel" type="channel" maxOccurs="unbounded" minOccurs="0" />	

Element license_basis / as_on_bundle

Namespace	No namespace						
Diagram	<p>The diagram shows a node labeled "as_on_bundle" with a "Type" of "xsd:boolean". A line connects it to another node labeled "xsd:boolean". A callout bubble indicates: "Built-in primitive type. It defines the boolean values true and false."</p>						
Type	xsd:boolean						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<xsd:element name="as_on_bundle" type="xsd:boolean" maxOccurs="1" minOccurs="0" />						

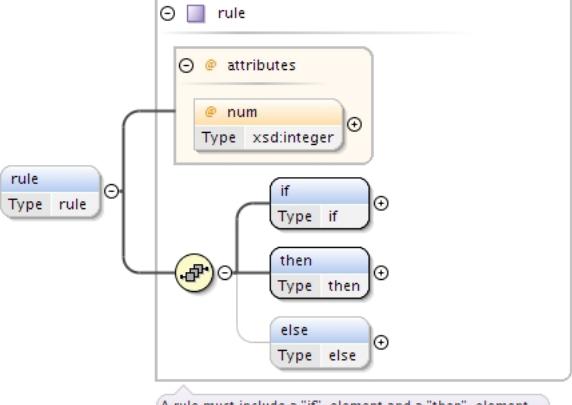
Element bundle / license_specifics

Namespace	No namespace						
Diagram	<p>The diagram shows a node labeled "license_specifics" with a "Type" of "license_specifics". A line connects it to another node labeled "rules" with a "Type" of "rules". A callout bubble indicates: "This element includes specific rules which should be applied. This can be achieved by given rules."</p>						
Type	license_specifics						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
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" />						

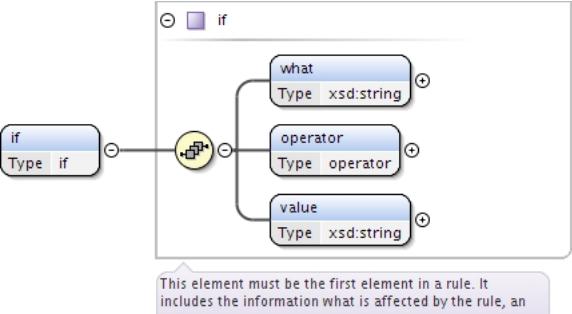
Element license_specifics / rules

Namespace	No namespace
Diagram	<p>The diagram shows a node labeled "rules" with a "Type" of "rules". A line connects it to another node labeled "rule" with a "Type" of "rule". A callout bubble indicates: "This element is a container for rules. It needs an ordered mode here - first come first match."</p>
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	<pre><rule num=""> <if>{1,1}</if> <then>{1,1}</then> <else>{0,1}</else> </rule></pre>										
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr> </thead> <tbody> <tr> <td>num</td><td>xsd:integer</td><td></td><td></td><td>optional</td></tr> </tbody> </table>	QName	Type	Fixed	Default	Use	num	xsd:integer			optional
QName	Type	Fixed	Default	Use							
num	xsd:integer			optional							
Source	<code><xsd:element name="rule" type="rule" maxOccurs="unbounded" minOccurs="0" /></code>										

Element rule / if

Namespace	No namespace
Diagram	
Type	if
Properties	content: complex
Model	what , operator , value
Children	operator, value, what
Instance	<pre><if> <what>{1,1}</what> <operator>{1,1}</operator> <value>{1,1}</value> </if></pre>
Source	<code><xsd:element name="if" type="if" /></code>

Element if / what

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "what" with a small "Type" label below it. An arrow points from "what" to a purple rounded rectangle labeled "xsd:string". A callout box below "xsd:string" states: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="what" type="xsd:string" />

Element if / operator

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "operator" with a small "Type" label below it. An arrow points from "operator" to a purple rounded rectangle labeled "operator".</p>
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	<p>The diagram shows a blue rounded rectangle labeled "value" with a small "Type" label below it. An arrow points from "value" to a purple rounded rectangle labeled "xsd:string". A callout box below "xsd:string" states: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="value" type="xsd:string" />

Element rule / then

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "then" with a small "Type" label below it. An arrow points from "then" to a purple rounded rectangle labeled "echo". Another arrow points from "echo" to a purple rounded rectangle labeled "break". A callout box below "echo" states: "This element must be the second in a rule and includes information 'echo' for debugging output and can include an..."</p>
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	<p>The diagram shows a blue rounded rectangle labeled "echo" with a small "Type" label below it. A line connects "echo" to a purple rounded rectangle labeled "xsd:string". A callout bubble points to "xsd:string" with the text: "Built-in primitive type. The string datatype represents character strings in XML."</p>						
Type	xsd:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
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	<p>The diagram shows a blue rounded rectangle labeled "break" with a small "Type" label below it.</p>				
Properties	<table border="1"> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	minOccurs:	0	maxOccurs:	1
minOccurs:	0				
maxOccurs:	1				
Source	<xsd:element name="break" maxOccurs="1" minOccurs="0" />				

Element rule / else

Namespace	No namespace						
Diagram	<p>The diagram shows a blue rounded rectangle labeled "else" with a small "Type" label below it. A line connects "else" to a yellow rounded rectangle labeled "proclaim". Another line connects "else" to a blue rounded rectangle labeled "break". A callout bubble points to "proclaim" with the text: "This element is optional. It includes information 'proclaim' and can include an element 'break' which means to not..."</p>						
Type	else						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	proclaim*, break{0,1}						
Children	break, proclaim						
Instance	<pre><else> <proclaim>{0 ,unbounded}</proclaim> <break>{0,1}</break> </else></pre>						
Source	<xsd:element name="else" type="else" maxOccurs="1" minOccurs="0" />						

Element else / proclaim

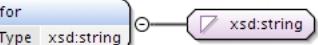
Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "proclaim" with a small "Type" label below it. A line connects "proclaim" to a yellow rounded rectangle labeled "what". Another line connects "proclaim" to a blue rounded rectangle labeled "for". A callout bubble points to "what" with the text: "This element includes the information what is affected and the corresponding value."</p>

Type	proclaim
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	what , for
Children	for, what
Instance	<pre><proclaim> <what>{1,1}</what> <for>{1,1}</for> </proclaim></pre>
Source	<code><xsd:element name="proclaim" type="proclaim" maxOccurs="unbounded" minOccurs="0" /></code>

Element proclaim / what

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="what" type="xsd:string" /></code>

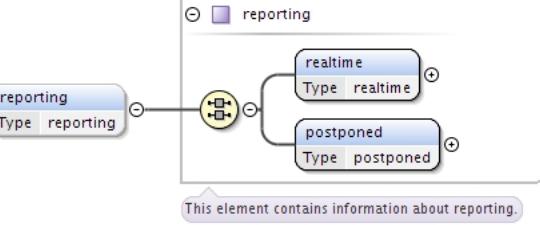
Element proclaim / for

Namespace	No namespace
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="for" type="xsd:string" /></code>

Element else / break

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

Element bundle / reporting

Namespace	No namespace
Diagram	 <p>This element contains information about reporting.</p>
Type	reporting
Properties	content: complex minOccurs: 0

	maxOccurs: 1
Model	ALL(realtime postponed)
Children	postponed, realtime
Instance	<pre><reporting> <realtime>{1,1}</realtime> <postponed>{1,1}</postponed> </reporting></pre>
Source	<code><xsd:element name="reporting" type="reporting" maxOccurs="1" minOccurs="0"/></code>

Element reporting / realtime

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

Element realtime / http

Namespace	No namespace
Diagram	
Type	http
Type hierarchy	<ul style="list-style-type: none"> • action • http
Properties	content: complex
Model	ALL(url type addheader addparams)
Children	addheader, addparams, type, url
Instance	<pre><http> <url>{1,1}</url> <type>{1,1}</type> <addheader>{1,1}</addheader> <addparams>{1,1}</addparams> </http></pre>
Source	<code><xsd:element name="http" type="http" /></code>

Element reporting / postponed

Namespace	No namespace
Diagram	<pre> graph TD postpone["postpone Type postponed"] --> postpone postpone --- id["id Type xsd:string"] </pre> <p>This element contains some info on reporting when doing the "usual" time-gap-reporting.</p>
Type	postpone
Properties	content: complex
Model	id
Children	id
Instance	<pre> <postpone> <id>{1,1}</id> </postpone> </pre>
Source	<pre><xsd:element name="postpone" type="postpone"/></pre>

Element postponed / id

Namespace	No namespace
Diagram	<pre> graph TD id["id Type xsd:string"] --> id id --- xsdstring["xsd:string"] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Properties	content: simple
Source	<pre><xsd:element name="id" type="xsd:string"/></pre>

Element bundle / tags

Namespace	No namespace
Diagram	<pre> graph TD tags["tags Type tags"] --> tags tags --- genres["genres Type genres"] genres --- bundleonly["bundle_only Type xsd:boolean"] genres --- explicitlyrics["explicit_lyrics Type explicitLyrics"] genres --- live["live Type xsd:boolean"] genres --- acoustic["acoustic Type xsd:boolean"] genres --- instrumental["instrumental Type xsd:boolean"] </pre> <p>This element contains information about genres and more.</p>
Type	tags
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	ALL(genres{0,1} bundle_only{0,1} explicit_lyrics{0,1} live{0,1} acoustic{0,1} instrumental{0,1})
Children	acoustic, bundle_only, explicit_lyrics, genres, instrumental, live
Instance	<pre> <tags> <genres>{0,1}</genres> </pre>

	<pre><bundle_only>{0,1}</bundle_only> <explicit_lyrics>{0,1}</explicit_lyrics> <live>{0,1}</live> <acoustic>{0,1}</acoustic> <instrumental>{0,1}</instrumental> </tags></pre>
Source	<code><xsd:element name="tags" type="tags" maxOccurs="1" minOccurs="0" /></code>

Element tags / genres

Namespace	No namespace
Diagram	<p>This element contains a list of genres.</p>
Type	genres
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	genre*
Children	genre
Instance	<pre><genres> <genre>{0,unbounded}</genre> </genres></pre>
Source	<code><xsd:element name="genres" type="genres" maxOccurs="1" minOccurs="0" /></code>

Element genres / genre

Namespace	No namespace
Diagram	<p>This element includes a list of ISO 3166-1 country codes.</p>
Type	genre
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Facets	<p>enumeration Rock</p> <p>enumeration Beat</p> <p>enumeration Blues Rock</p> <p>enumeration Rock'n'Roll</p> <p>enumeration Art Rock</p> <p>enumeration Classic Rock</p> <p>enumeration Experimental Rock</p> <p>enumeration Glam Rock</p> <p>enumeration Hard Rock</p> <p>enumeration Krautrock</p> <p>enumeration Progressive</p> <p>enumeration Psychedelic Rock</p> <p>enumeration Psychobilly</p> <p>enumeration Rockabilly</p> <p>enumeration Soft Rock</p> <p>enumeration Southern Rock</p> <p>enumeration Surf</p>

enumeration	Alternative
enumeration	Crossover
enumeration	Dark Wave
enumeration	Garage Rock
enumeration	Goth / Industrial
enumeration	Grunge
enumeration	Hardcore
enumeration	Indie Rock
enumeration	New Wave
enumeration	Punk
enumeration	Black Metal
enumeration	Death Metal
enumeration	Heavy Metal
enumeration	Power Metal
enumeration	Thrash / Speed Metal
enumeration	Doom Metal
enumeration	Grind Core
enumeration	Pop
enumeration	Dance Pop
enumeration	Deutschpop
enumeration	Disco
enumeration	Electropop
enumeration	Euro Dance
enumeration	Euro Pop
enumeration	French Pop
enumeration	Indie Pop
enumeration	Italo Pop
enumeration	J-Pop
enumeration	K-Pop
enumeration	Neue Deutsche Welle
enumeration	Pop Rock
enumeration	Power Pop
enumeration	Schlager
enumeration	Singer/Songwriter
enumeration	Synthpop
enumeration	Teen Pop
enumeration	Country
enumeration	Alternative Country
enumeration	Bluegrass
enumeration	Contemporary Folk
enumeration	Country Gospel
enumeration	Honky-Tonk
enumeration	Jewish / Yiddish Music
enumeration	Nashville Sound
enumeration	Outlaw / Progressive Country
enumeration	Texas Country
enumeration	Traditional Country
enumeration	Western Swing

enumeration	Folk
enumeration	Americana
enumeration	Folk Rock
enumeration	Irish Folk
enumeration	Jazz
enumeration	Acid Jazz
enumeration	Avantgarde
enumeration	Bebop
enumeration	Big Band
enumeration	Cajun / Zydeco
enumeration	Classic
enumeration	Cool Jazz
enumeration	Free jazz
enumeration	Hard Bop
enumeration	Jazz Fusion
enumeration	New Orleans Jazz
enumeration	Nu-Jazz
enumeration	Smooth Jazz
enumeration	Swing
enumeration	Vocal Jazz
enumeration	HipHop
enumeration	Alternative Hip Hop
enumeration	Crunk
enumeration	G-Funk
enumeration	Gangsta Rap
enumeration	Golden Era
enumeration	Grime
enumeration	Hyphy
enumeration	Instrumental
enumeration	New School
enumeration	Old School
enumeration	Turntablism
enumeration	US Eastcoast
enumeration	US Midwest
enumeration	US Southern
enumeration	US Westcoast
enumeration	Blues
enumeration	Boogie-Woogie
enumeration	Electric Blues Guitar
enumeration	Modern Blues
enumeration	Regional Blues
enumeration	Traditional Blues
enumeration	Soul
enumeration	Motown Sound
enumeration	Neo Soul
enumeration	Philly Sound
enumeration	Funk
enumeration	R&B
enumeration	Contemporary R&B

enumeration	Electronic
enumeration	Ambient
enumeration	Chill Out
enumeration	Lounge
enumeration	Downbeat
enumeration	Electronica
enumeration	Indie Disco
enumeration	Industrial / EBM
enumeration	Techno
enumeration	Dance
enumeration	Electro
enumeration	House
enumeration	Deep House
enumeration	Disco House
enumeration	Electro House
enumeration	Hard House
enumeration	Progressive House
enumeration	Soulful House
enumeration	Tech House
enumeration	Vocal House
enumeration	Big Beat
enumeration	Breakbeat
enumeration	Drum'n'Bass
enumeration	Dubstep
enumeration	Garage / UK Funky
enumeration	IDM
enumeration	Trip-Hop
enumeration	Trance
enumeration	Goa Trance
enumeration	Hard Trance
enumeration	Psychedelic Trance
enumeration	Gabba
enumeration	Jumpstyle / Hardstyle
enumeration	Classic
enumeration	12-Tone Composition
enumeration	Baroque
enumeration	Chamber Music
enumeration	Crossover / Popular Classicism
enumeration	Electronic Music / Computer Music
enumeration	Minimal Music
enumeration	Modern, 20th / 21st Century
enumeration	Music and other Media / Arts
enumeration	Music and Word
enumeration	Neo-Romanticism
enumeration	Neoclassicism
enumeration	New Music / Contemporary Music

enumeration	Postmodern Music
enumeration	Renaissance
enumeration	Romantic
enumeration	Serialism
enumeration	Waltz
enumeration	Ballet
enumeration	Brass Ensemble
enumeration	Concerto / Solo Instrument with Orchestra
enumeration	Mixed Ensemble (Strings / Wind)
enumeration	Mixed Wind Ensemble (Woodwind / Brass)
enumeration	Opera Arias
enumeration	Opera Baroque
enumeration	Opera Classical
enumeration	Opera Renaissance
enumeration	Opera Romantic
enumeration	Operettas
enumeration	Several Solo Instruments
enumeration	Solo Instrument
enumeration	String Ensemble
enumeration	String Orchestra
enumeration	String Quartet
enumeration	String Trio
enumeration	Symphonic Music / Orchestral Music
enumeration	Symphonies
enumeration	Vocal Ensemble
enumeration	Vocal Music
enumeration	Woodwind Ensemble
enumeration	Choir
enumeration	Boy's Choir
enumeration	Children's Choir
enumeration	Choir with Orchestra
enumeration	Female Choir
enumeration	Male Choir
enumeration	Mixed Choir
enumeration	Flute
enumeration	Recorder / English Flute
enumeration	Oboe
enumeration	Clarinet
enumeration	Bassoon
enumeration	Saxophone
enumeration	Horn
enumeration	Trumpet
enumeration	Trombone
enumeration	Tuba

enumeration	Violin
enumeration	Viola
enumeration	Violoncello
enumeration	Double Bass
enumeration	Guitar
enumeration	Mandolin
enumeration	Lute
enumeration	Harp
enumeration	Piano
enumeration	Harpsichord
enumeration	Accordion
enumeration	Organ
enumeration	Ancient Instruments
enumeration	Percussion (Vibraphone etc.)
enumeration	Miscellaneous Lead Instrument
enumeration	Reggae
enumeration	Contemporary
enumeration	Dancehall
enumeration	Dub
enumeration	Lover's Rock
enumeration	Reggaeton
enumeration	Roots
enumeration	Ska
enumeration	World
enumeration	African Music
enumeration	Afrobeat
enumeration	Asian and Middle Eastern Music
enumeration	Calypso
enumeration	Caribbean Music
enumeration	Celtic, European and Judaica Music
enumeration	Coupé Decalé
enumeration	Griot
enumeration	Highlife
enumeration	Kuduro
enumeration	Kwaito
enumeration	Makossa
enumeration	Marching Band
enumeration	Mento
enumeration	North American Music
enumeration	Parang
enumeration	Polka
enumeration	Rai
enumeration	Soca
enumeration	Soukous
enumeration	Zouk
enumeration	Latin

enumeration	Banda
enumeration	Bolero
enumeration	Bossa Nova
enumeration	Corridos
enumeration	Cumbia
enumeration	Fado
enumeration	Flamenco
enumeration	Grupero
enumeration	Mambo
enumeration	Mariachi
enumeration	Merengue
enumeration	Norteno
enumeration	Ranchero
enumeration	Rock En Espanol
enumeration	Salsa
enumeration	Samba
enumeration	Son Cubana
enumeration	Sonidero
enumeration	Tango
enumeration	Tejano
enumeration	Religious
enumeration	Christian Rock
enumeration	Christian Hip Hop
enumeration	Christian Pop
enumeration	Chants
enumeration	Gospel
enumeration	Hymn
enumeration	Mass
enumeration	Spiritual
enumeration	Worship
enumeration	Miscellaneous
enumeration	Anime / Video Game Soundtracks
enumeration	Bollywood
enumeration	Instrumental
enumeration	Karaoke
enumeration	Movie Scores
enumeration	Movie Soundtracks
enumeration	Sound Effects
enumeration	Soundtrack
enumeration	TV Soundtrack
enumeration	unclassifiable
enumeration	Wedding Music
enumeration	Holiday
enumeration	Word
enumeration	Business & Career
enumeration	Abstracts & Dossiers
enumeration	Accounting
enumeration	Business & Investing

enumeration	Communication
enumeration	Computers & Internet
enumeration	Economics
enumeration	Finance
enumeration	Management & Leadership
enumeration	Marketing & Sales
enumeration	Politics
enumeration	Self-Help
enumeration	Self-Organization
enumeration	Skills
enumeration	Small Business & Entrepeneurship
enumeration	Children's Audiobooks
enumeration	age
enumeration	Age
enumeration	Popular Characters
enumeration	Animal Stories
enumeration	Children's Book Classics
enumeration	Detective Stories
enumeration	Fairy Tales
enumeration	Fantasy & Spook
enumeration	Knowledge for Children
enumeration	Pirates, Knights & Historical
enumeration	Poems & Song
enumeration	Comedy & Humour
enumeration	Comedy & Cabaret
enumeration	Humoristic Novel
enumeration	Crime
enumeration	Detective Stories
enumeration	Detective Stories „Noir“
enumeration	Classic Detective Stories
enumeration	Scandinavian Detective Stories
enumeration	Temporary Detective Stories
enumeration	Education & Knowledge
enumeration	Art & Culture
enumeration	Biography & Memento
enumeration	Foreign Language
enumeration	History
enumeration	Philosophy
enumeration	Politics & Current Affairs
enumeration	Science & Technology
enumeration	Health, Mind & Body
enumeration	Autogenous Training
enumeration	Creativity

enumeration	Fitness
enumeration	Health
enumeration	Lifestyle
enumeration	Love & Erotic
enumeration	Meditation / Yoga
enumeration	Memory Training
enumeration	Mental Training
enumeration	Motivation
enumeration	Philosophy
enumeration	Positive Thinking & Attitude
enumeration	Psychology
enumeration	Spirituality & Religion
enumeration	Sports
enumeration	Wellness & Beauty
enumeration	Science Fiction & Fantasy
enumeration	Ancient World
enumeration	Fantasy-Romance
enumeration	Historical Thriller
enumeration	Horror Classics
enumeration	Medieval Times & Early Modern Era
enumeration	Thriller
enumeration	Mystery & Conspiracy
enumeration	Psychological Thriller
enumeration	Espionage, Politics & Justice
enumeration	Vatican & Secret Societies
enumeration	Science & Medicine
enumeration	Novels
enumeration	Erotica
enumeration	Contemporary German Literature
enumeration	Romance
enumeration	Entertainment
enumeration	Contemporary Literature
enumeration	Youth
enumeration	Detective Stories
enumeration	Fantasy
enumeration	For Girls
enumeration	Knowledge for Teenagers
enumeration	Mystery
enumeration	Pirates, Knights & Historical
enumeration	Youth Classics
enumeration	Youth Today
enumeration	Language

enumeration	Albanian
enumeration	Arabic
enumeration	Bengali
enumeration	Bosnian
enumeration	Bulgarian
enumeration	Cantonese/Yue
enumeration	Croatian
enumeration	Czech
enumeration	Danish
enumeration	Dutch
enumeration	English
enumeration	Finnish
enumeration	French
enumeration	German
enumeration	Greek
enumeration	Hebrew
enumeration	Hindi/Urdu
enumeration	Hungarian
enumeration	Italian
enumeration	Japanese
enumeration	Korean
enumeration	Macedonian
enumeration	Mandarin
enumeration	Norwegian
enumeration	Patois
enumeration	Portuguese
enumeration	Russian
enumeration	Serbian
enumeration	Spanish
enumeration	Swedish
enumeration	Tamil
enumeration	Turkish
enumeration	Vietnamese
enumeration	Film
enumeration	Action
enumeration	3D
enumeration	Adventure
enumeration	Animation
enumeration	Author's Film
enumeration	Biography
enumeration	Cartoon
enumeration	Children
enumeration	Comedy
enumeration	Crime & Gangster
enumeration	Disaster
enumeration	Documentary
enumeration	Drama
enumeration	Epic / Historical
enumeration	Erotic

enumeration	Expressionism
enumeration	Family
enumeration	Fantasy
enumeration	Film-Noir
enumeration	GLBT
enumeration	Horror
enumeration	Independent Film
enumeration	Martial-Arts / Eastern
enumeration	Monumental
enumeration	Musical / Dance
enumeration	Music
enumeration	Mystery
enumeration	Reality-TV
enumeration	Romantic
enumeration	Science Fiction
enumeration	Silent Movie
enumeration	Sport
enumeration	Thriller
enumeration	TV-Series
enumeration	Tragicomedy
enumeration	War / Anti-War
enumeration	Western
enumeration	Youth
enumeration	Country
enumeration	United Arab Emirates (AE)
enumeration	Afghanistan (AF)
enumeration	Antigua and Barbuda (AG)
enumeration	Anguilla (AI)
enumeration	Albania (AL)
enumeration	Armenia (AM)
enumeration	Angola (AO)
enumeration	Antarctica (AQ)
enumeration	Argentina (AR)
enumeration	American Samoa (AS)
enumeration	Austria (AT)
enumeration	Australia (AU)
enumeration	Aruba (AW)
enumeration	Åland Islands (AX)
enumeration	Azerbaijan (AZ)
enumeration	Bosnia and Herzegovina (BA)
enumeration	Barbados (BB)
enumeration	Bangladesh (BD)
enumeration	Belgium (BE)
enumeration	Burkina Faso (BF)
enumeration	Bulgaria (BG)
enumeration	Bahrain (BH)
enumeration	Burundi (BI)

enumeration	Benin (BJ)
enumeration	Saint Barthélemy (BL)
enumeration	Bermuda (BM)
enumeration	Brunei Darussalam (BN)
enumeration	Bolivia Plurinational State of (BO)
enumeration	Bonaire Saint Eustatius and Saba (BQ)
enumeration	Brazil (BR)
enumeration	Bahamas (BS)
enumeration	Bhutan (BT)
enumeration	Bouvet Island (BV)
enumeration	Botswana (BW)
enumeration	Belarus (BY)
enumeration	Belize (BZ)
enumeration	Canada (CA)
enumeration	Cocos (Keeling) Islands (CC)
enumeration	Congo the Democratic Republic of the (CD)
enumeration	Central African Republic (CF)
enumeration	Congo (CG)
enumeration	Switzerland (CH)
enumeration	Côte d'Ivoire (CI)
enumeration	Cook Islands (CK)
enumeration	Chile (CL)
enumeration	Cameroon (CM)
enumeration	China (CN)
enumeration	Colombia (CO)
enumeration	Costa Rica (CR)
enumeration	Cuba (CU)
enumeration	Cape Verde (CV)
enumeration	Curaçao (CW)
enumeration	Christmas Island (CX)
enumeration	Cyprus (CY)
enumeration	Czech Republic (CZ)
enumeration	Germany (DE)
enumeration	Djibouti (DJ)
enumeration	Denmark (DK)
enumeration	Dominica (DM)
enumeration	Dominican Republic (DO)
enumeration	Algeria (DZ)
enumeration	Ecuador (EC)
enumeration	Estonia (EE)
enumeration	Egypt (EG)
enumeration	Western Sahara (EH)
enumeration	Eritrea (ER)
enumeration	Spain (ES)

enumeration	Ethiopia (ET)
enumeration	Finland (FI)
enumeration	Fiji (FJ)
enumeration	Falkland Islands (Malvinas) (FK)
enumeration	Micronesia Federated States of (FM)
enumeration	Faroe Islands (FO)
enumeration	France (FR)
enumeration	Gabon (GA)
enumeration	United Kingdom (GB)
enumeration	Grenada (GD)
enumeration	Georgia (GE)
enumeration	French Guiana (GF)
enumeration	Guernsey (GG)
enumeration	Ghana (GH)
enumeration	Gibraltar (GI)
enumeration	Greenland (GL)
enumeration	Gambia (GM)
enumeration	Guinea (GN)
enumeration	Guadeloupe (GP)
enumeration	Equatorial Guinea (GQ)
enumeration	Greece (GR)
enumeration	South Georgia and the South Sandwich Islands (GS)
enumeration	Guatemala (GT)
enumeration	Guam (GU)
enumeration	Guinea-Bissau (GW)
enumeration	Guyana (GY)
enumeration	Hong Kong (HK)
enumeration	Heard Island and McDonald Islands (HM)
enumeration	Honduras (HN)
enumeration	Croatia (HR)
enumeration	Haiti (HT)
enumeration	Hungary (HU)
enumeration	Indonesia (ID)
enumeration	Ireland (IE)
enumeration	Israel (IL)
enumeration	Isle of Man (IM)
enumeration	India (IN)
enumeration	British Indian Ocean Territory (IO)
enumeration	Iraq (IQ)
enumeration	Iran Islamic Republic of (IR)
enumeration	Iceland (IS)
enumeration	Italy (IT)
enumeration	Jersey (JE)
enumeration	Jamaica (JM)

enumeration	Jordan (JO)
enumeration	Japan (JP)
enumeration	Kenya (KE)
enumeration	Kyrgyzstan (KG)
enumeration	Cambodia (KH)
enumeration	Kiribati (KI)
enumeration	Comoros (KM)
enumeration	Saint Kitts and Nevis (KN)
enumeration	Korea Democratic People's Republic of (KP)
enumeration	Korea Republic of (KR)
enumeration	Kuwait (KW)
enumeration	Cayman Islands (KY)
enumeration	Kazakhstan (KZ)
enumeration	Lao People's Democratic Republic (LA)
enumeration	Lebanon (LB)
enumeration	Saint Lucia (LC)
enumeration	Liechtenstein (LI)
enumeration	Sri Lanka (LK)
enumeration	Liberia (LR)
enumeration	Lesotho (LS)
enumeration	Lithuania (LT)
enumeration	Luxembourg (LU)
enumeration	Latvia (LV)
enumeration	Libyan Arab Jamahiriya (LY)
enumeration	Morocco (MA)
enumeration	Monaco (MC)
enumeration	Moldova Republic of (MD)
enumeration	Montenegro (ME)
enumeration	Saint Martin (French part) (MF)
enumeration	Madagascar (MG)
enumeration	Marshall Islands (MH)
enumeration	Macedonia the former Yugoslav Republic of (MK)
enumeration	Mali (ML)
enumeration	Myanmar (MM)
enumeration	Mongolia (MN)
enumeration	Macao (MO)
enumeration	Northern Mariana Islands (MP)
enumeration	Martinique (MQ)
enumeration	Mauritania (MR)
enumeration	Montserrat (MS)
enumeration	Malta (MT)
enumeration	Mauritius (MU)

enumeration	Maldives (MV)
enumeration	Malawi (MW)
enumeration	Mexico (MX)
enumeration	Malaysia (MY)
enumeration	Mozambique (MZ)
enumeration	Namibia (NA)
enumeration	New Caledonia (NC)
enumeration	Niger (NE)
enumeration	Norfolk Island (NF)
enumeration	Nigeria (NG)
enumeration	Nicaragua (NI)
enumeration	Netherlands (NL)
enumeration	Norway (NO)
enumeration	Nepal (NP)
enumeration	Nauru (NR)
enumeration	Niue (NU)
enumeration	New Zealand (NZ)
enumeration	Oman (OM)
enumeration	Panama (PA)
enumeration	Peru (PE)
enumeration	French Polynesia (PF)
enumeration	Papua New Guinea (PG)
enumeration	Philippines (PH)
enumeration	Pakistan (PK)
enumeration	Poland (PL)
enumeration	Saint Pierre and Miquelon (PM)
enumeration	Pitcairn (PN)
enumeration	Puerto Rico (PR)
enumeration	Palestinian Territory Occupied (PS)
enumeration	Portugal (PT)
enumeration	Palau (PW)
enumeration	Paraguay (PY)
enumeration	Qatar (QA)
enumeration	Réunion (RE)
enumeration	Romania (RO)
enumeration	Serbia (RS)
enumeration	Russian Federation (RU)
enumeration	Rwanda (RW)
enumeration	Saudi Arabia (SA)
enumeration	Solomon Islands (SB)
enumeration	Seychelles (SC)
enumeration	Sudan (SD)
enumeration	Sweden (SE)
enumeration	Singapore (SG)
enumeration	Saint Helena Ascension and Tristan da Cunha (SH)
enumeration	Slovenia (SI)

enumeration	Svalbard and Jan Mayen (SJ)
enumeration	Slovakia (SK)
enumeration	Sierra Leone (SL)
enumeration	San Marino (SM)
enumeration	Senegal (SN)
enumeration	Somalia (SO)
enumeration	Suriname (SR)
enumeration	Sao Tome and Principe (ST)
enumeration	El Salvador (SV)
enumeration	Sint Maarten (Dutch part) (SX)
enumeration	Syrian Arab Republic (SY)
enumeration	Swaziland (SZ)
enumeration	Turks and Caicos Islands (TC)
enumeration	Chad (TD)
enumeration	French Southern Territories (TF)
enumeration	Togo (TG)
enumeration	Thailand (TH)
enumeration	Tajikistan (TJ)
enumeration	Tokelau (TK)
enumeration	Timor-Leste (TL)
enumeration	Turkmenistan (TM)
enumeration	Tunisia (TN)
enumeration	Tonga (TO)
enumeration	Turkey (TR)
enumeration	Trinidad and Tobago (TT)
enumeration	Tuvalu (TV)
enumeration	Taiwan Province of China (TW)
enumeration	Tanzania United Republic of (TZ)
enumeration	Ukraine (UA)
enumeration	Uganda (UG)
enumeration	United States Minor Outlying Islands (UM)
enumeration	United States (US)
enumeration	Uruguay (UY)
enumeration	Uzbekistan (UZ)
enumeration	Holy See (Vatican City State) (VA)
enumeration	Saint Vincent and the Grenadines (VC)
enumeration	Venezuela Bolivarian Republic of (VE)
enumeration	Virgin Islands British (VG)
enumeration	Virgin Islands U.S. (VI)

	enumeration Viet Nam (VN)
	enumeration Vanuatu (VU)
	enumeration Wallis and Futuna (WF)
	enumeration Samoa (WS)
	enumeration Yemen (YE)
	enumeration Mayotte (YT)
	enumeration South Africa (ZA)
	enumeration Zambia (ZM)
	enumeration Zimbabwe (ZW)
Source	<xsd:element name="genre" type="g:genre" 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

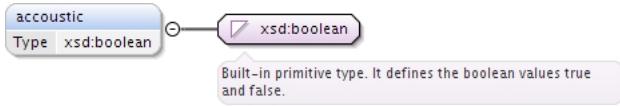
Namespace	No namespace
Diagram	
Type	explicitLyrics
Properties	content: simple minOccurs: 0 maxOccurs: 1
Facets	enumeration true enumeration false enumeration cleaned
Source	<xsd:element name="explicit_lyrics" type="explicitLyrics" maxOccurs="1" minOccurs="0"/>

Element tags / live

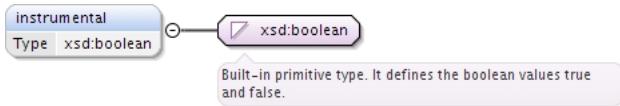
Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="live" type="xsd:boolean" maxOccurs="1" minOccurs="0"/>

Element tags / acoustic

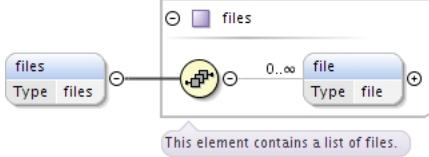
Namespace	No namespace
-----------	--------------

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

Element tags / instrumental

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

Element bundle / files

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

Element files / file

Namespace	No namespace
-----------	--------------

Diagram	
Type	file
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	ALL(location type{0,1} filetype{0,1} samplerate{0,1} prelistening_offset{0,1} prelistening_length{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, prelistening_length, prelistening_offset, samplerate, samplesize, type
Instance	<pre> <file> <location>{1,1}</location> <type>{0,1}</type> <filetype>{0,1}</filetype> <samplerate>{0,1}</samplerate> <prelistening_offset>{0,1}</prelistening_offset> <prelistening_length>{0,1}</prelistening_length> <samplesize>{0,1}</samplesize> <bitrate>{0,1}</bitrate> </pre>

	<pre><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></pre>
Source	<code><xsd:element name="file" type="file" maxOccurs="unbounded" minOccurs="0" /></code>

Element file / location

Namespace	No namespace
Diagram	<p>This element contains the path to the corresponding file.</p>
Type	fileLocation
Properties	content: complex
Model	ALL(path{0,1} http{0,1} ftp{0,1})
Children	ftp, http, path
Instance	<pre><location> <path>{0,1}</path> <http>{0,1}</http> <ftp>{0,1}</ftp> </location></pre>
Source	<code><xsd:element name="location" type="fileLocation" /></code>

Element fileLocation / path

Namespace	No namespace						
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>						
Type	xsd:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<code><xsd:element name="path" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>						

Element fileLocation / http

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram fileHttp < -- action fileHttp < -- http fileHttp < -- url fileHttp < -- user fileHttp < -- pass fileHttp < -- expiresdatetime </pre> <p>This element contains information about http access to file.</p>						
Type	fileHttp						
Type hierarchy	<ul style="list-style-type: none"> • action • fileHttp 						
Properties	<table border="1"> <tr> <td>content:</td><td>complex</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> <tr> <td>maxOccurs:</td><td>1</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	ALL(url user{0,1} pass{0,1} expiresdatetime)						
Children	expiresdatetime, pass, url, user						
Instance	<pre> <http> <url>{1,1}</url> <user>{0,1}</user> <pass>{0,1}</pass> <expiresdatetime>{1,1}</expiresdatetime> </http> </pre>						
Source	<code><xsd:element name="http" type="fileHttp" maxOccurs="1" minOccurs="0"/></code>						

Element fileHttp / url

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

Element fileHttp / user

Namespace	No namespace						
Diagram	<pre> classDiagram user < -- xsd:string </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>						
Type	xsd:string						
Properties	<table border="1"> <tr> <td>content:</td><td>simple</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> <tr> <td>maxOccurs:</td><td>1</td></tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<code><xsd:element name="user" type="xsd:string" maxOccurs="1" minOccurs="0"/></code>						

Element fileHttp / pass

Namespace	No namespace
-----------	--------------

Diagram	A diagram showing the 'pass' element type. It is of type 'xsd:string'. A callout box states: 'Built-in primitive type. The string datatype represents character strings in XML.'
Type	xsd:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Source	<xsd:element name="pass" type="xsd:string" maxOccurs="1" minOccurs="0" />

Element fileHttp / expiresdatetime

Namespace	No namespace
Diagram	A diagram showing the 'expiresdatetime' element type. It is of type 'datetimeGMT'. A callout box states: 'This element contains information about ftp access to file.'
Type	datetimeGMT
Properties	content: simple
Facets	pattern $\d{4}-\d{2}-\d{2}$ $\d{2}:\d{2}:\d{2}$ GMT $:\d{2}$
Source	<xsd:element name="expiresdatetime" type="datetimeGMT" />

Element fileLocation / ftp

Namespace	No namespace
Diagram	A diagram showing the 'ftp' element type. It is of type 'fileFtp'. A callout box states: 'This element contains information about ftp access to file.' Below the diagram is a detailed view of the 'fileFtp' type, which includes 'server', 'port', 'path', 'user', 'pass', and 'expiresdatetime' elements.
Type	fileFtp
Type hierarchy	<ul style="list-style-type: none"> • action • fileFtp
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	ALL(server port path user{0,1} pass{0,1} expiresdatetime)
Children	expiresdatetime, pass, path, port, server, user
Instance	<pre> <ftp> <server>{1,1}</server> <port>{1,1}</port> <path>{1,1}</path> </pre>

	<pre><user>{0,1}</user> <pass>{0,1}</pass> <expiresdatetime>{1,1}</expiresdatetime> </ftp></pre>
Source	<pre><xsd:element name="ftp" type="fileFtp" maxOccurs="1" minOccurs="0" /></pre>

Element fileFtp / server

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled 'server'. A line connects it to a purple rounded rectangle labeled 'xsd:string'. Below the 'xsd:string' box is a callout bubble containing the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<pre><xsd:element name="server" type="xsd:string" /></pre>

Element fileFtp / port

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled 'port'. A line connects it to a purple rounded rectangle labeled 'xsd:string'. Below the 'xsd:string' box is a callout bubble containing the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<pre><xsd:element name="port" type="xsd:string" /></pre>

Element fileFtp / path

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled 'path'. A line connects it to a purple rounded rectangle labeled 'xsd:string'. Below the 'xsd:string' box is a callout bubble containing the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<pre><xsd:element name="path" type="xsd:string" /></pre>

Element fileFtp / user

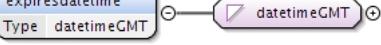
Namespace	No namespace						
Diagram	<p>The diagram shows a blue rounded rectangle labeled 'user'. A line connects it to a purple rounded rectangle labeled 'xsd:string'. Below the 'xsd:string' box is a callout bubble containing the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>						
Type	xsd:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<pre><xsd:element name="user" type="xsd:string" maxOccurs="1" minOccurs="0" /></pre>						

Element fileFtp / pass

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled 'pass'. A line connects it to a purple rounded rectangle labeled 'xsd:string'. Below the 'xsd:string' box is a callout bubble containing the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>

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

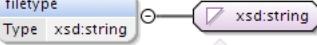
Element fileFtp / expiresdatetime

Namespace	No namespace
Diagram	
Type	datetimeGMT
Properties	content: simple
Facets	pattern $\backslash d\{4\}-\backslash d\{2\}-\backslash d\{2\}$ $\backslash d\{2\}:\backslash d\{2\}:\backslash d\{2\}$ GMT\ $+\backslash d\{2\}:\backslash d\{2\}$
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 frontcover enumeration backcover enumeration booklet
Source	<xsd:element name="type" type="fileType" maxOccurs="1" minOccurs="0" />

Element file / filetype

Namespace	No namespace
Diagram	 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> Built-in primitive type. The string datatype represents character strings in XML. </div>
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	 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> Built-in primitive type. The string datatype represents character strings in XML. </div>

Type	xsd:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<xsd:element name="samplerate" type="xsd:string" maxOccurs="1" minOccurs="0"/>						

Element file / prelistening_offset

Namespace	No namespace				
Annotations	prelistening_offset is in seconds. prelistening_offset is a "SHOULD" for filetype "prelistening". prelistening_offset is a MUST NOT for any other filetype.				
Diagram	<pre> classDiagram class prelistening_offset { <<Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...>> } class xsd:int prelistening_offset < -- xsd:int </pre>				
Type	xsd:int				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xsd:element minOccurs="0" name="prelistening_offset" type="xsd:int"> <xsd:annotation> <xsd:documentation>prelistening_offset is in seconds. prelistening_offset is a "SHOULD" for filetype "prelistening". prelistening_offset is a MUST NOT for any other filetype.</xsd:documentation> </xsd:annotation> </xsd:element>				

Element file / prelistening_length

Namespace	No namespace				
Annotations	prelistening_length is in seconds. prelistening_offset is a "SHOULD" for filetype "prelistening". prelistening_length is a MUST NOT for any other filetype.				
Diagram	<pre> classDiagram class prelistening_length { <<Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...>> } class xsd:int prelistening_length < -- xsd:int </pre>				
Type	xsd:int				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xsd:element minOccurs="0" name="prelistening_length" type="xsd:int"> <xsd:annotation> <xsd:documentation>prelistening_length is in seconds. prelistening_offset is a "SHOULD" for filetype "prelistening". prelistening_length is a MUST NOT for any other filetype.</xsd:documentation> </xsd:annotation> </xsd:element>				

Element file / samplesize

Namespace	No namespace		
Diagram	<pre> classDiagram class samplesize { <<Built-in primitive type. The string datatype represents character strings in XML.>> } class xsd:string samplesize < -- xsd:string </pre>		
Type	xsd:string		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
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	<pre><xsd:element name="bytes" type="xsd:integer" maxOccurs="1" minOccurs="0"/></pre>

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	<pre><checksums> <md5>{0,1}</md5> <sha1>{0,1}</sha1> <sha256>{0,1}</sha256> </checksums></pre>
Source	<pre><xsd:element name="checksums" type="checkums"/></pre>

Element checksums / md5

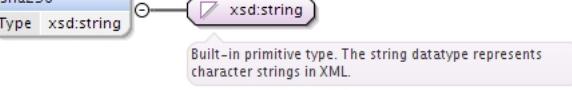
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<pre><xsd:element name="md5" type="xsd:string" maxOccurs="1" minOccurs="0"/></pre>

Element checksums / sha1

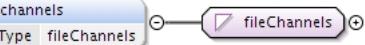
Namespace	No namespace
Diagram	

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

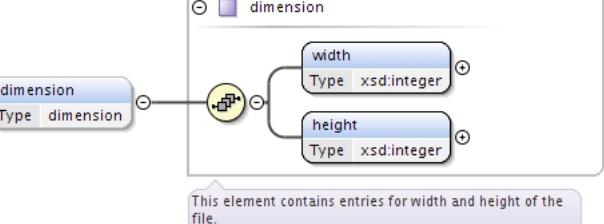
Element checksums / sha256

Namespace	No namespace
Diagram	 <p>The diagram shows a node labeled "sha256" with a "Type" of "xsd:string". A line connects "sha256" to "xsd:string". A callout box below "xsd:string" states: "Built-in primitive type. The string datatype represents character strings in XML."</p>
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	 <p>The diagram shows a node labeled "channels" with a "Type" of "fileChannels". A line connects "channels" to "fileChannels".</p>
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	 <p>The diagram shows a node labeled "dimension" with a "Type" of "dimension". A line connects "dimension" to "dimension". Inside the "dimension" node, there are two child nodes: "width" (Type: xsd:integer) and "height" (Type: xsd:integer). A callout box below the "dimension" node states: "This element contains entries for width and height of the file."</p>
Type	dimension
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	width , height
Children	height, width
Instance	<dimension> <width>{1,1}</width>

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

Element dimension / width

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

Element dimension / height

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

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	<pre><decryptinfo> <cipher>{0,1}</cipher> <initvector>{0,1}</initvector> <key>{0,1}</key> <bytes>{0,1}</bytes> <checksums>{0,1}</checksums> </decryptinfo></pre>
Source	<code><xsd:element name="decryptinfo" type="decryptinfo" maxOccurs="1" minOccurs="0" /></code>

Element decryptinfo / cipher

Namespace	No namespace
Diagram	<p>The diagram shows the element 'cipher' with its type 'xsd:string'. A callout box indicates that 'xsd:string' is a 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
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	<p>The diagram shows the element 'initvector' with its type 'xsd:string'. A callout box indicates that 'xsd:string' is a 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
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	<p>The diagram shows the element 'key' with its type 'xsd:string'. A callout box indicates that 'xsd:string' is a 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
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	<p>The diagram shows the element 'bytes' with its type 'xsd:string'. A callout box indicates that 'xsd:string' is a 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
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	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	ALL(md5{0,1} sha1{0,1} sha256{0,1})
Children	md5, sha1, sha256
Instance	<pre><checksums> <md5>{0,1}</md5> <sha1>{0,1}</sha1> <sha256>{0,1}</sha256> </checksums></pre>
Source	<code><xsd:element name="checksums" type="checksums" minOccurs="0" maxOccurs="1"/></code>

Element bundle / purchase

Namespace	No namespace
Diagram	
Type	purchase
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	ALL(pos url)
Children	pos, url
Instance	<pre><purchase> <pos>{1,1}</pos> <url>{1,1}</url> </purchase></pre>
Source	<code><xsd:element name="purchase" type="purchase" maxOccurs="1" minOccurs="0"/></code>

Element purchase / pos

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

Element purchase / url

Namespace	No namespace
Diagram	<p>The diagram shows the 'url' element with its type 'xsd:string'. A callout bubble indicates: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="url" type="xsd:string"/>

Element item / license_basis

Namespace	No namespace
Diagram	<p>The diagram shows the 'license_basis' element with its children: 'territorial', 'timeframe', 'pricing', 'streaming_allowed', 'channels', and 'as_on_bundle'. A callout bubble indicates: 'This element includes the basic rules and information under which this bundle is provided.'</p>
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	<pre><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></pre>
Source	<xsd:element name="license_basis" type="license_basis"/>

Element item / license_specifics

Namespace	No namespace
Diagram	<p>The diagram shows the 'license_specifics' element with its child 'rules'. A callout bubble indicates: 'This element includes specific rules which should be applied. This can be achieved by given rules.'</p>
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	<pre>tags +-- genres +-- bundle_only +-- explicit_lyrics +-- live +-- acoustic +-- instrumental</pre> <p>This element contains information about genres and more.</p>						
Type	tags						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	ALL(genres{0,1} bundle_only{0,1} explicit_lyrics{0,1} live{0,1} acoustic{0,1} instrumental{0,1})						
Children	acoustic, 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> <acoustic>{0,1}</acoustic> <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	<pre>fingerprint +-- echoprint</pre> <p>This element includes an element "echoprint" (http://echoprint.me https://github.com/echonest/echoprint-codegen).</p>						
Type	fingerprint						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	echoprint{0,1}						
Children	echoprint						
Instance	<pre><fingerprint> <echoprint>{0,1}</echoprint> </fingerprint></pre>						

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

Element **fingerprint** / **echoprint**

Namespace	No namespace						
Diagram	<p>The diagram shows a relationship between the <code>echoprint</code> element (xsd:string) and the <code>xsd:string</code> primitive type. A callout box indicates that <code>xsd:string</code> is a built-in primitive type representing character strings in XML.</p>						
Type	<code>xsd:string</code>						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	1
content:	simple						
minOccurs:	0						
maxOccurs:	1						
Source	<code><xsd:element name="echoprint" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>						

Element **item** / **reporting**

Namespace	No namespace						
Diagram	<p>The diagram shows the <code>reporting</code> element containing two children: <code>realtime</code> and <code>postponed</code>. A callout box indicates that this element contains information about reporting.</p>						
Type	<code>reporting</code>						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	ALL(realtime postponing)						
Children	postponed, realtime						
Instance	<code><reporting> <realtime>{1,1}</realtime> <postponed>{1,1}</postponed> </reporting></code>						
Source	<code><xsd:element name="reporting" type="reporting" maxOccurs="1" minOccurs="0" /></code>						

Element **item** / **files**

Namespace	No namespace						
Diagram	<p>The diagram shows the <code>files</code> element containing multiple <code>file</code> elements. A callout box indicates that this element contains a list of files.</p>						
Type	<code>files</code>						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	file*						
Children	file						
Instance	<code><files> <file>{0,unbounded}</file> </files></code>						
Source	<code><xsd:element name="files" type="files" maxOccurs="1" minOccurs="0" /></code>						

Element feed / item

Namespace	No namespace						
Diagram	<p>This element contains information about a item just like a track. The type describes what the item is e.g. audio,...</p>						
Type	item						
Properties	<table border="1"> <tr> <td>content:</td><td>complex</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> <tr> <td>maxOccurs:</td><td>unbounded</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
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	<pre> <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> </pre>						

Source

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

Complex Type(s)

Complex Type feedinfo

Namespace	No namespace
Annotations	<p>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.).</p>
Diagram	<pre> classDiagram class feedinfo { onlytest :xsd:boolean feedid :xsd:string creationdatetime :datetimeGMT effectivedatetime :datetimeGMT creator :creator receiver :receiver sender :sender licensor :licensor licensee :licensee actions :actions } </pre> <p>On feedinfo-level there are the global information needed or at least valuable for ingesting / identifying the content...</p>
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	<pre> <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,</pre>

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; whether 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	<pre> classDiagram class creator { email : email userid : userid keyid : xsd:string } </pre> <p>This element contains information about the creator of that feed.</p>
Used by	Element feedinfo/creator
Model	ALL(email userid{0,1} keyid{0,1})
Children	email, keyid, userid
Source	<pre> <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" maxOccurs="1" minOccurs="0"/> <xsd:element name="keyid" type="xsd:string" maxOccurs="1" minOccurs="0"/> </xsd:all> </xsd:complexType> </pre>

Complex Type receiver

Namespace	No namespace
Annotations	This element contains information about the receiver of that feed.
Diagram	<pre> classDiagram class receiver { type : receivertypes servername : iporhostname serveripv4 : ipv4 serveripv6 : ipv6 authtype : authtype username : xsd:string crypto : crypto } </pre> <p>This element contains information about the receiver of that feed.</p>

Used by	Element feedinfo/receiver
Model	ALL(type servername serveripv4 serveripv6 authtype username crypto)
Children	authtype, crypto, serveripv4, serveripv6, servername, type, username
Source	<pre><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></pre>

Complex Type crypto

Namespace	No namespace
Annotations	This element contains crypto information for secure and authenticated transfer.
Diagram	<pre> classDiagram class crypto { <<This element contains crypto information for secure and authenticated transfer.>> relatedemail "Type email" usedkeyid "Type keyid" usedpubkey "Type xsd:base64Binary" } </pre>
Used by	Element receiver/crypto
Model	ALL(relatedemail usedkeyid usedpubkey)
Children	relatedemail, usedkeyid, usedpubkey
Source	<pre><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></pre>

Complex Type sender

Namespace	No namespace
Annotations	This element contains information about the sender of that feed.
Diagram	<pre> classDiagram class sender { <<This element contains information about the sender of that feed.>> contractpartnerid "Type xsd:string" ourcontractpartnerid "Type xsd:string" email "Type email" keyid "Type xsd:string" } </pre>
Used by	Element feedinfo/sender
Model	ALL(contractpartnerid ourcontractpartnerid email{0,1} keyid{0,1})
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Source	<pre><xsd:complexType name="sender"> <xsd:annotation></pre>

```

<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" maxOccurs="1" minOccurs="0" />
  <xsd:element name="keyid" type="xsd:string" maxOccurs="1" minOccurs="0" />
</xsd:all>
</xsd:complexType>
```

Complex Type licensor

Namespace	No namespace
Annotations	This element contains information about the licensor of that feed.
Diagram	<pre> classDiagram class licensor { contractpartnerid : xsd:string ourcontractpartnerid : xsd:string email : email keyid : xsd:string } </pre> <p>This element contains information about the licensor of that feed.</p>
Used by	Element feedinfo/licensor
Model	ALL(contractpartnerid ourcontractpartnerid email{0,1} keyid{0,1})
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Source	<pre> <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" maxOccurs="1" minOccurs="0" /> <xsd:element name="keyid" type="xsd:string" maxOccurs="1" minOccurs="0" /> </xsd:all> </xsd:complexType></pre>

Complex Type licensee

Namespace	No namespace
Annotations	This element contains information about the licensee of that feed.
Diagram	<pre> classDiagram class licensee { contractpartnerid : xsd:string ourcontractpartnerid : xsd:string email : email keyid : xsd:string } </pre> <p>This element contains information about the licensee of that feed.</p>
Used by	Element feedinfo/licensee
Model	ALL(contractpartnerid ourcontractpartnerid email{0,1} keyid{0,1})
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Source	<pre> <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" maxOccurs="1" minOccurs="0" /> </xsd:all> </xsd:complexType></pre>

```
<xsd:element name="keyid" type="xsd:string" maxOccurs="1" minOccurs="0" />
</xsd:all>
</xsd:complexType>
```

Complex Type actions

Namespace	No namespace
Annotations	This element contains information about possible actions with the feed.
Diagram	<p>This element contains information about possible actions with the feed.</p>
Used by	Element feedinfo/actions
Model	ALL(oninitialreceive onprocessstart onprocessend onfullsuccess onerror)
Children	onerror, onfullsuccess, oninitialreceive, onprocessend, onprocessstart
Source	<pre><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="onprocessstart" type="event" /> <xsd:element name="onprocessend" type="event" /> <xsd:element name="onfullsuccess" type="event" /> <xsd:element name="onerror" type="event" /> </xsd:all> </xsd:complexType></pre>

Complex Type event

Namespace	No namespace
Annotations	This element contains information about possible events and actions.
Diagram	<p>This element contains information about possible events and actions.</p>
Used by	Elements actions/onerror, actions/onfullsuccess, actions/oninitialreceive, actions/onprocessend, actions/onprocessstart Complex Types onerror, onfullsuccess, oninitialreceive, onprocessend, onprocessstart
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Source	<pre><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></pre>

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

Complex Type mailto

Namespace	No namespace
Annotations	This element contains information about mailto-event.
Diagram	<pre> classDiagram class mailto { <<This element contains information about mailto-event.>> <<action (extension base)>> } class receiver { <<emailist>> <<1..oo>> } class subject { <<xsd:string>> } class text { <<xsd:string>> } mailto < -- action mailto --> receiver : "1..oo" mailto --> subject mailto --> text </pre>
Type	extension of action
Type hierarchy	<ul style="list-style-type: none"> • action • mailto
Used by	Element event/mailto
Model	receiver+, subject, text
Children	receiver, subject, text
Source	<pre> <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> </pre>

Complex Type action

Namespace	No namespace
Diagram	<pre> classDiagram class action { <<action (extension base)>> } class url { <<url>> <<Type url>> } class type { <<httpmethods>> <<Type httpmethods>> } class addheader { <<http_addheader>> <<Type http_addheader>> } class addparams { <<http_addparams>> <<Type http_addparams>> } action < -- Base Type action --> url action --> type action --> addheader action --> addparams </pre>
Used by	Complex Types fax, fileFtp, fileHttp, http, mailto
Source	<pre> <xsd:complexType name="action"> </xsd:complexType> </pre>

Complex Type http

Namespace	No namespace
Annotations	This element contains information about http-event.
Diagram	<pre> classDiagram class http { <<action (extension base)>> <<Base Type>> } class url { <<url>> <<Type url>> } class type { <<httpmethods>> <<Type httpmethods>> } class addheader { <<http_addheader>> <<Type http_addheader>> } class addparams { <<http_addparams>> <<Type http_addparams>> } http < -- action http --> url http --> type http --> addheader http --> addparams </pre>

Type	extension of action
Type hierarchy	<ul style="list-style-type: none"> • action <ul style="list-style-type: none"> • http
Used by	Elements event/http, realtime/http
Model	ALL(url type addheader addparams)
Children	addheader, addparams, type, url
Source	<pre><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></pre>

Complex Type http_addheader

Namespace	No namespace
Diagram	<pre> classDiagram class http_addheader { <<Base Type>> } class action_instruction { <<extension base>> } http_addheader < -- action_instruction </pre>
Type	extension of action_instruction
Type hierarchy	<ul style="list-style-type: none"> • action_instruction <ul style="list-style-type: none"> • http_addheader
Used by	Element http/addheader
Model	ANY element from ANY namespace
Source	<pre><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></pre>

Complex Type action_instruction

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

Complex Type http_addparams

Namespace	No namespace
Diagram	<pre> classDiagram class http_addparams { <<Base Type>> } class action_instruction { <<extension base>> } http_addparams < -- action_instruction </pre>
Type	extension of action_instruction
Type hierarchy	<ul style="list-style-type: none"> • action_instruction

	<ul style="list-style-type: none"> http_addparams
Used by	Element http/addparams
Model	ANY element from ANY namespace
Source	<pre><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></pre>

Complex Type fax

Namespace	No namespace
Diagram	<pre> classDiagram fax "Base Type" --o action fax "*" --o "#any" </pre>
Type	extension of action
Type hierarchy	<ul style="list-style-type: none"> action fax
Used by	Element event/fax
Model	ANY element from ANY namespace
Source	<pre><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></pre>

Complex Type letter

Namespace	No namespace
Annotations	This element contains information about the letter event.
Diagram	<pre> classDiagram letter "*" --o registered "Type xsd:boolean" letter "*" --o to "Type xsd:string" letter "*" --o text "Type xsd:string" letter "*" --o costscoveredby "Type xsd:string" note over letter: This element contains information about the letter event. </pre>
Used by	Element event/letter
Model	ALL(registered to text costscoveredby)
Children	costscoveredby, registered, text, to
Source	<pre><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="xsd:string" /> <xsd:element name="text" type="xsd:string" /> <xsd:element name="costscoveredby" type="xsd:string" /> </xsd:all> </xsd:complexType></pre>

Complex Type to

Namespace	No namespace
Annotations	This element contains information about recipient.
Diagram	<pre> classDiagram class to { name : xsd:string department : xsd:string nameperson : xsd:string street : xsd:string postcode : xsd:string country : xsd:string additionaladdressinfo : xsd:string } note over to: This element contains information about recipient. </pre>
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	<pre> <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> </pre>

Complex Type costscoveredby

Namespace	No namespace
Annotations	This element contains information about who covered the costs of event.
Diagram	<pre> classDiagram class costscoveredby { contractpartnerid : xsd:string ourcontractpartnerid : xsd:string maxcostscovered : xsd:string } note over costscoveredby: This element contains information about who covered the costs of event. </pre>
Used by	Element letter/costscoveredby
Model	ALL(contractpartnerid ourcontractpartnerid maxcostscovered{0,1})
Children	contractpartnerid, maxcostscovered, ourcontractpartnerid
Source	<pre> <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> </pre>

Complex Type bundle

Namespace	No namespace
Annotations	<p>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.</p>
Diagram	<pre> classDiagram class bundle { <<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.>> } displayname name version display_artistname ids items contributors information license_basis license_specifics reporting tags files purchase bundle <--> bundle </pre>
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	<pre> <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"/> </pre>

```

<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	<pre> <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> </pre>

Complex Type items

Namespace	No namespace
Annotations	This element is a container for item-elements.
Diagram	<p>This element is a container for item-elements.</p>
Used by	Element bundle/items
Model	item+
Children	item
Source	<pre><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></pre>

Complex Type item

Namespace	No namespace
Annotations	<p>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.</p>

Diagram	<pre> classDiagram class item { displayname name version type display_artistname ids contributors information license_basis license_specifics tags fingerprint reporting files } item "1..*" --> "1..*" item </pre>
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	<pre> <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> </pre>

Complex Type contributors

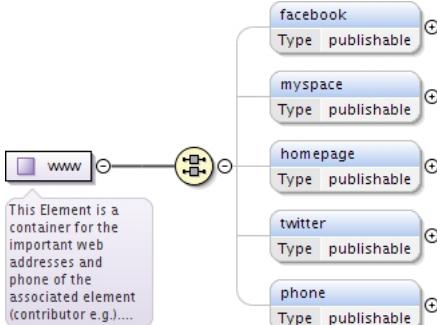
Namespace	No namespace
-----------	--------------

Annotations	This element contains a list of contributor.
Diagram	<pre> graph LR contributors["contributors"] --> contributor["contributor"] contributor --> contributorType["contributorType"] </pre>
Used by	Elements bundle/contributors, item/contributors
Model	contributor+
Children	contributor
Source	<pre> <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> </pre>

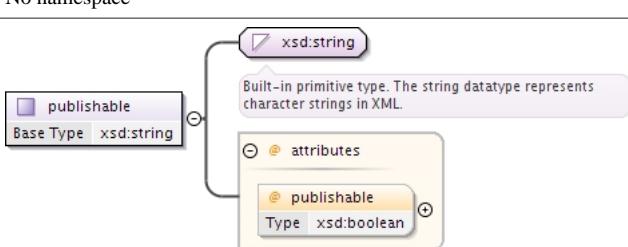
Complex Type contributor

Namespace	No namespace										
Annotations	<p>This element contains information of one contributor. A contributor can be a label, performer, texter, editor, conductor, artist, singer, composer, mixer, remixer, producer, featuring-Artist, with-Artist, DJ, versus-Artist, meets-Artist, presents-Artist, compilator, copyright, production or clearinghouse. A Year should be provided in case the type equals copyright or production.</p>										
Diagram	<pre> graph TD contributor["contributor"] --> attributes["attributes"] attributes --> num["num"] num --> type["type"] type --> name["name"] type --> year["year"] type --> ids["ids"] type --> www["www"] </pre>										
Used by	Element contributors/contributor										
Model	ALL(name type year{0,1} ids www{0,1})										
Children	ids, name, type, www, year										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>num</td> <td>xsd:integer</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	num	xsd:integer			optional
QName	Type	Fixed	Default	Use							
num	xsd:integer			optional							
Source	<pre> <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, editor, conductor, artist, singer, composer, mixer, remixer, producer, featuring-Artist, with-Artist, DJ, versus-Artist, meets-Artist, presents-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> </pre>										

Complex Type www

Namespace	No namespace
Annotations	<p>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.</p> <p>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.</p>
Diagram	 <pre> classDiagram www < -- facebook www < -- myspace www < -- homepage www < -- twitter www < -- phone www --> Note: This Element is a container for the important web addresses and phone of the associated element (contributor e.g.) facebook --> Note: Type publishable myspace --> Note: Type publishable homepage --> Note: Type publishable twitter --> Note: Type publishable phone --> Note: Type publishable </pre>
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	<pre> <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> </pre>

Complex Type publishable

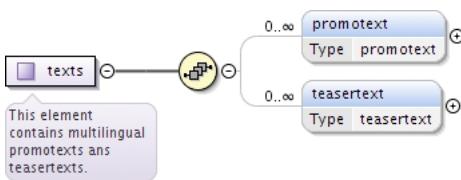
Namespace	No namespace										
Diagram	 <pre> classDiagram publishable < -- xsd:string xsd:string --> Note: Built-in primitive type. The string datatype represents character strings in XML. publishable --> Note: @ publishable publishable --> Note: Type xsd:boolean </pre>										
Type	extension of xsd:string										
Used by	Elements www/facebook, www/homepage, www/myspace, www/phone, www/twitter										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>publishable</td> <td>xsd:boolean</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	publishable	xsd:boolean			optional
QName	Type	Fixed	Default	Use							
publishable	xsd:boolean			optional							
Source	<pre> <xsd:complexType name="publishable"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="publishable" type="xsd:boolean"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType> </pre>										

Complex Type information

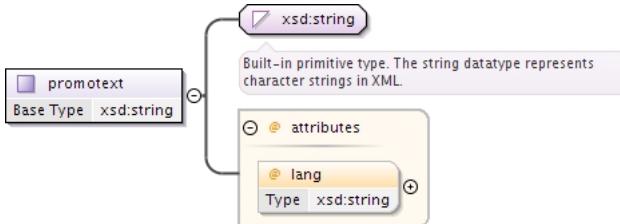
Namespace	No namespace
Annotations	<p>This element contains important data for an item/file. Multilingual promotexts ans teasertexts, dates of physical and digital release, playlength of file, position of the file in relation to other file of bundle, number of set (e.g. 2 for cd 2), the suggested prelistining offset if the file not starts e.g. with significant content, origin country and main language of file and information about related bundles.</p>
Diagram	<pre> classDiagram class information { <<This element contains important data for an item/file. Multilingual promotexts ans teasertexts, dates of physical and...>> } class texts class physical_release_datetime class digital_release_datetime class playlength class num class setnum class suggested_prelistening_offset class origin_country class main_language class related information "0..1" -- "*" texts information "0..1" -- "*" physical_release_datetime information "0..1" -- "*" digital_release_datetime information "0..1" -- "*" playlength information "0..1" -- "*" num information "0..1" -- "*" setnum information "0..1" -- "*" suggested_prelistening_offset information "0..1" -- "*" origin_country information "0..1" -- "*" main_language information "0..1" -- "*" related </pre>
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	<pre> <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="cc:countryCode" maxOccurs="1" minOccurs="0"/> <xsd:element name="main_language" type="l:language" maxOccurs="1" minOccurs="0"/> <xsd:element name="related" type="related" maxOccurs="1" minOccurs="0"/> </xsd:all> </xsd:complexType> </pre>

Complex Type texts

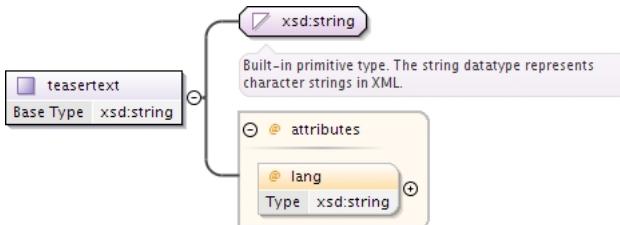
Namespace	No namespace
Annotations	This element contains multilingual promotexts ans teasertexts.

Diagram	
Used by	Element information/texts
Model	promotext*, teasertext*
Children	promotext, teasertext
Source	<pre><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></pre>

Complex Type promotext

Namespace	No namespace										
Diagram											
Type	extension of xsd:string										
Used by	Element texts/promotext										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>lang</td> <td>xsd:string</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	lang	xsd:string			optional
QName	Type	Fixed	Default	Use							
lang	xsd:string			optional							
Source	<pre><xsd:complexType name="promotext"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="lang" type="xsd:string"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType></pre>										

Complex Type teasertext

Namespace	No namespace										
Diagram											
Type	extension of xsd:string										
Used by	Element texts/teasertext										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>lang</td> <td>xsd:string</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	lang	xsd:string			optional
QName	Type	Fixed	Default	Use							
lang	xsd:string			optional							
Source	<pre><xsd:complexType name="teasertext"></pre>										

```
<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	<p>This element contains informations of bundles which are related to the bundle of the actual feed.</p>
Used by	Element information/related
Model	physical_distributor*, utube{0,1}, bundle*
Children	bundle, physical_distributor, utube
Source	<pre><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></pre>

Complex Type physical_distributor

Namespace	No namespace										
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>										
Type	extension of xsd:string										
Used by	Element related/physical_distributor										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>publishable</td> <td>xsd:boolean</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	publishable	xsd:boolean			optional
QName	Type	Fixed	Default	Use							
publishable	xsd:boolean			optional							
Source	<pre><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></pre>										

Complex Type utube

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

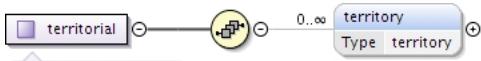
Diagram	
Used by	Element related/utube
Model	ALL(url{0,1} channel{0,1})
Children	channel, url
Source	<pre><xsd:complexType name="utube"> <xsd:annotation> <xsd:documentation xml:lang="en">Contains information about youtube url und channel.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="url"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

Complex Type license_basis

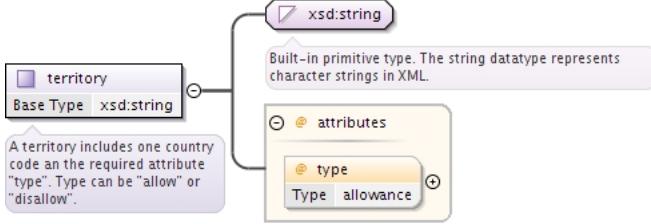
Namespace	No namespace
Annotations	This element includes the basic rules and information under which this bundle is provided.
Diagram	
Used by	Elements bundle/license_basis, item/license_basis
Model	ALL(territorial{0,1} timeframe{0,1} pricing{0,1} streaming_allowed{0,1} channels{0,1} as_on_bundle{0,1})
Children	as_on_bundle, channels, pricing, streaming_allowed, territorial, timeframe
Source	<pre><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></pre>

Complex Type territorial

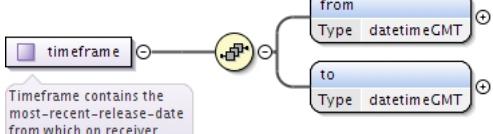
Namespace	No namespace
Annotations	This Element is a container for territories. There should be a entry for all territories with a attribute if distribution is allowed or not.

Diagram	
	This Element is a container for territories. There should be a entry for all territories with a attribute if...
Used by	Element license_basis/territorial
Model	territory*
Children	territory
Source	<pre><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></pre>

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	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Default</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>type</td> <td>allowance</td> <td></td> <td></td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Fixed	Default	Use	type	allowance			optional
QName	Type	Fixed	Default	Use							
type	allowance			optional							
Source	<pre><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></pre>										

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	<pre><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></pre>

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	<pre> classDiagram class pricing { <<Pricecode is an arbitrary-info. An explicitly given wholesale-price overrides the basic pricecode-given-wp. Most...>> +pricecode +wholesale } class pricecode { <<Type xsd:string>> } class wholesale { <<Type xsd:string>> } pricing < -- pricecode pricing < -- wholesale </pre>
Used by	Element license_basis/pricing
Model	ALL(pricecode{0,1} wholesale{0,1})
Children	pricecode, wholesale
Source	<pre><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></pre>

Complex Type channels

Namespace	No namespace
Annotations	This element is a container for channels which can be either "all", "ad supported" or "premium".
Diagram	<pre> classDiagram class channels { <<This element is a container for channels which can be either "all", "ad supported" or "premium".>> +channel } class channel { <<Type channel>> } channels < -- channel </pre>
Used by	Element license_basis/channels
Model	channel*
Children	channel
Source	<pre><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></pre>

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	<pre> classDiagram class channel { <<Base Type xsd:string>> } xsd:string < -- channel channel < -- @type : allowance note over xsd:string: Built-in primitive type. The string datatype represents character strings in XML. note over @type: @ type note over allowance: Type allowance </pre>										
Type	extension of xsd:string										
Used by	Element channels/channel										
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr> </thead> <tbody> <tr> <td>type</td><td>allowance</td><td></td><td></td><td>required</td></tr> </tbody> </table>	QName	Type	Fixed	Default	Use	type	allowance			required
QName	Type	Fixed	Default	Use							
type	allowance			required							
Source	<pre> <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> </pre>										

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	<pre> classDiagram class license_specifics class rules license_specifics --> rules note over license_specifics: This element includes specific rules which should be applied. This can be achieved by given rules. note over rules: rules </pre>
Used by	Elements bundle/license_specifics, item/license_specifics
Model	rules
Children	rules
Source	<pre> <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> </pre>

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	<pre> classDiagram class rules class rule rules --> rule note over rules: This element is a container for rules. It needs an ordered mode here - first come first match. note over rule: rule </pre>

Used by	Element	license_specifics/rules
Model	rule*	
Children	rule	
Source		<pre><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></pre>

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	<pre> classDiagram class rule { num : xsd:integer if then else } rule < -- rule </pre>				
Used by	Element				
Model	if , then , else{0,1}				
Children	else, if, then				
Attributes	QName	Type	Fixed	Default	Use
	num	xsd:integer			optional
Source	<pre><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></pre>				

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	<pre> classDiagram class if { what operator value } if < -- if </pre>				
	<p>This element must be the first element in a rule. It includes the information what is affected by the rule, an operator...</p>				

Used by	Element rule/if
Model	what , operator , value
Children	operator, value, what
Source	<pre><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></pre>

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	<pre><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></pre>

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	<pre><xsd:complexType name="else"></pre>

```

<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

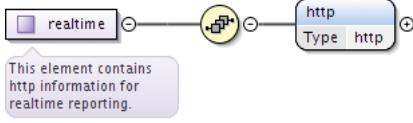
Namespace	No namespace
Annotations	This element includes the information what is affected and the corresponding value.
Diagram	<p>This element includes the information what is affected and the corresponding value.</p>
Used by	Element else/proclaim
Model	what , for
Children	for, what
Source	<pre> <xsd:complexType 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> </pre>

Complex Type reporting

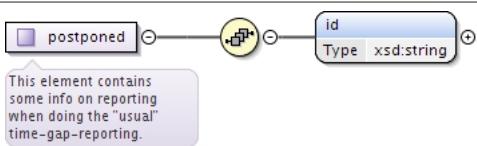
Namespace	No namespace
Annotations	This element contains information about reporting.
Diagram	<p>This element contains information about reporting.</p>
Used by	Elements bundle/reporting, item/reporting
Model	ALL(realtime postponed)
Children	postponed, realtime
Source	<pre> <xsd:complexType name="reporting"> <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> </pre>

Complex Type realtime

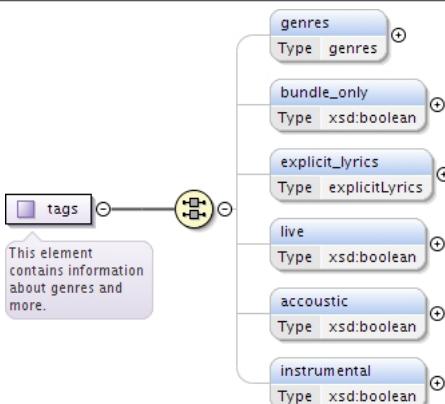
Namespace	No namespace
Annotations	This element contains http information for realtime reporting.

Diagram	 This element contains http information for realtime reporting.
Used by	Element reporting/realtime
Model	http
Children	http
Source	<pre><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></pre>

Complex Type postponed

Namespace	No namespace
Annotations	This element contains some info on reporting when doing the "usual" time-gap-reporting.
Diagram	 This element contains some info on reporting when doing the "usual" time-gap-reporting.
Used by	Element reporting/postponed
Model	id
Children	id
Source	<pre><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></pre>

Complex Type tags

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

Children	accoustic, bundle_only, explicit_lyrics, genres, instrumental, live
Source	<pre> <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></pre>

Complex Type genres

Namespace	No namespace
Annotations	This element contains a list of genres.
Diagram	<pre> graph LR genres[genres] -- "0..∞" --> genre[genre Type g:genre] subgraph direction TB subgraph direction LR genre end end </pre>
Used by	Element tags/genres
Model	genre*
Children	genre
Source	<pre> <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="g:genre" maxOccurs="unbounded" minOccurs="0"/> </xsd:sequence> </xsd:complexType></pre>

Complex Type files

Namespace	No namespace
Annotations	This element contains a list of files.
Diagram	<pre> graph LR files[files] -- "0..∞" --> file[file Type file] subgraph direction TB subgraph direction LR file end end </pre>
Used by	Elements bundle/files, item/files
Model	file*
Children	file
Source	<pre> <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></pre>

Complex Type file

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

Diagram	<pre> classDiagram class file { location <--> fileLocation type <--> fileType filetype <--> xsd:string samplerate <--> xsd:string prelistening_offset <--> xsd:int prelistening_length <--> xsd:int samplesize <--> xsd:string bitrate <--> xsd:string bitratetype <--> xsd:string codec <--> xsd:string codecsettings <--> xsd:string bytes <--> xsd:integer checksums <--> checksums channels <--> fileChannels dimension <--> dimension decryptinfo <--> decryptinfo } note over file: This element contains information and location of a file. </pre>
Used by	Element files/file
Model	ALL(location type{0,1} filetype{0,1} samplerate{0,1} prelistening_offset{0,1} prelistening_length{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, prelistening_length, prelistening_offset, samplerate, samplesize, type
Source	<pre> <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 minOccurs="0" name="prelistening_offset" type="xsd:int"> <xsd:annotation> <xsd:documentation>prelistening_offset is in seconds. prelistening_offset is a "SHOULD" for filetype "prelistening". prelistening_offset is a MUST NOT for any other filetype.</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element minOccurs="0" name="prelistening_length" type="xsd:int"> </pre>

```

<xsd:annotation>
  <xsd:documentation>prelistening_length is in seconds. prelistening_offset is a
  "SHOULD" for filetype "prelistening". prelistening_length is a MUST NOT for any other
  filetype.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<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	<pre> classDiagram class fileLocation { <<This element contains the path to the corresponding file.>> } class path { Type xsd:string } class http { Type fileHttp } class ftp { Type fileFtp } fileLocation --o path fileLocation --o http fileLocation --o ftp </pre>
Used by	Element file/location
Model	ALL(path{0,1} http{0,1} ftp{0,1})
Children	ftp, http, path
Source	<pre> <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> </pre>

Complex Type fileHttp

Namespace	No namespace
Annotations	This element contains information about http access to file.
Diagram	<pre> classDiagram class fileHttp { <<This element contains information about http access to file.>> <<action (extension base)>> } class action { <<action (extension base)>> } class url { Type url } class user { Type xsd:string } class pass { Type xsd:string } class expiresdatetime { Type datetimeGMT } fileHttp --o action action --o url action --o user action --o pass action --o expiresdatetime </pre>
Type	extension of action
Type hierarchy	<ul style="list-style-type: none"> • action • fileHttp
Used by	Element fileLocation/http

Model	ALL(url user{0,1} pass{0,1} expiresdatetime)
Children	expiresdatetime, pass, url, user
Source	<pre><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></pre>

Complex Type fileFtp

Namespace	No namespace
Annotations	This element contains information about ftp access to file.
Diagram	<pre> classDiagram class fileFtp { <<Base Type action>> <<This element contains information about ftp access to file.>> } class action { <<extension base>> } fileFtp < -- action fileFtp < -- server :xsd:string fileFtp < -- port :xsd:string fileFtp < -- path :xsd:string fileFtp < -- user :xsd:string fileFtp < -- pass :xsd:string fileFtp < -- expiresdatetime :xsd:string </pre>
Type	extension of action
Type hierarchy	<ul style="list-style-type: none"> • 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	<pre><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></pre>

Complex Type checksums

Namespace	No namespace
-----------	--------------

Annotations	This element contains checksums for the file.
Diagram	<pre> graph LR checksums[checksums] --> md5[md5] checksums --> sha1[sha1] checksums --> sha256[sha256] </pre> <p>This element contains checksums for the file.</p>
Used by	Elements decryptinfo/checksums, file/checksums
Model	ALL(md5{0,1} sha1{0,1} sha256{0,1})
Children	md5, sha1, sha256
Source	<pre> <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> </pre>

Complex Type dimension

Namespace	No namespace
Annotations	This element contains entries for width and height of the file.
Diagram	<pre> graph LR dimension[dimension] --> width[width] dimension --> height[height] </pre> <p>This element contains entries for width and height of the file.</p>
Used by	Element file/dimension
Model	width , height
Children	height, width
Source	<pre> <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> </pre>

Complex Type decryptinfo

Namespace	No namespace
Annotations	This element contains information about decryption of corresponding file.
Diagram	<pre> graph LR decryptinfo[decryptinfo] --> cipher[cipher] decryptinfo --> initvector[initvector] decryptinfo --> key[key] decryptinfo --> bytes[bytes] decryptinfo --> checksums[checksums] </pre> <p>This element contains information about decryption of corresponding file.</p>

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	<pre><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></pre>

Complex Type purchase

Namespace	No namespace
Annotations	This element contains information about purchase. Mostly when this feeds recipient is a POS.
Diagram	<pre> classDiagram purchase "1..2" -- "1..2" pos : Type xsd:string purchase "1..2" -- "1..2" url : Type xsd:string purchase <<This element contains information about purchase. Mostly when this feeds recipient is a POS. >> </pre>
Used by	Element bundle/purchase
Model	ALL(pos url)
Children	pos, url
Source	<pre><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></pre>

Complex Type fingerprint

Namespace	No namespace
Annotations	This element includes an element "echoprint" (http://echoprint.me https://github.com/echonest/echoprint-codegen).
Diagram	<pre> classDiagram fingerprint "1..2" -- "1..2" echoprint : Type xsd:string fingerprint <<This element includes an element "echoprint" (http://echoprint.me https://github.com/echonest/echoprint-codegen).>> </pre>
Used by	Element item/fingerprint
Model	echoprint{0,1}
Children	echoprint
Source	<pre><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></pre>

Complex Type oninitialreceive

Namespace	No namespace
Annotations	This element contains information about what should be done on initial receive of the feed.
Diagram	<pre> classDiagram event < -- oninitialreceive event < -- mailto event < -- http event < -- fax event < -- letter </pre> <p>This element contains information about what should be done on initial receive of the feed.</p> <p>This element contains information about possible events and actions.</p>
Type	extension of event
Type hierarchy	<ul style="list-style-type: none"> • event <ul style="list-style-type: none"> • oninitialreceive
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Source	<pre> <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> </pre>

Complex Type onprocessstart

Namespace	No namespace
Annotations	This element contains information about what should be done on the start of processing the feed.
Diagram	<pre> classDiagram event < -- onprocessstart event < -- mailto event < -- http event < -- fax event < -- letter </pre> <p>This element contains information about what should be done on the start of processing the feed.</p> <p>This element contains information about possible events and actions.</p>
Type	extension of event
Type hierarchy	<ul style="list-style-type: none"> • event <ul style="list-style-type: none"> • onprocessstart
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Source	<pre> <xsd:complexType name="onprocessstart"> <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:complexType> </pre>

```
<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	<p>This element contains information about what should be done on the end of processing the feed.</p> <p>This element contains information about possible events and actions.</p>
Type	extension of event
Type hierarchy	<ul style="list-style-type: none"> • event <ul style="list-style-type: none"> • onprocessend
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Source	<pre><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></pre>

Complex Type onfullsuccess

Namespace	No namespace
Annotations	This element contains information about what should be done on full success processing the feed.
Diagram	<p>This element contains information about what should be done on full success processing the feed.</p> <p>This element contains information about possible events and actions.</p>
Type	extension of event
Type hierarchy	<ul style="list-style-type: none"> • event <ul style="list-style-type: none"> • onfullsuccess
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto

Source	<pre><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></pre>
--------	--

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	<ul style="list-style-type: none"> • event • onerror
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Source	<pre><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></pre>

Simple Type(s)

Simple Type datetimetypeGMT

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	<p>pattern</p> $\begin{aligned} &\backslash d\{4\}-\backslash d\{2\}-\backslash d\{2\} \\ &\backslash d\{2\}:\backslash d\{2\}:\backslash d\{2\} \text{ GMT} \\ &+\backslash d\{2\}:\backslash d\{2\} \end{aligned}$
Used by	<p>Elements</p> <p>feedinfo/creationdatetime, feedinfo/effectivedatetime, fileFtp/expiredatetime, fileHttp/expiredatetime, information/digital_release_datetime, information/physical_release_datetime, timeframe/from, timeframe/to</p>
Source	<pre><xsd:simpleType name="datetimetypeGMT"> <xsd:restriction base="xsd:string"> <xsd:pattern value="\backslash d\{4\}-\backslash d\{2\}-\backslash d\{2\} \backslash d\{2\}:\backslash d\{2\}:\backslash d\{2\} \text{ GMT}\+\backslash d\{2\}:\backslash d\{2\}" /> <!-- "2010-01-31 00:00:00 GMT+00:00" - should be altered to some NMTOKENS or such ... --> </xsd:restriction></pre>

</xsd:simpleType>

Simple Type email

Namespace	No namespace
Diagram	<pre> graph LR email[email] --> xsdString[xsd:string] style email fill:#e0e0ff,stroke:#8080ff style xsdString fill:#e0e0ff,stroke:#8080ff </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Elements creator/email, crypto/relatedemail, licensee/email, licensor/email, sender/email
Source	<pre> <xsd:simpleType name="email"> <xsd:restriction base="xsd:string"/> </xsd:simpleType> </pre>

Simple Type userid

Namespace	No namespace
Diagram	<pre> graph LR userid[userid] --> xsdString[xsd:string] style userid fill:#e0e0ff,stroke:#8080ff style xsdString fill:#e0e0ff,stroke:#8080ff </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Element creator/userid
Source	<pre> <xsd:simpleType name="userid"> <xsd:restriction base="xsd:string"/> </xsd:simpleType> </pre>

Simple Type receivertypes

Namespace	No namespace										
Diagram	<pre> graph LR receivertypes[receivertypes] --> xsdString[xsd:string] style receivertypes fill:#e0e0ff,stroke:#8080ff style xsdString fill:#e0e0ff,stroke:#8080ff </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>										
Type	restriction of xsd:string										
Facets	<table border="1"> <tr> <td>enumeration</td> <td>ftp</td> </tr> <tr> <td>enumeration</td> <td>ftps</td> </tr> <tr> <td>enumeration</td> <td>sftp</td> </tr> <tr> <td>enumeration</td> <td>webdav</td> </tr> <tr> <td>enumeration</td> <td>openSDX fileserver</td> </tr> </table>	enumeration	ftp	enumeration	ftps	enumeration	sftp	enumeration	webdav	enumeration	openSDX fileserver
enumeration	ftp										
enumeration	ftps										
enumeration	sftp										
enumeration	webdav										
enumeration	openSDX fileserver										
Used by	Element receiver/type										
Source	<pre> <xsd:simpleType name="receivertypes"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="ftp"/> <xsd:enumeration value="ftps"/> <xsd:enumeration value="sftp"/> <xsd:enumeration value="webdav"/> <xsd:enumeration value="openSDX fileserver"/> </xsd:restriction> </xsd:simpleType> </pre>										

Simple Type iporhostname

Namespace	No namespace
Diagram	<pre> graph LR iporhostname[iporhostname] --> xsdString[xsd:string] style iporhostname fill:#e0e0ff,stroke:#8080ff style xsdString fill:#e0e0ff,stroke:#8080ff </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Element receiver/servername
Source	<pre> <xsd:simpleType name="iporhostname"> </pre>

```
<xsd:restriction base="xsd:string">
</xsd:restriction>
</xsd:simpleType>
```

Simple Type ipv4

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	restriction of xsd:string
Facets	<p>pattern</p> $((25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1-9)[0-9] ([0-9])(\.)\{3\})(25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1-9)[0-9] ([0-9])$
Used by	Element receiver/serverip4
Source	<pre><xsd:simpleType name="ipv4"> <xsd:restriction base="xsd:string"> <xsd:pattern value="((25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1-9)[0-9] ([0-9])(\.)\{3\})(25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1-9)[0-9] ([0-9])"/> <!-- should be checked for 0-255 etc. --&gt; &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;</pre> </pre>

Simple Type ipv6

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Element receiver/serverip6
Source	<pre><xsd:simpleType name="ipv6"> <xsd:restriction base="xsd:string"> <!-- not pattern defined yet... --&gt; &lt;/xsd:restriction&gt; &lt;/xsd:simpleType&gt;</pre> </pre>

Simple Type authtype

Namespace	No namespace										
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>										
Type	restriction of xsd:string										
Facets	<table border="1"> <tr> <td>enumeration</td> <td>login</td> </tr> <tr> <td>enumeration</td> <td>keyfile</td> </tr> <tr> <td>enumeration</td> <td>kerberos</td> </tr> <tr> <td>enumeration</td> <td>keyfile+login</td> </tr> <tr> <td>enumeration</td> <td>keyfile+username</td> </tr> </table>	enumeration	login	enumeration	keyfile	enumeration	kerberos	enumeration	keyfile+login	enumeration	keyfile+username
enumeration	login										
enumeration	keyfile										
enumeration	kerberos										
enumeration	keyfile+login										
enumeration	keyfile+username										
Used by	Element receiver/authtype										
Source	<pre><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></pre>										

</xsd:simpleType>

Simple Type keyid

Namespace	No namespace
Diagram	<pre> graph LR keyid[keyid] --> xsdstring[xsd:string] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Element crypto/usedkeyid
Source	<pre> <xsd:simpleType name="keyid"> <xsd:restriction base="xsd:string"> </xsd:restriction> </xsd:simpleType> </pre>

Simple Type emaillist

Namespace	No namespace
Diagram	<pre> graph LR emaillist[emaillist] --> xsdstring[xsd:string] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Element mailto/receiver
Source	<pre> <xsd:simpleType name="emaillist"> <xsd:restriction base="xsd:string"/> <!-- make to NMTOKENS or such... --> </xsd:simpleType> </pre>

Simple Type url

Namespace	No namespace
Diagram	<pre> graph LR url[url] --> xsdstring[xsd:string] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Elements fileHttp/url, http/url
Source	<pre> <xsd:simpleType name="url"> <xsd:restriction base="xsd:string"> <!-- not pattern defined yet... --> </xsd:restriction> </xsd:simpleType> </pre>

Simple Type httpmethods

Namespace	No namespace						
Diagram	<pre> graph LR httpmethods[httpmethods] --> xsdstring[xsd:string] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>						
Type	restriction of xsd:string						
Facets	<table border="1"> <tr> <td>enumeration</td> <td>GET</td> </tr> <tr> <td>enumeration</td> <td>POST</td> </tr> <tr> <td>enumeration</td> <td>HEAD</td> </tr> </table>	enumeration	GET	enumeration	POST	enumeration	HEAD
enumeration	GET						
enumeration	POST						
enumeration	HEAD						
Used by	Element http/type						
Source	<pre> <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> </pre>						

Simple Type contributorType

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	enumeration label enumeration performer enumeration texter enumeration editor enumeration conductor enumeration artist enumeration singer enumeration composer enumeration mixer enumeration remixer enumeration producer enumeration featuring enumeration with enumeration DJ enumeration versus enumeration meets enumeration presents enumeration compilator enumeration copyright enumeration production enumeration publisher enumeration clearinghouse
Used by	Element contributor/type
Source	<pre> <xsd:simpleType name="contributorType"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="label"/> <xsd:enumeration value="performer"/> <xsd:enumeration value="texter"/> <xsd:enumeration value="editor"/> <xsd:enumeration value="conductor"/> <xsd:enumeration value="artist"/> <xsd:enumeration value="singer"/> <xsd:enumeration value="composer"/> <xsd:enumeration value="mixer"/> <xsd:enumeration value="remixer"/> <xsd:enumeration value="producer"/> <xsd:enumeration value="featuring"/> <xsd:enumeration value="with"/> <xsd:enumeration value="DJ"/> <xsd:enumeration value="versus"/> <xsd:enumeration value="meets"/> <xsd:enumeration value="presents"/> <xsd:enumeration value="compilator"/> <xsd:enumeration value="copyright"/> <xsd:enumeration value="production"/> <xsd:enumeration value="publisher"/> <xsd:enumeration value="clearinghouse"/> </xsd:restriction> </xsd:simpleType></pre>

Simple Type allowance

Namespace	No namespace
-----------	--------------

Diagram	
	Built-in primitive type. The string datatype represents character strings in XML.
Type	restriction of xsd:string
Facets	enumeration allow enumeration disallow
Used by	Attributes channel/@type, territory/@type
Source	<pre><xsd:simpleType name="allowance"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="allow"/> <xsd:enumeration value="disallow"/> </xsd:restriction> </xsd:simpleType></pre>

Simple Type operator

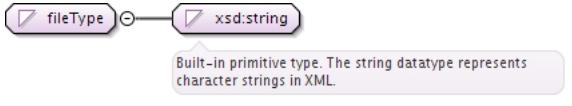
Namespace	No namespace
Diagram	
	Built-in primitive type. The string datatype represents character strings in XML.
Type	restriction of xsd:string
Facets	enumeration equals enumeration before enumeration after enumeration contains enumeration containedin
Used by	Element if/operator
Source	<pre><xsd:simpleType 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></pre>

Simple Type explicitLyrics

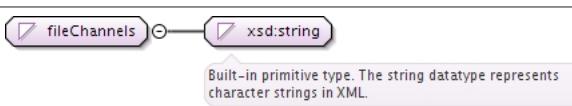
Namespace	No namespace
Diagram	
	Built-in primitive type. The string datatype represents character strings in XML.
Type	restriction of xsd:string
Facets	enumeration true enumeration false enumeration cleaned
Used by	Element tags/explicit_lyrics
Source	<pre><xsd:simpleType name="explicitLyrics"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="true"/> <xsd:enumeration value="false"/> <xsd:enumeration value="cleaned"/> </xsd:restriction> </xsd:simpleType></pre>

Simple Type fileType

Namespace	No namespace
-----------	--------------

Diagram											
Type	restriction of xsd:string										
Facets	<table> <tr><td>enumeration</td><td>full</td></tr> <tr><td>enumeration</td><td>prelistening</td></tr> <tr><td>enumeration</td><td>frontcover</td></tr> <tr><td>enumeration</td><td>backcover</td></tr> <tr><td>enumeration</td><td>booklet</td></tr> </table>	enumeration	full	enumeration	prelistening	enumeration	frontcover	enumeration	backcover	enumeration	booklet
enumeration	full										
enumeration	prelistening										
enumeration	frontcover										
enumeration	backcover										
enumeration	booklet										
Used by	Element file/type										
Source	<pre><xsd:simpleType name="fileType"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="full"/> <xsd:enumeration value="prelistening"/> <xsd:enumeration value="frontcover"/> <xsd:enumeration value="backcover"/> <xsd:enumeration value="booklet"/> </xsd:restriction> </xsd:simpleType></pre>										

Simple Type fileChannels

Namespace	No namespace								
Diagram									
Type	restriction of xsd:string								
Facets	<table> <tr><td>enumeration</td><td>mono</td></tr> <tr><td>enumeration</td><td>stereo</td></tr> <tr><td>enumeration</td><td>joint-stereo</td></tr> <tr><td>enumeration</td><td>5.1</td></tr> </table>	enumeration	mono	enumeration	stereo	enumeration	joint-stereo	enumeration	5.1
enumeration	mono								
enumeration	stereo								
enumeration	joint-stereo								
enumeration	5.1								
Used by	Element file/channels								
Source	<pre><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></pre>								

Attribute(s)

Attribute publishable / @publishable

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

Attribute contributor / @num

Namespace	No namespace
Type	xsd:integer
Properties	content: simple
Used by	Complex Type contributor

Source	<code><xsd:attribute name="num" type="xsd:integer" /></code>
--------	--

Attribute **promotext** / @lang

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

Attribute **teaserText** / @lang

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

Attribute **physical_distributor** / @publishable

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

Attribute **territory** / @type

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

Attribute **channel** / @type

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

Attribute **rule** / @num

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

Namespace: "http://fnppl.org/opensdx/countrycodes"

Schema(s)

Imported schema `openSDX_countryCodes.xsd`

Namespace	http://fnppl.org/opensdx/countrycodes
Properties	attribute form default: unqualified element form default: unqualified

Simple Type(s)

Simple Type `countryCode`

Namespace	http://fnppl.org/opensdx/countrycodes																																																																																																	
Annotations	This element includes a list of ISO 3166-1 country codes.																																																																																																	
Diagram	<pre> classDiagram class countryCode class xsd:string countryCode "1" -- "1" xsd:string </pre>	<p>This element includes a list of ISO 3166-1 country codes.</p> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>																																																																																																
Type	restriction of xsd:string																																																																																																	
Facets	<table> <tr><td>enumeration</td><td>AD</td><td>ANDORRA</td></tr> <tr><td>enumeration</td><td>AE</td><td>UNITED ARAB EMIRATES</td></tr> <tr><td>enumeration</td><td>AF</td><td>AFGHANISTAN</td></tr> <tr><td>enumeration</td><td>AG</td><td>ANTIGUA AND BARBUDA</td></tr> <tr><td>enumeration</td><td>AI</td><td>ANGUILLA</td></tr> <tr><td>enumeration</td><td>AL</td><td>ALBANIA</td></tr> <tr><td>enumeration</td><td>AM</td><td>ARMENIA</td></tr> <tr><td>enumeration</td><td>AN</td><td>NETHERLANDS ANTILLES</td></tr> <tr><td>enumeration</td><td>AO</td><td>ANGOLA</td></tr> <tr><td>enumeration</td><td>AQ</td><td>ANTARCTICA</td></tr> <tr><td>enumeration</td><td>AR</td><td>ARGENTINA</td></tr> <tr><td>enumeration</td><td>AS</td><td>AMERICAN SAMOA</td></tr> <tr><td>enumeration</td><td>AT</td><td>AUSTRIA</td></tr> <tr><td>enumeration</td><td>AU</td><td>AUSTRALIA</td></tr> <tr><td>enumeration</td><td>AW</td><td>ARUBA</td></tr> <tr><td>enumeration</td><td>AX</td><td>ÅLAND ISLANDS</td></tr> <tr><td>enumeration</td><td>AZ</td><td>AZERBAIJAN</td></tr> <tr><td>enumeration</td><td>BA</td><td>BOSNIA AND HERZEGOVINA</td></tr> <tr><td>enumeration</td><td>BB</td><td>BARBADOS</td></tr> <tr><td>enumeration</td><td>BD</td><td>BANGLADESH</td></tr> <tr><td>enumeration</td><td>BE</td><td>BELGIUM</td></tr> <tr><td>enumeration</td><td>BF</td><td>BURKINA FASO</td></tr> <tr><td>enumeration</td><td>BG</td><td>BULGARIA</td></tr> <tr><td>enumeration</td><td>BH</td><td>BAHRAIN</td></tr> <tr><td>enumeration</td><td>BI</td><td>BURUNDI</td></tr> <tr><td>enumeration</td><td>BJ</td><td>BENIN</td></tr> <tr><td>enumeration</td><td>BL</td><td>SAINT BARTHÉLEMY</td></tr> <tr><td>enumeration</td><td>BM</td><td>BERMUDA</td></tr> <tr><td>enumeration</td><td>BN</td><td>BRUNEI DARUSSALAM</td></tr> <tr><td>enumeration</td><td>BO</td><td>BOLIVIA</td></tr> <tr><td>enumeration</td><td>BR</td><td>BRAZIL</td></tr> <tr><td>enumeration</td><td>BS</td><td>BAHAMAS</td></tr> </table>		enumeration	AD	ANDORRA	enumeration	AE	UNITED ARAB EMIRATES	enumeration	AF	AFGHANISTAN	enumeration	AG	ANTIGUA AND BARBUDA	enumeration	AI	ANGUILLA	enumeration	AL	ALBANIA	enumeration	AM	ARMENIA	enumeration	AN	NETHERLANDS ANTILLES	enumeration	AO	ANGOLA	enumeration	AQ	ANTARCTICA	enumeration	AR	ARGENTINA	enumeration	AS	AMERICAN SAMOA	enumeration	AT	AUSTRIA	enumeration	AU	AUSTRALIA	enumeration	AW	ARUBA	enumeration	AX	ÅLAND ISLANDS	enumeration	AZ	AZERBAIJAN	enumeration	BA	BOSNIA AND HERZEGOVINA	enumeration	BB	BARBADOS	enumeration	BD	BANGLADESH	enumeration	BE	BELGIUM	enumeration	BF	BURKINA FASO	enumeration	BG	BULGARIA	enumeration	BH	BAHRAIN	enumeration	BI	BURUNDI	enumeration	BJ	BENIN	enumeration	BL	SAINT BARTHÉLEMY	enumeration	BM	BERMUDA	enumeration	BN	BRUNEI DARUSSALAM	enumeration	BO	BOLIVIA	enumeration	BR	BRAZIL	enumeration	BS	BAHAMAS
enumeration	AD	ANDORRA																																																																																																
enumeration	AE	UNITED ARAB EMIRATES																																																																																																
enumeration	AF	AFGHANISTAN																																																																																																
enumeration	AG	ANTIGUA AND BARBUDA																																																																																																
enumeration	AI	ANGUILLA																																																																																																
enumeration	AL	ALBANIA																																																																																																
enumeration	AM	ARMENIA																																																																																																
enumeration	AN	NETHERLANDS ANTILLES																																																																																																
enumeration	AO	ANGOLA																																																																																																
enumeration	AQ	ANTARCTICA																																																																																																
enumeration	AR	ARGENTINA																																																																																																
enumeration	AS	AMERICAN SAMOA																																																																																																
enumeration	AT	AUSTRIA																																																																																																
enumeration	AU	AUSTRALIA																																																																																																
enumeration	AW	ARUBA																																																																																																
enumeration	AX	ÅLAND ISLANDS																																																																																																
enumeration	AZ	AZERBAIJAN																																																																																																
enumeration	BA	BOSNIA AND HERZEGOVINA																																																																																																
enumeration	BB	BARBADOS																																																																																																
enumeration	BD	BANGLADESH																																																																																																
enumeration	BE	BELGIUM																																																																																																
enumeration	BF	BURKINA FASO																																																																																																
enumeration	BG	BULGARIA																																																																																																
enumeration	BH	BAHRAIN																																																																																																
enumeration	BI	BURUNDI																																																																																																
enumeration	BJ	BENIN																																																																																																
enumeration	BL	SAINT BARTHÉLEMY																																																																																																
enumeration	BM	BERMUDA																																																																																																
enumeration	BN	BRUNEI DARUSSALAM																																																																																																
enumeration	BO	BOLIVIA																																																																																																
enumeration	BR	BRAZIL																																																																																																
enumeration	BS	BAHAMAS																																																																																																

enumeration	BT	BHUTAN
enumeration	BV	BOUVET ISLAND
enumeration	BW	BOTSWANA
enumeration	BY	BELARUS
enumeration	BZ	BELIZE
enumeration	CA	CANADA
enumeration	CC	COCOS (KEELING) ISLANDS
enumeration	CD	CONGO, THE DEMOCRATIC REPUBLIC OF THE
enumeration	CF	CENTRAL AFRICAN REPUBLIC
enumeration	CG	CONGO
enumeration	CH	SWITZERLAND
enumeration	CI	COTE D'IVOIRE
enumeration	CK	COOK ISLANDS
enumeration	CL	CHILE
enumeration	CM	CAMEROON
enumeration	CN	CHINA
enumeration	CO	COLOMBIA
enumeration	CR	COSTA RICA
enumeration	CU	CUBA
enumeration	CV	CAPE VERDE
enumeration	CX	CHRISTMAS ISLAND
enumeration	CY	CYPRUS
enumeration	CZ	CZECH REPUBLIC
enumeration	DE	GERMANY
enumeration	DJ	DJIBOUTI
enumeration	DK	DENMARK
enumeration	DM	DOMINICA
enumeration	DO	DOMINICAN REPUBLIC
enumeration	DZ	ALGERIA
enumeration	EC	ECUADOR
enumeration	EE	ESTONIA
enumeration	EG	EGYPT
enumeration	EH	WESTERN SAHARA
enumeration	ER	ERITREA
enumeration	ES	SPAIN
enumeration	ET	ETHIOPIA
enumeration	FI	FINLAND
enumeration	FJ	FIJI
enumeration	FK	FALKLAND ISLANDS (MALVINAS)
enumeration	FM	MICRONESIA, FEDERATED STATES OF
enumeration	FO	FAROE ISLANDS
enumeration	FR	FRANCE
enumeration	GA	GABON
enumeration	GB	UNITED KINGDOM
enumeration	GD	GRENADA
enumeration	GE	GEORGIA
enumeration	GF	FRENCH GUIANA
enumeration	GG	GUERNSEY
enumeration	GH	GHANA

enumeration	GI	GIBRALTAR
enumeration	GL	GREENLAND
enumeration	GM	GAMBIA
enumeration	GN	GUINEA
enumeration	GP	GUADELOUPE
enumeration	GQ	EQUATORIAL GUINEA
enumeration	GR	GREECE
enumeration	GS	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS
enumeration	GT	GUATEMALA
enumeration	GU	GUAM
enumeration	GW	GUINEA-BISSAU
enumeration	GY	GUYANA
enumeration	HK	HONG KONG
enumeration	HM	HEARD ISLAND AND MCDONALD ISLANDS
enumeration	HN	HONDURAS
enumeration	HR	CROATIA
enumeration	HT	HAITI
enumeration	HU	HUNGARY
enumeration	ID	INDONESIA
enumeration	IE	IRELAND
enumeration	IL	ISRAEL
enumeration	IM	ISLE OF MAN
enumeration	IN	INDIA
enumeration	IO	BRITISH INDIAN OCEAN TERRITORY
enumeration	IQ	IRAQ
enumeration	IR	IRAN, ISLAMIC REPUBLIC OF
enumeration	IS	ICELAND
enumeration	IT	ITALY
enumeration	JE	JERSEY
enumeration	JM	JAMAICA
enumeration	JO	JORDAN
enumeration	JP	JAPAN
enumeration	KE	KENYA
enumeration	KG	KYRGYZSTAN
enumeration	KH	CAMBODIA
enumeration	KI	KIRIBATI
enumeration	KM	COMOROS
enumeration	KN	SAINT KITTS AND NEVIS
enumeration	KP	KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF
enumeration	KR	KOREA, REPUBLIC OF
enumeration	KW	KUWAIT
enumeration	KY	CAYMAN ISLANDS
enumeration	KZ	KAZAKHSTAN
enumeration	LA	LAO PEOPLE'S DEMOCRATIC REPUBLIC
enumeration	LB	LEBANON
enumeration	LC	SAINT LUCIA
enumeration	LI	LIECHTENSTEIN
enumeration	LK	SRI LANKA
enumeration	LR	LIBERIA

enumeration	LS	LESOTHO
enumeration	LT	LITHUANIA
enumeration	LU	LUXEMBOURG
enumeration	LV	LATVIA
enumeration	LY	LIBYAN ARAB JAMAHIRIYA
enumeration	MA	MOROCCO
enumeration	MC	MONACO
enumeration	MD	MOLDOVA, REPUBLIC OF
enumeration	ME	MONTENEGRO
enumeration	MF	SAINT MARTIN (FRENCH PART)
enumeration	MG	MADAGASCAR
enumeration	MH	MARSHALL ISLANDS
enumeration	MK	MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF
enumeration	ML	MALI
enumeration	MM	MYANMAR
enumeration	MN	MONGOLIA
enumeration	MO	MACAO
enumeration	MP	NORTHERN MARIANA ISLANDS
enumeration	MQ	MARTINIQUE
enumeration	MR	MAURITANIA
enumeration	MS	MONTSERRAT
enumeration	MT	MALTA
enumeration	MU	MAURITIUS
enumeration	MV	MALDIVES
enumeration	MW	MALAWI
enumeration	MX	MEXICO
enumeration	MY	MALAYSIA
enumeration	MZ	MOZAMBIQUE
enumeration	NA	NAMIBIA
enumeration	NC	NEW CALEDONIA
enumeration	NE	NIGER
enumeration	NF	NORFOLK ISLAND
enumeration	NG	NIGERIA
enumeration	NI	NICARAGUA
enumeration	NL	NETHERLANDS
enumeration	NO	NORWAY
enumeration	NP	NEPAL
enumeration	NR	NAURU
enumeration	NU	NIUE
enumeration	NZ	NEW ZEALAND
enumeration	OM	OMAN
enumeration	PA	PANAMA
enumeration	PE	PERU
enumeration	PF	FRENCH POLYNESIA
enumeration	PG	PAPUA NEW GUINEA
enumeration	PH	PHILIPPINES
enumeration	PK	PAKISTAN
enumeration	PL	POLAND
enumeration	PM	SAINT PIERRE AND MIQUELON

enumeration	PN	PITCAIRN
enumeration	PR	PUERTO RICO
enumeration	PS	PALESTINIAN TERRITORY, OCCUPIED
enumeration	PT	PORTUGAL
enumeration	PW	PALAU
enumeration	PY	PARAGUAY
enumeration	QA	QATAR
enumeration	RE	REUNION
enumeration	RO	ROMANIA
enumeration	RS	SERBIA
enumeration	RU	RUSSIAN FEDERATION
enumeration	RW	RWANDA
enumeration	SA	SAUDI ARABIA
enumeration	SB	SOLOMON ISLANDS
enumeration	SC	SEYCHELLES
enumeration	SD	SUDAN
enumeration	SE	SWEDEN
enumeration	SG	SINGAPORE
enumeration	SH	SAINT HELENA
enumeration	SI	SLOVENIA
enumeration	SJ	SVALBARD AND JAN MAYEN
enumeration	SK	SLOVAKIA
enumeration	SL	SIERRA LEONE
enumeration	SM	SAN MARINO
enumeration	SN	SENEGAL
enumeration	SO	SOMALIA
enumeration	SR	SURINAME
enumeration	ST	SAO TOME AND PRINCIPE
enumeration	SV	EL SALVADOR
enumeration	SY	SYRIAN ARAB REPUBLIC
enumeration	SZ	SWAZILAND
enumeration	TC	TURKS AND CAICOS ISLANDS
enumeration	TD	CHAD
enumeration	TF	FRENCH SOUTHERN TERRITORIES
enumeration	TG	TOGO
enumeration	TH	THAILAND
enumeration	TJ	TAJIKISTAN
enumeration	TK	TOKELAU
enumeration	TL	TIMOR-LESTE
enumeration	TM	TURKMENISTAN
enumeration	TN	TUNISIA
enumeration	TO	TONGA
enumeration	TR	TURKEY
enumeration	TT	TRINIDAD AND TOBAGO
enumeration	TV	TUVALU
enumeration	TW	TAIWAN, PROVINCE OF CHINA
enumeration	TZ	TANZANIA, UNITED REPUBLIC OF
enumeration	UA	UKRAINE
enumeration	UG	UGANDA

	enumeration	UM	UNITED STATES MINOR OUTLYING ISLANDS
	enumeration	US	UNITED STATES
	enumeration	UY	URUGUAY
	enumeration	UZ	UZBEKISTAN
	enumeration	VA	HOLY SEE (VATICAN CITY STATE)
	enumeration	VC	SAINT VINCENT AND THE GRENADINES
	enumeration	VE	VENEZUELA, PLURINATIONAL STATE OF
	enumeration	VG	VIRGIN ISLANDS, BRITISH
	enumeration	VI	VIRGIN ISLANDS, U.S.
	enumeration	VN	VIET NAM
	enumeration	VU	VANUATU
	enumeration	WF	WALLIS AND FUTUNA
	enumeration	WS	SAMOA
	enumeration	YE	YEMEN
	enumeration	YT	MAYOTTE
	enumeration	ZA	SOUTH AFRICA
	enumeration	ZM	ZAMBIA
	enumeration	ZW	ZIMBABWE
Used by	Element	information/origin_country	
Source	<pre> <xsd:simpleType name="countryCode"> <xsd:annotation> <xsd:documentation xml:lang="en">This element includes a list of ISO 3166-1 country codes.</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration value="AD"> <xsd:annotation> <xsd:documentation>ANDORRA</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AE"> <xsd:annotation> <xsd:documentation>UNITED ARAB EMIRATES</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AF"> <xsd:annotation> <xsd:documentation>AFGHANISTAN</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AG"> <xsd:annotation> <xsd:documentation>ANTIGUA AND BARBUDA</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AI"> <xsd:annotation> <xsd:documentation>ANGUILLA</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AL"> <xsd:annotation> <xsd:documentation>ALBANIA</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AM"> <xsd:annotation> <xsd:documentation>ARMENIA</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AN"> <xsd:annotation> <xsd:documentation>NETHERLANDS ANTILLES</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AO"> <xsd:annotation> <xsd:documentation>ANGOLA</xsd:documentation> </xsd:annotation> </xsd:enumeration> </xsd:restriction> </xsd:simpleType> </pre>		

```
</xsd:enumeration>
<xsd:enumeration value="AQ">
  <xsd:annotation>
    <xsd:documentation>ANTARCTICA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AR">
  <xsd:annotation>
    <xsd:documentation>ARGENTINA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AS">
  <xsd:annotation>
    <xsd:documentation>AMERICAN SAMOA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AT">
  <xsd:annotation>
    <xsd:documentation>AUSTRIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AU">
  <xsd:annotation>
    <xsd:documentation>AUSTRALIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AW">
  <xsd:annotation>
    <xsd:documentation>ARUBA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AX">
  <xsd:annotation>
    <xsd:documentation>ÅLAND ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AZ">
  <xsd:annotation>
    <xsd:documentation>AZERBAIJAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BA">
  <xsd:annotation>
    <xsd:documentation>BOSNIA AND HERZEGOVINA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BB">
  <xsd:annotation>
    <xsd:documentation>BARBADOS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BD">
  <xsd:annotation>
    <xsd:documentation>BANGLADESH</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BE">
  <xsd:annotation>
    <xsd:documentation>BELGIUM</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BF">
  <xsd:annotation>
    <xsd:documentation>BURKINA FASO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BG">
  <xsd:annotation>
    <xsd:documentation>BULGARIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BH">
  <xsd:annotation>
    <xsd:documentation>BAHRAIN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BI">
  <xsd:annotation>
    <xsd:documentation>BURUNDI</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BJ">
  <xsd:annotation>
```

```
<xsd:documentation>BENIN</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BL">
  <xsd:annotation>
    <xsd:documentation>SAINT BARTH#LEMY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BM">
  <xsd:annotation>
    <xsd:documentation>BERMUDA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BN">
  <xsd:annotation>
    <xsd:documentation>BRUNEI DARUSSALAM</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BO">
  <xsd:annotation>
    <xsd:documentation>BOLIVIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BR">
  <xsd:annotation>
    <xsd:documentation>BRAZIL</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BS">
  <xsd:annotation>
    <xsd:documentation>BAHAMAS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BT">
  <xsd:annotation>
    <xsd:documentation>BHUTAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BV">
  <xsd:annotation>
    <xsd:documentation>BOUVET ISLAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BW">
  <xsd:annotation>
    <xsd:documentation>BOTSWANA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BY">
  <xsd:annotation>
    <xsd:documentation>BELARUS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BZ">
  <xsd:annotation>
    <xsd:documentation>BELIZE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CA">
  <xsd:annotation>
    <xsd:documentation>CANADA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CC">
  <xsd:annotation>
    <xsd:documentation>COCOS (KEELING) ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CD">
  <xsd:annotation>
    <xsd:documentation>CONGO, THE DEMOCRATIC REPUBLIC OF THE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CF">
  <xsd:annotation>
    <xsd:documentation>CENTRAL AFRICAN REPUBLIC</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CG">
  <xsd:annotation>
    <xsd:documentation>CONGO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
```

```
<xsd:enumeration value="CH">
  <xsd:annotation>
    <xsd:documentation>SWITZERLAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CI">
  <xsd:annotation>
    <xsd:documentation>COTE D'IVOIRE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CK">
  <xsd:annotation>
    <xsd:documentation>COOK ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CL">
  <xsd:annotation>
    <xsd:documentation>CHILE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CM">
  <xsd:annotation>
    <xsd:documentation>CAMEROON</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CN">
  <xsd:annotation>
    <xsd:documentation>CHINA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CO">
  <xsd:annotation>
    <xsd:documentation>COLOMBIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CR">
  <xsd:annotation>
    <xsd:documentation>COSTA RICA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CU">
  <xsd:annotation>
    <xsd:documentation>CUBA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CV">
  <xsd:annotation>
    <xsd:documentation>CAPE VERDE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CX">
  <xsd:annotation>
    <xsd:documentation>CHRISTMAS ISLAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CY">
  <xsd:annotation>
    <xsd:documentation>CYPRUS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CZ">
  <xsd:annotation>
    <xsd:documentation>CZECH REPUBLIC</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DE">
  <xsd:annotation>
    <xsd:documentation>GERMANY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DJ">
  <xsd:annotation>
    <xsd:documentation>DJIBOUTI</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DK">
  <xsd:annotation>
    <xsd:documentation>DENMARK</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DM">
  <xsd:annotation>
    <xsd:documentation>DOMINICA</xsd:documentation>
```

```
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DO">
  <xsd:annotation>
    <xsd:documentation>DOMINICAN REPUBLIC</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DZ">
  <xsd:annotation>
    <xsd:documentation>ALGERIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="EC">
  <xsd:annotation>
    <xsd:documentation>ECUADOR</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="EE">
  <xsd:annotation>
    <xsd:documentation>ESTONIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="EG">
  <xsd:annotation>
    <xsd:documentation>EGYPT</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="EH">
  <xsd:annotation>
    <xsd:documentation>WESTERN SAHARA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ER">
  <xsd:annotation>
    <xsd:documentation>ERITREA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ES">
  <xsd:annotation>
    <xsd:documentation>SPAIN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ET">
  <xsd:annotation>
    <xsd:documentation>ETHIOPIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FI">
  <xsd:annotation>
    <xsd:documentation>FINLAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FJ">
  <xsd:annotation>
    <xsd:documentation>FIJI</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FK">
  <xsd:annotation>
    <xsd:documentation>FALKLAND ISLANDS (MALVINAS)</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FM">
  <xsd:annotation>
    <xsd:documentation>MICRONESIA, FEDERATED STATES OF</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FO">
  <xsd:annotation>
    <xsd:documentation>FAROE ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FR">
  <xsd:annotation>
    <xsd:documentation>FRANCE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GA">
  <xsd:annotation>
    <xsd:documentation>GABON</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GB">
```

```
<xsd:annotation>
  <xsd:documentation>UNITED KINGDOM</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GD">
  <xsd:annotation>
    <xsd:documentation>GRENADE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GE">
  <xsd:annotation>
    <xsd:documentation>GEORGIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GF">
  <xsd:annotation>
    <xsd:documentation>FRENCH GUIANA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GG">
  <xsd:annotation>
    <xsd:documentation>GUERNSEY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GH">
  <xsd:annotation>
    <xsd:documentation>GHANA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GI">
  <xsd:annotation>
    <xsd:documentation>GIBRALTAR</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GL">
  <xsd:annotation>
    <xsd:documentation>GREENLAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GM">
  <xsd:annotation>
    <xsd:documentation>GAMBIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GN">
  <xsd:annotation>
    <xsd:documentation>GUINEA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GP">
  <xsd:annotation>
    <xsd:documentation>GUADELOUPE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GQ">
  <xsd:annotation>
    <xsd:documentation>EQUATORIAL GUINEA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GR">
  <xsd:annotation>
    <xsd:documentation>GREECE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GS">
  <xsd:annotation>
    <xsd:documentation>SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GT">
  <xsd:annotation>
    <xsd:documentation>GUATEMALA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GU">
  <xsd:annotation>
    <xsd:documentation>GUAM</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GW">
  <xsd:annotation>
    <xsd:documentation>GUINEA-BISSAU</xsd:documentation>
```

```
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GY">
  <xsd:annotation>
    <xsd:documentation>GUYANA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HK">
  <xsd:annotation>
    <xsd:documentation>HONG KONG</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HM">
  <xsd:annotation>
    <xsd:documentation>HEARD ISLAND AND MCDONALD ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HN">
  <xsd:annotation>
    <xsd:documentation>HONDURAS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HR">
  <xsd:annotation>
    <xsd:documentation>CROATIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HT">
  <xsd:annotation>
    <xsd:documentation>HAITI</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HU">
  <xsd:annotation>
    <xsd:documentation>HUNGARY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ID">
  <xsd:annotation>
    <xsd:documentation>INDONESIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IE">
  <xsd:annotation>
    <xsd:documentation>IRELAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IL">
  <xsd:annotation>
    <xsd:documentation>ISRAEL</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IM">
  <xsd:annotation>
    <xsd:documentation>ISLE OF MAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IN">
  <xsd:annotation>
    <xsd:documentation>INDIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IO">
  <xsd:annotation>
    <xsd:documentation>BRITISH INDIAN OCEAN TERRITORY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IQ">
  <xsd:annotation>
    <xsd:documentation>IRAQ</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IR">
  <xsd:annotation>
    <xsd:documentation>IRAN, ISLAMIC REPUBLIC OF</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IS">
  <xsd:annotation>
    <xsd:documentation>ICELAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IT">
```

```
<xsd:annotation>
  <xsd:documentation>ITALY</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="JE">
  <xsd:annotation>
    <xsd:documentation>JERSEY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="JM">
  <xsd:annotation>
    <xsd:documentation>JAMAICA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="JO">
  <xsd:annotation>
    <xsd:documentation>JORDAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="JP">
  <xsd:annotation>
    <xsd:documentation>JAPAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KE">
  <xsd:annotation>
    <xsd:documentation>KENYA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KG">
  <xsd:annotation>
    <xsd:documentation>KYRGYZSTAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KH">
  <xsd:annotation>
    <xsd:documentation>CAMBODIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KI">
  <xsd:annotation>
    <xsd:documentation>KIRIBATI</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KM">
  <xsd:annotation>
    <xsd:documentation>COMOROS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KN">
  <xsd:annotation>
    <xsd:documentation>SAINT KITTS AND NEVIS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KP">
  <xsd:annotation>
    <xsd:documentation>KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KR">
  <xsd:annotation>
    <xsd:documentation>KOREA, REPUBLIC OF</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KW">
  <xsd:annotation>
    <xsd:documentation>KUWAIT</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KY">
  <xsd:annotation>
    <xsd:documentation>CAYMAN ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KZ">
  <xsd:annotation>
    <xsd:documentation>KAZAKHSTAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LA">
  <xsd:annotation>
    <xsd:documentation>LAO PEOPLE'S DEMOCRATIC REPUBLIC</xsd:documentation>
  </xsd:annotation>
```

```
</xsd:enumeration>
<xsd:enumeration value="LB">
  <xsd:annotation>
    <xsd:documentation>LEBANON</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LC">
  <xsd:annotation>
    <xsd:documentation>SAINT LUCIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LI">
  <xsd:annotation>
    <xsd:documentation>LIECHTENSTEIN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LK">
  <xsd:annotation>
    <xsd:documentation>SRI LANKA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LR">
  <xsd:annotation>
    <xsd:documentation>LIBERIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LS">
  <xsd:annotation>
    <xsd:documentation>LESOTHO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LT">
  <xsd:annotation>
    <xsd:documentation>LITHUANIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LU">
  <xsd:annotation>
    <xsd:documentation>LUXEMBOURG</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LV">
  <xsd:annotation>
    <xsd:documentation>LATVIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LY">
  <xsd:annotation>
    <xsd:documentation>LIBYAN ARAB JAMAHIRIYA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MA">
  <xsd:annotation>
    <xsd:documentation>MOROCCO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MC">
  <xsd:annotation>
    <xsd:documentation>MONACO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MD">
  <xsd:annotation>
    <xsd:documentation>MOLDOVA, REPUBLIC OF</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ME">
  <xsd:annotation>
    <xsd:documentation>MONTENEGRO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MF">
  <xsd:annotation>
    <xsd:documentation>SAINT MARTIN (FRENCH PART)</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MG">
  <xsd:annotation>
    <xsd:documentation>MADAGASCAR</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MH">
  <xsd:annotation>
```

```
<xsd:documentation>MARSHALL ISLANDS</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MK">
  <xsd:annotation>
    <xsd:documentation>MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ML">
  <xsd:annotation>
    <xsd:documentation>MALI</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MM">
  <xsd:annotation>
    <xsd:documentation>MYANMAR</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MN">
  <xsd:annotation>
    <xsd:documentation>MONGOLIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MO">
  <xsd:annotation>
    <xsd:documentation>MACAO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MP">
  <xsd:annotation>
    <xsd:documentation>NORTHERN MARIANA ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MQ">
  <xsd:annotation>
    <xsd:documentation>MARTINIQUE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MR">
  <xsd:annotation>
    <xsd:documentation>MAURITANIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MS">
  <xsd:annotation>
    <xsd:documentation>MONTSERRAT</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MT">
  <xsd:annotation>
    <xsd:documentation>MALTA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MU">
  <xsd:annotation>
    <xsd:documentation>MAURITIUS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MV">
  <xsd:annotation>
    <xsd:documentation>MALDIVES</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MW">
  <xsd:annotation>
    <xsd:documentation>MALAWI</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MX">
  <xsd:annotation>
    <xsd:documentation>MEXICO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MY">
  <xsd:annotation>
    <xsd:documentation>MALAYSIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MZ">
  <xsd:annotation>
    <xsd:documentation>MOZAMBIQUE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
```

```
<xsd:enumeration value="NA">
  <xsd:annotation>
    <xsd:documentation>NAMIBIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NC">
  <xsd:annotation>
    <xsd:documentation>NEW CALEDONIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NE">
  <xsd:annotation>
    <xsd:documentation>NIGER</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NF">
  <xsd:annotation>
    <xsd:documentation>NORFOLK ISLAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NG">
  <xsd:annotation>
    <xsd:documentation>NIGERIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NI">
  <xsd:annotation>
    <xsd:documentation>NICARAGUA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NL">
  <xsd:annotation>
    <xsd:documentation>NETHERLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NO">
  <xsd:annotation>
    <xsd:documentation>NORWAY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NP">
  <xsd:annotation>
    <xsd:documentation>NEPAL</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NR">
  <xsd:annotation>
    <xsd:documentation>NAURU</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NU">
  <xsd:annotation>
    <xsd:documentation>NIUE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NZ">
  <xsd:annotation>
    <xsd:documentation>NEW ZEALAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="OM">
  <xsd:annotation>
    <xsd:documentation>OMAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PA">
  <xsd:annotation>
    <xsd:documentation>PANAMA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PE">
  <xsd:annotation>
    <xsd:documentation>PERU</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PF">
  <xsd:annotation>
    <xsd:documentation>FRENCH POLYNESIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PG">
  <xsd:annotation>
    <xsd:documentation>PAPUA NEW GUINEA</xsd:documentation>
```

```
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PH">
  <xsd:annotation>
    <xsd:documentation>PHILIPPINES</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PK">
  <xsd:annotation>
    <xsd:documentation>PAKISTAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PL">
  <xsd:annotation>
    <xsd:documentation>POLAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PM">
  <xsd:annotation>
    <xsd:documentation>SAINT PIERRE AND MIQUELON</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PN">
  <xsd:annotation>
    <xsd:documentation>PITCAIRN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PR">
  <xsd:annotation>
    <xsd:documentation>PUERTO RICO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PS">
  <xsd:annotation>
    <xsd:documentation>PALESTINIAN TERRITORY, OCCUPIED</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PT">
  <xsd:annotation>
    <xsd:documentation>PORTUGAL</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PW">
  <xsd:annotation>
    <xsd:documentation>PALAU</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PY">
  <xsd:annotation>
    <xsd:documentation>PARAGUAY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="QA">
  <xsd:annotation>
    <xsd:documentation>QATAR</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="RE">
  <xsd:annotation>
    <xsd:documentation>REUNION</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="RO">
  <xsd:annotation>
    <xsd:documentation>ROMANIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="RS">
  <xsd:annotation>
    <xsd:documentation>SERBIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="RU">
  <xsd:annotation>
    <xsd:documentation>RUSSIAN FEDERATION</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="RW">
  <xsd:annotation>
    <xsd:documentation>RWANDA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SA">
```

```
<xsd:annotation>
  <xsd:documentation>SAUDI ARABIA</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SB">
  <xsd:annotation>
    <xsd:documentation>SOLOMON ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SC">
  <xsd:annotation>
    <xsd:documentation>SEYCHELLES</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SD">
  <xsd:annotation>
    <xsd:documentation>SUDAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SE">
  <xsd:annotation>
    <xsd:documentation>SWEDEN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SG">
  <xsd:annotation>
    <xsd:documentation>SINGAPORE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SH">
  <xsd:annotation>
    <xsd:documentation>SAINT HELENA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SI">
  <xsd:annotation>
    <xsd:documentation>SLOVENIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SJ">
  <xsd:annotation>
    <xsd:documentation>SVALBARD AND JAN MAYEN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SK">
  <xsd:annotation>
    <xsd:documentation>SLOVAKIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SL">
  <xsd:annotation>
    <xsd:documentation>SIERRA LEONE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SM">
  <xsd:annotation>
    <xsd:documentation>SAN MARINO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SN">
  <xsd:annotation>
    <xsd:documentation>SENEGAL</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SO">
  <xsd:annotation>
    <xsd:documentation>SOMALIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SR">
  <xsd:annotation>
    <xsd:documentation>SURINAME</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ST">
  <xsd:annotation>
    <xsd:documentation>SAO TOME AND PRINCIPE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SV">
  <xsd:annotation>
    <xsd:documentation>EL SALVADOR</xsd:documentation>
  </xsd:annotation>
```

```
</xsd:enumeration>
<xsd:enumeration value="SY">
  <xsd:annotation>
    <xsd:documentation>SYRIAN ARAB REPUBLIC</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SZ">
  <xsd:annotation>
    <xsd:documentation>SWAZILAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TC">
  <xsd:annotation>
    <xsd:documentation>TURKS AND CAICOS ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TD">
  <xsd:annotation>
    <xsd:documentation>CHAD</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TF">
  <xsd:annotation>
    <xsd:documentation>FRENCH SOUTHERN TERRITORIES</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TG">
  <xsd:annotation>
    <xsd:documentation>TOGO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TH">
  <xsd:annotation>
    <xsd:documentation>THAILAND</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TJ">
  <xsd:annotation>
    <xsd:documentation>TAJIKISTAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TK">
  <xsd:annotation>
    <xsd:documentation>TOKELAU</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TL">
  <xsd:annotation>
    <xsd:documentation>TIMOR-LESTE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TM">
  <xsd:annotation>
    <xsd:documentation>TURKMENISTAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TN">
  <xsd:annotation>
    <xsd:documentation>TUNISIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TO">
  <xsd:annotation>
    <xsd:documentation>TONGA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TR">
  <xsd:annotation>
    <xsd:documentation>TURKEY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TT">
  <xsd:annotation>
    <xsd:documentation>TRINIDAD AND TOBAGO</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TV">
  <xsd:annotation>
    <xsd:documentation>TUVALU</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TW">
  <xsd:annotation>
```

```
<xsd:documentation>TAIWAN, PROVINCE OF CHINA</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TZ">
  <xsd:annotation>
    <xsd:documentation>TANZANIA, UNITED REPUBLIC OF</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="UA">
  <xsd:annotation>
    <xsd:documentation>UKRAINE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="UG">
  <xsd:annotation>
    <xsd:documentation>UGANDA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="UM">
  <xsd:annotation>
    <xsd:documentation>UNITED STATES MINOR OUTLYING ISLANDS</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="US">
  <xsd:annotation>
    <xsd:documentation>UNITED STATES</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="UY">
  <xsd:annotation>
    <xsd:documentation>URUGUAY</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="UZ">
  <xsd:annotation>
    <xsd:documentation>UZBEKISTAN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="VA">
  <xsd:annotation>
    <xsd:documentation>HOLY SEE (VATICAN CITY STATE)</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="VC">
  <xsd:annotation>
    <xsd:documentation>SAINT VINCENT AND THE GRENADINES</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="VE">
  <xsd:annotation>
    <xsd:documentation>VENEZUELA, PLURINATIONAL STATE OF</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="VG">
  <xsd:annotation>
    <xsd:documentation>VIRGIN ISLANDS, BRITISH</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="VI">
  <xsd:annotation>
    <xsd:documentation>VIRGIN ISLANDS, U.S.</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="VN">
  <xsd:annotation>
    <xsd:documentation>VIET NAM</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="VU">
  <xsd:annotation>
    <xsd:documentation>VANUATU</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="WF">
  <xsd:annotation>
    <xsd:documentation>WALLIS AND FUTUNA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="WS">
  <xsd:annotation>
    <xsd:documentation>SAMOA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
```

```

<xsd:enumeration value="YE">
  <xsd:annotation>
    <xsd:documentation>YEMEN</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="YT">
  <xsd:annotation>
    <xsd:documentation>MAYOTTE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ZA">
  <xsd:annotation>
    <xsd:documentation>SOUTH AFRICA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ZM">
  <xsd:annotation>
    <xsd:documentation>ZAMBIA</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ZW">
  <xsd:annotation>
    <xsd:documentation>ZIMBABWE</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>

```

Namespace: "http://fnppl.org/opensdx/genres"

Schema(s)

Imported schema openSDX_genres.xsd

Namespace	http://fnppl.org/opensdx/genres
Properties	attribute form default: unqualified element form default: unqualified

Simple Type(s)

Simple Type genre

Namespace	http://fnppl.org/opensdx/genres																														
Annotations	This element includes a list of ISO 3166-1 country codes.																														
Diagram	<p>This element includes a list of ISO 3166-1 country codes.</p> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>																														
Type	restriction of xsd:string																														
Facets	<table border="1"> <tr><td>enumeration</td><td>Rock</td></tr> <tr><td>enumeration</td><td>Beat</td></tr> <tr><td>enumeration</td><td>Blues Rock</td></tr> <tr><td>enumeration</td><td>Rock'n'Roll</td></tr> <tr><td>enumeration</td><td>Art Rock</td></tr> <tr><td>enumeration</td><td>Classic Rock</td></tr> <tr><td>enumeration</td><td>Experimental Rock</td></tr> <tr><td>enumeration</td><td>Glam Rock</td></tr> <tr><td>enumeration</td><td>Hard Rock</td></tr> <tr><td>enumeration</td><td>Krautrock</td></tr> <tr><td>enumeration</td><td>Progressive</td></tr> <tr><td>enumeration</td><td>Psychedelic Rock</td></tr> <tr><td>enumeration</td><td>Psychobilly</td></tr> <tr><td>enumeration</td><td>Rockabilly</td></tr> <tr><td>enumeration</td><td>Soft Rock</td></tr> </table>	enumeration	Rock	enumeration	Beat	enumeration	Blues Rock	enumeration	Rock'n'Roll	enumeration	Art Rock	enumeration	Classic Rock	enumeration	Experimental Rock	enumeration	Glam Rock	enumeration	Hard Rock	enumeration	Krautrock	enumeration	Progressive	enumeration	Psychedelic Rock	enumeration	Psychobilly	enumeration	Rockabilly	enumeration	Soft Rock
enumeration	Rock																														
enumeration	Beat																														
enumeration	Blues Rock																														
enumeration	Rock'n'Roll																														
enumeration	Art Rock																														
enumeration	Classic Rock																														
enumeration	Experimental Rock																														
enumeration	Glam Rock																														
enumeration	Hard Rock																														
enumeration	Krautrock																														
enumeration	Progressive																														
enumeration	Psychedelic Rock																														
enumeration	Psychobilly																														
enumeration	Rockabilly																														
enumeration	Soft Rock																														

enumeration	Southern Rock
enumeration	Surf
enumeration	Alternative
enumeration	Crossover
enumeration	Dark Wave
enumeration	Garage Rock
enumeration	Goth / Industrial
enumeration	Grunge
enumeration	Hardcore
enumeration	Indie Rock
enumeration	New Wave
enumeration	Punk
enumeration	Black Metal
enumeration	Death Metal
enumeration	Heavy Metal
enumeration	Power Metal
enumeration	Thrash / Speed Metal
enumeration	Doom Metal
enumeration	Grind Core
enumeration	Pop
enumeration	Dance Pop
enumeration	Deutschpop
enumeration	Disco
enumeration	Electropop
enumeration	Euro Dance
enumeration	Euro Pop
enumeration	French Pop
enumeration	Indie Pop
enumeration	Italo Pop
enumeration	J-Pop
enumeration	K-Pop
enumeration	Neue Deutsche Welle
enumeration	Pop Rock
enumeration	Power Pop
enumeration	Schlager
enumeration	Singer/Songwriter
enumeration	Synthpop
enumeration	Teen Pop
enumeration	Country
enumeration	Alternative Country
enumeration	Bluegrass
enumeration	Contemporary Folk
enumeration	Country Gospel
enumeration	Honky-Tonk
enumeration	Jewish / Yiddish Music
enumeration	Nashville Sound
enumeration	Outlaw / Progressive Country
enumeration	Texas Country

enumeration	Traditional Country
enumeration	Western Swing
enumeration	Folk
enumeration	Americana
enumeration	Folk Rock
enumeration	Irish Folk
enumeration	Jazz
enumeration	Acid Jazz
enumeration	Avantgarde
enumeration	Bebop
enumeration	Big Band
enumeration	Cajun / Zydeco
enumeration	Classic
enumeration	Cool Jazz
enumeration	Free jazz
enumeration	Hard Bop
enumeration	Jazz Fusion
enumeration	New Orleans Jazz
enumeration	Nu-Jazz
enumeration	Smooth Jazz
enumeration	Swing
enumeration	Vocal Jazz
enumeration	HipHop
enumeration	Alternative Hip Hop
enumeration	Crunk
enumeration	G-Funk
enumeration	Gangsta Rap
enumeration	Golden Era
enumeration	Grime
enumeration	Hyphy
enumeration	Instrumental
enumeration	New School
enumeration	Old School
enumeration	Turntablism
enumeration	US Eastcoast
enumeration	US Midwest
enumeration	US Southern
enumeration	US Westcoast
enumeration	Blues
enumeration	Boogie-Woogie
enumeration	Electric Blues Guitar
enumeration	Modern Blues
enumeration	Regional Blues
enumeration	Traditional Blues
enumeration	Soul
enumeration	Motown Sound
enumeration	Neo Soul
enumeration	Philly Sound
enumeration	Funk

enumeration	R&B
enumeration	Contemporary R&B
enumeration	Electronic
enumeration	Ambient
enumeration	Chill Out
enumeration	Lounge
enumeration	Downbeat
enumeration	Electronica
enumeration	Indie Disco
enumeration	Industrial / EBM
enumeration	Techno
enumeration	Dance
enumeration	Electro
enumeration	House
enumeration	Deep House
enumeration	Disco House
enumeration	Electro House
enumeration	Hard House
enumeration	Progressive House
enumeration	Soulful House
enumeration	Tech House
enumeration	Vocal House
enumeration	Big Beat
enumeration	Breakbeat
enumeration	Drum'n'Bass
enumeration	Dubstep
enumeration	Garage / UK Funky
enumeration	IDM
enumeration	Trip-Hop
enumeration	Trance
enumeration	Goa Trance
enumeration	Hard Trance
enumeration	Psychedelic Trance
enumeration	Gabba
enumeration	Jumpstyle / Hardstyle
enumeration	Classic
enumeration	12-Tone Composition
enumeration	Baroque
enumeration	Chamber Music
enumeration	Crossover / Popular Classicism
enumeration	Electronic Music / Computer Music
enumeration	Minimal Music
enumeration	Modern, 20th / 21st Century
enumeration	Music and other Media / Arts
enumeration	Music and Word
enumeration	Neo-Romanticism

enumeration	Neoclassicism
enumeration	New Music / Contemporary Music
enumeration	Postmodern Music
enumeration	Renaissance
enumeration	Romantic
enumeration	Serialism
enumeration	Waltz
enumeration	Ballet
enumeration	Brass Ensemble
enumeration	Concerto / Solo Instrument with Orchestra
enumeration	Mixed Ensemble (Strings / Wind)
enumeration	Mixed Wind Ensemble (Woodwind / Brass)
enumeration	Opera Arias
enumeration	Opera Baroque
enumeration	Opera Classical
enumeration	Opera Renaissance
enumeration	Opera Romantic
enumeration	Operettas
enumeration	Several Solo Instruments
enumeration	Solo Instrument
enumeration	String Ensemble
enumeration	String Orchestra
enumeration	String Quartet
enumeration	String Trio
enumeration	Symphonic Music / Orchestral Music
enumeration	Symphonies
enumeration	Vocal Ensemble
enumeration	Vocal Music
enumeration	Woodwind Ensemble
enumeration	Choir
enumeration	Boy's Choir
enumeration	Children's Choir
enumeration	Choir with Orchestra
enumeration	Female Choir
enumeration	Male Choir
enumeration	Mixed Choir
enumeration	Flute
enumeration	Recorder / English Flute
enumeration	Oboe
enumeration	Clarinet
enumeration	Bassoon
enumeration	Saxophone
enumeration	Horn
enumeration	Trumpet

enumeration	Trombone
enumeration	Tuba
enumeration	Violin
enumeration	Viola
enumeration	Violoncello
enumeration	Double Bass
enumeration	Guitar
enumeration	Mandolin
enumeration	Lute
enumeration	Harp
enumeration	Piano
enumeration	Harpsichord
enumeration	Accordion
enumeration	Organ
enumeration	Ancient Instruments
enumeration	Percussion (Vibraphone etc.)
enumeration	Miscellaneous Lead Instrument
enumeration	Reggae
enumeration	Contemporary
enumeration	Dancehall
enumeration	Dub
enumeration	Lover's Rock
enumeration	Reggaeton
enumeration	Roots
enumeration	Ska
enumeration	World
enumeration	African Music
enumeration	Afrobeat
enumeration	Asian and Middle Eastern Music
enumeration	Calypso
enumeration	Caribbean Music
enumeration	Celtic, European and Judaica Music
enumeration	Coupé Decalé
enumeration	Griot
enumeration	Highlife
enumeration	Kuduro
enumeration	Kwaito
enumeration	Makossa
enumeration	Marching Band
enumeration	Mento
enumeration	North American Music
enumeration	Parang
enumeration	Polka
enumeration	Rai
enumeration	Soca
enumeration	Soukous

enumeration	Zouk
enumeration	Latin
enumeration	Banda
enumeration	Bolero
enumeration	Bossa Nova
enumeration	Corridos
enumeration	Cumbia
enumeration	Fado
enumeration	Flamenco
enumeration	Grupero
enumeration	Mambo
enumeration	Mariachi
enumeration	Merengue
enumeration	Norteno
enumeration	Ranchero
enumeration	Rock En Espanol
enumeration	Salsa
enumeration	Samba
enumeration	Son Cubana
enumeration	Sonidero
enumeration	Tango
enumeration	Tejano
enumeration	Religious
enumeration	Christian Rock
enumeration	Christian Hip Hop
enumeration	Christian Pop
enumeration	Chants
enumeration	Gospel
enumeration	Hymn
enumeration	Mass
enumeration	Spiritual
enumeration	Worship
enumeration	Miscellaneous
enumeration	Anime / Video Game Soundtracks
enumeration	Bollywood
enumeration	Instrumental
enumeration	Karaoke
enumeration	Movie Scores
enumeration	Movie Soundtracks
enumeration	Sound Effects
enumeration	Soundtrack
enumeration	TV Soundtrack
enumeration	unclassifiable
enumeration	Wedding Music
enumeration	Holiday
enumeration	Word
enumeration	Business & Career
enumeration	Abstracts & Dossiers

enumeration	Accounting
enumeration	Business & Investing
enumeration	Communication
enumeration	Computers & Internet
enumeration	Economics
enumeration	Finance
enumeration	Management & Leadership
enumeration	Marketing & Sales
enumeration	Politics
enumeration	Self-Help
enumeration	Self-Organization
enumeration	Skills
enumeration	Small Business & Entrepeneurship
enumeration	Children's Audiobooks
enumeration	age
enumeration	Age
enumeration	Popular Characters
enumeration	Animal Stories
enumeration	Children's Book Classics
enumeration	Detective Stories
enumeration	Fairy Tales
enumeration	Fantasy & Spook
enumeration	Knowledge for Children
enumeration	Pirates, Knights & Historical
enumeration	Poems & Song
enumeration	Comedy & Humour
enumeration	Comedy & Cabaret
enumeration	Humoristic Novel
enumeration	Crime
enumeration	Detective Stories
enumeration	Detective Stories „Noir“
enumeration	Classic Detective Stories
enumeration	Scandinavian Detective Stories
enumeration	Temporary Detective Stories
enumeration	Education & Knowledge
enumeration	Art & Culture
enumeration	Biography & Memento
enumeration	Foreign Language
enumeration	History
enumeration	Philosophy
enumeration	Politics & Current Affairs
enumeration	Science & Technology
enumeration	Health, Mind & Body

enumeration	Autogenous Training
enumeration	Creativity
enumeration	Fitness
enumeration	Health
enumeration	Lifestyle
enumeration	Love & Erotic
enumeration	Meditation / Yoga
enumeration	Memory Training
enumeration	Mental Training
enumeration	Motivation
enumeration	Philosophy
enumeration	Positive Thinking & Attitude
enumeration	Psychology
enumeration	Spirituality & Religion
enumeration	Sports
enumeration	Wellness & Beauty
enumeration	Science Fiction & Fantasy
enumeration	Ancient World
enumeration	Fantasy-Romance
enumeration	Historical Thriller
enumeration	Horror Classics
enumeration	Medieval Times & Early Modern Era
enumeration	Thriller
enumeration	Mystery & Conspiracy
enumeration	Psychological Thriller
enumeration	Espionage, Politics & Justice
enumeration	Vatican & Secret Societies
enumeration	Science & Medicine
enumeration	Novels
enumeration	Erotica
enumeration	Contemporary German Literature
enumeration	Romance
enumeration	Entertainment
enumeration	Contemporary Literature
enumeration	Youth
enumeration	Detective Stories
enumeration	Fantasy
enumeration	For Girls
enumeration	Knowledge for Teenagers
enumeration	Mystery
enumeration	Pirates, Knights & Historical
enumeration	Youth Classics

enumeration	Youth Today
enumeration	Language
enumeration	Albanian
enumeration	Arabic
enumeration	Bengali
enumeration	Bosnian
enumeration	Bulgarian
enumeration	Cantonese/Yue
enumeration	Croatian
enumeration	Czech
enumeration	Danish
enumeration	Dutch
enumeration	English
enumeration	Finnish
enumeration	French
enumeration	German
enumeration	Greek
enumeration	Hebrew
enumeration	Hindi/Urdu
enumeration	Hungarian
enumeration	Italian
enumeration	Japanese
enumeration	Korean
enumeration	Macedonian
enumeration	Mandarin
enumeration	Norwegian
enumeration	Patois
enumeration	Portuguese
enumeration	Russian
enumeration	Serbian
enumeration	Spanish
enumeration	Swedish
enumeration	Tamil
enumeration	Turkish
enumeration	Vietnamese
enumeration	Film
enumeration	Action
enumeration	3D
enumeration	Adventure
enumeration	Animation
enumeration	Author's Film
enumeration	Biography
enumeration	Cartoon
enumeration	Children
enumeration	Comedy
enumeration	Crime & Gangster
enumeration	Disaster
enumeration	Documentary
enumeration	Drama

enumeration	Epic / Historical
enumeration	Erotic
enumeration	Expressionism
enumeration	Family
enumeration	Fantasy
enumeration	Film-Noir
enumeration	GLBT
enumeration	Horror
enumeration	Independent Film
enumeration	Martial-Arts / Eastern
enumeration	Monumental
enumeration	Musical / Dance
enumeration	Music
enumeration	Mystery
enumeration	Reality-TV
enumeration	Romantic
enumeration	Science Fiction
enumeration	Silent Movie
enumeration	Sport
enumeration	Thriller
enumeration	TV-Series
enumeration	Tragicomedy
enumeration	War / Anti-War
enumeration	Western
enumeration	Youth
enumeration	Country
enumeration	United Arab Emirates (AE)
enumeration	Afghanistan (AF)
enumeration	Antigua and Barbuda (AG)
enumeration	Anguilla (AI)
enumeration	Albania (AL)
enumeration	Armenia (AM)
enumeration	Angola (AO)
enumeration	Antarctica (AQ)
enumeration	Argentina (AR)
enumeration	American Samoa (AS)
enumeration	Austria (AT)
enumeration	Australia (AU)
enumeration	Aruba (AW)
enumeration	Åland Islands (AX)
enumeration	Azerbaijan (AZ)
enumeration	Bosnia and Herzegovina (BA)
enumeration	Barbados (BB)
enumeration	Bangladesh (BD)
enumeration	Belgium (BE)
enumeration	Burkina Faso (BF)
enumeration	Bulgaria (BG)

enumeration	Bahrain (BH)
enumeration	Burundi (BI)
enumeration	Benin (BJ)
enumeration	Saint Barthélemy (BL)
enumeration	Bermuda (BM)
enumeration	Brunei Darussalam (BN)
enumeration	Bolivia Plurinational State of (BO)
enumeration	Bonaire Saint Eustatius and Saba (BQ)
enumeration	Brazil (BR)
enumeration	Bahamas (BS)
enumeration	Bhutan (BT)
enumeration	Bouvet Island (BV)
enumeration	Botswana (BW)
enumeration	Belarus (BY)
enumeration	Belize (BZ)
enumeration	Canada (CA)
enumeration	Cocos (Keeling) Islands (CC)
enumeration	Congo the Democratic Republic of the (CD)
enumeration	Central African Republic (CF)
enumeration	Congo (CG)
enumeration	Switzerland (CH)
enumeration	Côte d'Ivoire (CI)
enumeration	Cook Islands (CK)
enumeration	Chile (CL)
enumeration	Cameroon (CM)
enumeration	China (CN)
enumeration	Colombia (CO)
enumeration	Costa Rica (CR)
enumeration	Cuba (CU)
enumeration	Cape Verde (CV)
enumeration	Curaçao (CW)
enumeration	Christmas Island (CX)
enumeration	Cyprus (CY)
enumeration	Czech Republic (CZ)
enumeration	Germany (DE)
enumeration	Djibouti (DJ)
enumeration	Denmark (DK)
enumeration	Dominica (DM)
enumeration	Dominican Republic (DO)
enumeration	Algeria (DZ)
enumeration	Ecuador (EC)
enumeration	Estonia (EE)
enumeration	Egypt (EG)
enumeration	Western Sahara (EH)

enumeration	Eritrea (ER)
enumeration	Spain (ES)
enumeration	Ethiopia (ET)
enumeration	Finland (FI)
enumeration	Fiji (FJ)
enumeration	Falkland Islands (Malvinas) (FK)
enumeration	Micronesia Federated States of (FM)
enumeration	Faroe Islands (FO)
enumeration	France (FR)
enumeration	Gabon (GA)
enumeration	United Kingdom (GB)
enumeration	Grenada (GD)
enumeration	Georgia (GE)
enumeration	French Guiana (GF)
enumeration	Guernsey (GG)
enumeration	Ghana (GH)
enumeration	Gibraltar (GI)
enumeration	Greenland (GL)
enumeration	Gambia (GM)
enumeration	Guinea (GN)
enumeration	Guadeloupe (GP)
enumeration	Equatorial Guinea (GQ)
enumeration	Greece (GR)
enumeration	South Georgia and the South Sandwich Islands (GS)
enumeration	Guatemala (GT)
enumeration	Guam (GU)
enumeration	Guinea-Bissau (GW)
enumeration	Guyana (GY)
enumeration	Hong Kong (HK)
enumeration	Heard Island and McDonald Islands (HM)
enumeration	Honduras (HN)
enumeration	Croatia (HR)
enumeration	Haiti (HT)
enumeration	Hungary (HU)
enumeration	Indonesia (ID)
enumeration	Ireland (IE)
enumeration	Israel (IL)
enumeration	Isle of Man (IM)
enumeration	India (IN)
enumeration	British Indian Ocean Territory (IO)
enumeration	Iraq (IQ)
enumeration	Iran Islamic Republic of (IR)
enumeration	Iceland (IS)
enumeration	Italy (IT)

enumeration	Jersey (JE)
enumeration	Jamaica (JM)
enumeration	Jordan (JO)
enumeration	Japan (JP)
enumeration	Kenya (KE)
enumeration	Kyrgyzstan (KG)
enumeration	Cambodia (KH)
enumeration	Kiribati (KI)
enumeration	Comoros (KM)
enumeration	Saint Kitts and Nevis (KN)
enumeration	Korea Democratic People's Republic of (KP)
enumeration	Korea Republic of (KR)
enumeration	Kuwait (KW)
enumeration	Cayman Islands (KY)
enumeration	Kazakhstan (KZ)
enumeration	Lao People's Democratic Republic (LA)
enumeration	Lebanon (LB)
enumeration	Saint Lucia (LC)
enumeration	Liechtenstein (LI)
enumeration	Sri Lanka (LK)
enumeration	Liberia (LR)
enumeration	Lesotho (LS)
enumeration	Lithuania (LT)
enumeration	Luxembourg (LU)
enumeration	Latvia (LV)
enumeration	Libyan Arab Jamahiriya (LY)
enumeration	Morocco (MA)
enumeration	Monaco (MC)
enumeration	Moldova Republic of (MD)
enumeration	Montenegro (ME)
enumeration	Saint Martin (French part) (MF)
enumeration	Madagascar (MG)
enumeration	Marshall Islands (MH)
enumeration	Macedonia the former Yugoslav Republic of (MK)
enumeration	Mali (ML)
enumeration	Myanmar (MM)
enumeration	Mongolia (MN)
enumeration	Macao (MO)
enumeration	Northern Mariana Islands (MP)
enumeration	Martinique (MQ)
enumeration	Mauritania (MR)
enumeration	Montserrat (MS)

enumeration	Malta (MT)
enumeration	Mauritius (MU)
enumeration	Maldives (MV)
enumeration	Malawi (MW)
enumeration	Mexico (MX)
enumeration	Malaysia (MY)
enumeration	Mozambique (MZ)
enumeration	Namibia (NA)
enumeration	New Caledonia (NC)
enumeration	Niger (NE)
enumeration	Norfolk Island (NF)
enumeration	Nigeria (NG)
enumeration	Nicaragua (NI)
enumeration	Netherlands (NL)
enumeration	Norway (NO)
enumeration	Nepal (NP)
enumeration	Nauru (NR)
enumeration	Niue (NU)
enumeration	New Zealand (NZ)
enumeration	Oman (OM)
enumeration	Panama (PA)
enumeration	Peru (PE)
enumeration	French Polynesia (PF)
enumeration	Papua New Guinea (PG)
enumeration	Philippines (PH)
enumeration	Pakistan (PK)
enumeration	Poland (PL)
enumeration	Saint Pierre and Miquelon (PM)
enumeration	Pitcairn (PN)
enumeration	Puerto Rico (PR)
enumeration	Palestinian Territory Occupied (PS)
enumeration	Portugal (PT)
enumeration	Palau (PW)
enumeration	Paraguay (PY)
enumeration	Qatar (QA)
enumeration	Réunion (RE)
enumeration	Romania (RO)
enumeration	Serbia (RS)
enumeration	Russian Federation (RU)
enumeration	Rwanda (RW)
enumeration	Saudi Arabia (SA)
enumeration	Solomon Islands (SB)
enumeration	Seychelles (SC)
enumeration	Sudan (SD)
enumeration	Sweden (SE)
enumeration	Singapore (SG)

enumeration	Saint Helena Ascension and Tristan da Cunha (SH)
enumeration	Slovenia (SI)
enumeration	Svalbard and Jan Mayen (SJ)
enumeration	Slovakia (SK)
enumeration	Sierra Leone (SL)
enumeration	San Marino (SM)
enumeration	Senegal (SN)
enumeration	Somalia (SO)
enumeration	Suriname (SR)
enumeration	Sao Tome and Principe (ST)
enumeration	El Salvador (SV)
enumeration	Sint Maarten (Dutch part) (SX)
enumeration	Syrian Arab Republic (SY)
enumeration	Swaziland (SZ)
enumeration	Turks and Caicos Islands (TC)
enumeration	Chad (TD)
enumeration	French Southern Territories (TF)
enumeration	Togo (TG)
enumeration	Thailand (TH)
enumeration	Tajikistan (TJ)
enumeration	Tokelau (TK)
enumeration	Timor-Leste (TL)
enumeration	Turkmenistan (TM)
enumeration	Tunisia (TN)
enumeration	Tonga (TO)
enumeration	Turkey (TR)
enumeration	Trinidad and Tobago (TT)
enumeration	Tuvalu (TV)
enumeration	Taiwan Province of China (TW)
enumeration	Tanzania United Republic of (TZ)
enumeration	Ukraine (UA)
enumeration	Uganda (UG)
enumeration	United States Minor Outlying Islands (UM)
enumeration	United States (US)
enumeration	Uruguay (UY)
enumeration	Uzbekistan (UZ)
enumeration	Holy See (Vatican City State) (VA)
enumeration	Saint Vincent and the Grenadines (VC)
enumeration	Venezuela Bolivarian Republic of (VE)

	enumeration	Virgin Islands British (VG)
	enumeration	Virgin Islands U.S. (VI)
	enumeration	Viet Nam (VN)
	enumeration	Vanuatu (VU)
	enumeration	Wallis and Futuna (WF)
	enumeration	Samoa (WS)
	enumeration	Yemen (YE)
	enumeration	Mayotte (YT)
	enumeration	South Africa (ZA)
	enumeration	Zambia (ZM)
	enumeration	Zimbabwe (ZW)
Used by	Element	genres/genre
Source	<pre> <xsd:simpleType name="genre"> <xsd:annotation> <xsd:documentation xml:lang="en">This element includes a list of ISO 3166-1 country codes.</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Rock"/> <xsd:enumeration value="Beat"/> <xsd:enumeration value="Blues Rock"/> <xsd:enumeration value="Rock'n'Roll"/> <xsd:enumeration value="Art Rock"/> <xsd:enumeration value="Classic Rock"/> <xsd:enumeration value="Experimental Rock"/> <xsd:enumeration value="Glam Rock"/> <xsd:enumeration value="Hard Rock"/> <xsd:enumeration value="Krautrock"/> <xsd:enumeration value="Progressive"/> <xsd:enumeration value="Psychedelic Rock"/> <xsd:enumeration value="Psychobilly"/> <xsd:enumeration value="Rockabilly"/> <xsd:enumeration value="Soft Rock"/> <xsd:enumeration value="Southern Rock"/> <xsd:enumeration value="Surf"/> <xsd:enumeration value="Alternative"/> <xsd:enumeration value="Crossover"/> <xsd:enumeration value="Dark Wave"/> <xsd:enumeration value="Garage Rock"/> <xsd:enumeration value="Goth / Industrial"/> <xsd:enumeration value="Grunge"/> <xsd:enumeration value="Hardcore"/> <xsd:enumeration value="Indie Rock"/> <xsd:enumeration value="New Wave"/> <xsd:enumeration value="Punk"/> <xsd:enumeration value="Black Metal"/> <xsd:enumeration value="Death Metal"/> <xsd:enumeration value="Heavy Metal"/> <xsd:enumeration value="Power Metal"/> <xsd:enumeration value="Thrash / Speed Metal"/> <xsd:enumeration value="Doom Metal"/> <xsd:enumeration value="Grind Core"/> <xsd:enumeration value="Pop"/> <xsd:enumeration value="Dance Pop"/> <xsd:enumeration value="Deutschpop"/> <xsd:enumeration value="Disco"/> <xsd:enumeration value="Electropop"/> <xsd:enumeration value="Euro Dance"/> <xsd:enumeration value="Euro Pop"/> <xsd:enumeration value="French Pop"/> <xsd:enumeration value="Indie Pop"/> <xsd:enumeration value="Italo Pop"/> <xsd:enumeration value="J-Pop"/> <xsd:enumeration value="K-Pop"/> <xsd:enumeration value="Neue Deutsche Welle"/> <xsd:enumeration value="Pop Rock"/> <xsd:enumeration value="Power Pop"/> <xsd:enumeration value="Schlager"/> <xsd:enumeration value="Singer/Songwriter"/> <xsd:enumeration value="Synthpop"/> <xsd:enumeration value="Teen Pop"/></pre>	

```
<xsd:enumeration value="Country"/>
<xsd:enumeration value="Alternative Country"/>
<xsd:enumeration value="Bluegrass"/>
<xsd:enumeration value="Contemporary Folk"/>
<xsd:enumeration value="Country Gospel"/>
<xsd:enumeration value="Honky-Tonk"/>
<xsd:enumeration value="Jewish / Yiddish Music"/>
<xsd:enumeration value="Nashville Sound"/>
<xsd:enumeration value="Outlaw / Progressive Country"/>
<xsd:enumeration value="Texas Country"/>
<xsd:enumeration value="Traditional Country"/>
<xsd:enumeration value="Western Swing"/>
<xsd:enumeration value="Folk"/>
<xsd:enumeration value="Americana"/>
<xsd:enumeration value="Folk Rock"/>
<xsd:enumeration value="Irish Folk"/>
<xsd:enumeration value="Jazz"/>
<xsd:enumeration value="Acid Jazz"/>
<xsd:enumeration value="Avantgarde"/>
<xsd:enumeration value="Bebop"/>
<xsd:enumeration value="Big Band"/>
<xsd:enumeration value="Cajun / Zydeco"/>
<xsd:enumeration value="Classic"/>
<xsd:enumeration value="Cool Jazz"/>
<xsd:enumeration value="Free jazz"/>
<xsd:enumeration value="Hard Bop"/>
<xsd:enumeration value="Jazz Fusion"/>
<xsd:enumeration value="New Orleans Jazz"/>
<xsd:enumeration value="Nu-Jazz"/>
<xsd:enumeration value="Smooth Jazz"/>
<xsd:enumeration value="Swing"/>
<xsd:enumeration value="Vocal Jazz"/>
<xsd:enumeration value="HipHop"/>
<xsd:enumeration value="Alternative Hip Hop"/>
<xsd:enumeration value="Crunk"/>
<xsd:enumeration value="G-Funk"/>
<xsd:enumeration value="Gangsta Rap"/>
<xsd:enumeration value="Golden Era"/>
<xsd:enumeration value="Grime"/>
<xsd:enumeration value="Hyphy"/>
<xsd:enumeration value="Instrumental"/>
<xsd:enumeration value="New School"/>
<xsd:enumeration value="Old School"/>
<xsd:enumeration value="Turntablism"/>
<xsd:enumeration value="US Eastcoast"/>
<xsd:enumeration value="US Midwest"/>
<xsd:enumeration value="US Southern"/>
<xsd:enumeration value="US Westcoast"/>
<xsd:enumeration value="Blues"/>
<xsd:enumeration value="Boogie-Woogie"/>
<xsd:enumeration value="Electric Blues Guitar"/>
<xsd:enumeration value="Modern Blues"/>
<xsd:enumeration value="Regional Blues"/>
<xsd:enumeration value="Traditional Blues"/>
<xsd:enumeration value="Soul"/>
<xsd:enumeration value="Motown Sound"/>
<xsd:enumeration value="Neo Soul"/>
<xsd:enumeration value="Philly Sound"/>
<xsd:enumeration value="Funk"/>
<xsd:enumeration value="R&B"/>
<xsd:enumeration value="Contemporary R&B"/>
<xsd:enumeration value="Electronic"/>
<xsd:enumeration value="Ambient"/>
<xsd:enumeration value="Chill Out"/>
<xsd:enumeration value="Lounge"/>
<xsd:enumeration value="Downbeat"/>
<xsd:enumeration value="Electronica"/>
<xsd:enumeration value="Indie Disco"/>
<xsd:enumeration value="Industrial / EBM"/>
<xsd:enumeration value="Techno"/>
<xsd:enumeration value="Dance"/>
<xsd:enumeration value="Electro"/>
<xsd:enumeration value="House"/>
<xsd:enumeration value="Deep House"/>
<xsd:enumeration value="Disco House"/>
<xsd:enumeration value="Electro House"/>
<xsd:enumeration value="Hard House"/>
<xsd:enumeration value="Progressive House"/>
<xsd:enumeration value="Soulful House"/>
<xsd:enumeration value="Tech House"/>
<xsd:enumeration value="Vocal House"/>
<xsd:enumeration value="Big Beat"/>
<xsd:enumeration value="Breakbeat"/>
```

```
<xsd:enumeration value="Drum'n'Bass"/>
<xsd:enumeration value="Dubstep"/>
<xsd:enumeration value="Garage / UK Funky"/>
<xsd:enumeration value="IDM"/>
<xsd:enumeration value="Trip-Hop"/>
<xsd:enumeration value="Trance"/>
<xsd:enumeration value="Goa Trance"/>
<xsd:enumeration value="Hard Trance"/>
<xsd:enumeration value="Psychedelic Trance"/>
<xsd:enumeration value="Gabba"/>
<xsd:enumeration value="Jumpstyle / Hardstyle"/>
<xsd:enumeration value="Classic"/>
<xsd:enumeration value="12-Tone Composition"/>
<xsd:enumeration value="Baroque"/>
<xsd:enumeration value="Chamber Music"/>
<xsd:enumeration value="Crossover / Popular Classicism"/>
<xsd:enumeration value="Electronic Music / Computer Music"/>
<xsd:enumeration value="Minimal Music"/>
<xsd:enumeration value="Modern, 20th / 21st Century"/>
<xsd:enumeration value="Music and other Media / Arts"/>
<xsd:enumeration value="Music and Word"/>
<xsd:enumeration value="Neo-Romanticism"/>
<xsd:enumeration value="Neoclassicism"/>
<xsd:enumeration value="New Music / Contemporary Music"/>
<xsd:enumeration value="Postmodern Music"/>
<xsd:enumeration value="Renaissance"/>
<xsd:enumeration value="Romantic"/>
<xsd:enumeration value="Serialism"/>
<xsd:enumeration value="Waltz"/>
<xsd:enumeration value="Ballet"/>
<xsd:enumeration value="Brass Ensemble"/>
<xsd:enumeration value="Concerto / Solo Instrument with Orchestra"/>
<xsd:enumeration value="Mixed Ensemble (Strings / Wind)"/>
<xsd:enumeration value="Mixed Wind Ensemble (Woodwind / Brass)"/>
<xsd:enumeration value="Opera Arias"/>
<xsd:enumeration value="Opera Baroque"/>
<xsd:enumeration value="Opera Classical"/>
<xsd:enumeration value="Opera Renaissance"/>
<xsd:enumeration value="Opera Romantic"/>
<xsd:enumeration value="Operettas"/>
<xsd:enumeration value="Several Solo Instruments"/>
<xsd:enumeration value="Solo Instrument"/>
<xsd:enumeration value="String Ensemble"/>
<xsd:enumeration value="String Orchestra"/>
<xsd:enumeration value="String Quartet"/>
<xsd:enumeration value="String Trio"/>
<xsd:enumeration value="Symphonic Music / Orchestral Music"/>
<xsd:enumeration value="Symphonies"/>
<xsd:enumeration value="Vocal Ensemble"/>
<xsd:enumeration value="Vocal Music"/>
<xsd:enumeration value="Woodwind Ensemble"/>
<xsd:enumeration value="Choir"/>
<xsd:enumeration value="Boy's Choir"/>
<xsd:enumeration value="Children's Choir"/>
<xsd:enumeration value="Choir with Orchestra"/>
<xsd:enumeration value="Female Choir"/>
<xsd:enumeration value="Male Choir"/>
<xsd:enumeration value="Mixed Choir"/>
<xsd:enumeration value="Flute"/>
<xsd:enumeration value="Recorder / English Flute"/>
<xsd:enumeration value="Oboe"/>
<xsd:enumeration value="Clarinet"/>
<xsd:enumeration value="Bassoon"/>
<xsd:enumeration value="Saxophone"/>
<xsd:enumeration value="Horn"/>
<xsd:enumeration value="Trumpet"/>
<xsd:enumeration value="Trombone"/>
<xsd:enumeration value="Tuba"/>
<xsd:enumeration value="Violin"/>
<xsd:enumeration value="Viola"/>
<xsd:enumeration value="Violoncello"/>
<xsd:enumeration value="Double Bass"/>
<xsd:enumeration value="Guitar"/>
<xsd:enumeration value="Mandolin"/>
<xsd:enumeration value="Lute"/>
<xsd:enumeration value="Harp"/>
<xsd:enumeration value="Piano"/>
<xsd:enumeration value="Harpsichord"/>
<xsd:enumeration value="Accordion"/>
<xsd:enumeration value="Organ"/>
<xsd:enumeration value="Ancient Instruments"/>
<xsd:enumeration value="Percussion (Vibraphone etc.)"/>
<xsd:enumeration value="Miscellaneous Lead Instrument"/>
```

```
<xsd:enumeration value="Reggae"/>
<xsd:enumeration value="Contemporary"/>
<xsd:enumeration value="Dancehall"/>
<xsd:enumeration value="Dub"/>
<xsd:enumeration value="Lover's Rock"/>
<xsd:enumeration value="Reggaeton"/>
<xsd:enumeration value="Roots"/>
<xsd:enumeration value="Ska"/>
<xsd:enumeration value="World"/>
<xsd:enumeration value="African Music"/>
<xsd:enumeration value="Afrobeat"/>
<xsd:enumeration value="Asian and Middle Eastern Music"/>
<xsd:enumeration value="Calypso"/>
<xsd:enumeration value="Caribbean Music"/>
<xsd:enumeration value="Celtic, European and Judaica Music"/>
<xsd:enumeration value="Coupé Decalé"/>
<xsd:enumeration value="Griot"/>
<xsd:enumeration value="Highlife"/>
<xsd:enumeration value="Kuduro"/>
<xsd:enumeration value="Kwaito"/>
<xsd:enumeration value="Makossa"/>
<xsd:enumeration value="Marching Band"/>
<xsd:enumeration value="Mento"/>
<xsd:enumeration value="North American Music"/>
<xsd:enumeration value="Parang"/>
<xsd:enumeration value="Polka"/>
<xsd:enumeration value="Rai"/>
<xsd:enumeration value="Soca"/>
<xsd:enumeration value="Soukous"/>
<xsd:enumeration value="Zouk"/>
<xsd:enumeration value="Latin"/>
<xsd:enumeration value="Banda"/>
<xsd:enumeration value="Bolero"/>
<xsd:enumeration value="Bossa Nova"/>
<xsd:enumeration value="Corridos"/>
<xsd:enumeration value="Cumbia"/>
<xsd:enumeration value="Fado"/>
<xsd:enumeration value="Flamenco"/>
<xsd:enumeration value="Grupero"/>
<xsd:enumeration value="Mambo"/>
<xsd:enumeration value="Mariachi"/>
<xsd:enumeration value="Merengue"/>
<xsd:enumeration value="Norteno"/>
<xsd:enumeration value="Ranchero"/>
<xsd:enumeration value="Rock En Espanol"/>
<xsd:enumeration value="Salsa"/>
<xsd:enumeration value="Samba"/>
<xsd:enumeration value="Son Cubana"/>
<xsd:enumeration value="Sonidero"/>
<xsd:enumeration value="Tango"/>
<xsd:enumeration value="Tejano"/>
<xsd:enumeration value="Religious"/>
<xsd:enumeration value="Christian Rock"/>
<xsd:enumeration value="Christian Hip Hop"/>
<xsd:enumeration value="Christian Pop"/>
<xsd:enumeration value="Chants"/>
<xsd:enumeration value="Gospel"/>
<xsd:enumeration value="Hymn"/>
<xsd:enumeration value="Mass"/>
<xsd:enumeration value="Spiritual"/>
<xsd:enumeration value="Worship"/>
<xsd:enumeration value="Miscellaneous"/>
<xsd:enumeration value="Anime / Video Game Soundtracks"/>
<xsd:enumeration value="Bollywood"/>
<xsd:enumeration value="Instrumental"/>
<xsd:enumeration value="Karaoke"/>
<xsd:enumeration value="Movie Scores"/>
<xsd:enumeration value="Movie Soundtracks"/>
<xsd:enumeration value="Sound Effects"/>
<xsd:enumeration value="Soundtrack"/>
<xsd:enumeration value="TV Soundtrack"/>
<xsd:enumeration value="unclassifiable"/>
<xsd:enumeration value="Wedding Music"/>
<xsd:enumeration value="Holiday"/>
<xsd:enumeration value="Word"/>
<xsd:enumeration value="Business & Career"/>
<xsd:enumeration value="Abstracts & Dossiers"/>
<xsd:enumeration value="Accounting"/>
<xsd:enumeration value="Business & Investing"/>
<xsd:enumeration value="Communication"/>
<xsd:enumeration value="Computers & Internet"/>
<xsd:enumeration value="Economics"/>
<xsd:enumeration value="Finance"/>
```

```
<xsd:enumeration value="Management & Leadership"/>
<xsd:enumeration value="Marketing & Sales"/>
<xsd:enumeration value="Politics"/>
<xsd:enumeration value="Self-Help"/>
<xsd:enumeration value="Self-Organization"/>
<xsd:enumeration value="Skills"/>
<xsd:enumeration value="Small Business & Entrepreneurship"/>
<xsd:enumeration value="Children's Audiobooks"/>
<xsd:enumeration value="age"/>
<xsd:enumeration value="Age"/>
<xsd:enumeration value="Popular Characters"/>
<xsd:enumeration value="Animal Stories"/>
<xsd:enumeration value="Children's Book Classics"/>
<xsd:enumeration value="Detective Stories"/>
<xsd:enumeration value="Fairy Tales"/>
<xsd:enumeration value="Fantasy & Spook"/>
<xsd:enumeration value="Knowledge for Children"/>
<xsd:enumeration value="Pirates, Knights & Historical"/>
<xsd:enumeration value="Poems & Song"/>
<xsd:enumeration value="Comedy & Humour"/>
<xsd:enumeration value="Comedy & Cabaret"/>
<xsd:enumeration value="Humoristic Novel"/>
<xsd:enumeration value="Crime"/>
<xsd:enumeration value="Detective Stories"/>
<xsd:enumeration value="Detective Stories „Noir“"/>
<xsd:enumeration value="Classic Detective Stories"/>
<xsd:enumeration value="Scandinavian Detective Stories"/>
<xsd:enumeration value="Temporary Detective Stories"/>
<xsd:enumeration value="Education & Knowledge"/>
<xsd:enumeration value="Art & Culture"/>
<xsd:enumeration value="Biography & Memento"/>
<xsd:enumeration value="Foreign Language"/>
<xsd:enumeration value="History"/>
<xsd:enumeration value="Philosophy"/>
<xsd:enumeration value="Politics & Current Affairs"/>
<xsd:enumeration value="Science & Technology"/>
<xsd:enumeration value="Health, Mind & Body"/>
<xsd:enumeration value="Autogenous Training"/>
<xsd:enumeration value="Creativity"/>
<xsd:enumeration value="Fitness"/>
<xsd:enumeration value="Health"/>
<xsd:enumeration value="Lifestyle"/>
<xsd:enumeration value="Love & Erotic"/>
<xsd:enumeration value="Meditation / Yoga"/>
<xsd:enumeration value="Memory Training"/>
<xsd:enumeration value="Mental Training"/>
<xsd:enumeration value="Motivation"/>
<xsd:enumeration value="Philosophy"/>
<xsd:enumeration value="Positive Thinking & Attitude"/>
<xsd:enumeration value="Psychology"/>
<xsd:enumeration value="Spirituality & Religion"/>
<xsd:enumeration value="Sports"/>
<xsd:enumeration value="Wellness & Beauty"/>
<xsd:enumeration value="Science Fiction & Fantasy"/>
<xsd:enumeration value="Ancient World"/>
<xsd:enumeration value="Fantasy-Romance"/>
<xsd:enumeration value="Historical Thriller"/>
<xsd:enumeration value="Horror Classics"/>
<xsd:enumeration value="Medieval Times & Early Modern Era"/>
<xsd:enumeration value="Thriller"/>
<xsd:enumeration value="Mystery & Conspiracy"/>
<xsd:enumeration value="Psychological Thriller"/>
<xsd:enumeration value="Espionage, Politics & Justice"/>
<xsd:enumeration value="Vatican & Secret Societies"/>
<xsd:enumeration value="Science & Medicine"/>
<xsd:enumeration value="Novels"/>
<xsd:enumeration value="Erotica"/>
<xsd:enumeration value="Contemporary German Literature"/>
<xsd:enumeration value="Romance"/>
<xsd:enumeration value="Entertainment"/>
<xsd:enumeration value="Contemporary Literature"/>
<xsd:enumeration value="Youth"/>
<xsd:enumeration value="Detective Stories"/>
<xsd:enumeration value="Fantasy"/>
<xsd:enumeration value="For Girls"/>
<xsd:enumeration value="Knowledge for Teenagers"/>
<xsd:enumeration value="Mystery"/>
<xsd:enumeration value="Pirates, Knights & Historical"/>
<xsd:enumeration value="Youth Classics"/>
<xsd:enumeration value="Youth Today"/>
<xsd:enumeration value="Language"/>
<xsd:enumeration value="Albanian"/>
<xsd:enumeration value="Arabic"/>
```

```
<xsd:enumeration value="Bengali"/>
<xsd:enumeration value="Bosnian"/>
<xsd:enumeration value="Bulgarian"/>
<xsd:enumeration value="Cantonese/Yue"/>
<xsd:enumeration value="Croatian"/>
<xsd:enumeration value="Czech"/>
<xsd:enumeration value="Danish"/>
<xsd:enumeration value="Dutch"/>
<xsd:enumeration value="English"/>
<xsd:enumeration value="Finnish"/>
<xsd:enumeration value="French"/>
<xsd:enumeration value="German"/>
<xsd:enumeration value="Greek"/>
<xsd:enumeration value="Hebrew"/>
<xsd:enumeration value="Hindi/Urdu"/>
<xsd:enumeration value="Hungarian"/>
<xsd:enumeration value="Italian"/>
<xsd:enumeration value="Japanese"/>
<xsd:enumeration value="Korean"/>
<xsd:enumeration value="Macedonian"/>
<xsd:enumeration value="Mandarin"/>
<xsd:enumeration value="Norwegian"/>
<xsd:enumeration value="Patois"/>
<xsd:enumeration value="Portuguese"/>
<xsd:enumeration value="Russian"/>
<xsd:enumeration value="Serbian"/>
<xsd:enumeration value="Spanish"/>
<xsd:enumeration value="Swedish"/>
<xsd:enumeration value="Tamil"/>
<xsd:enumeration value="Turkish"/>
<xsd:enumeration value="Vietnamese"/>
<xsd:enumeration value="Film"/>
<xsd:enumeration value="Action"/>
<xsd:enumeration value="3D"/>
<xsd:enumeration value="Adventure"/>
<xsd:enumeration value="Animation"/>
<xsd:enumeration value="Author's Film"/>
<xsd:enumeration value="Biography"/>
<xsd:enumeration value="Cartoon"/>
<xsd:enumeration value="Children"/>
<xsd:enumeration value="Comedy"/>
<xsd:enumeration value="Crime & Gangster"/>
<xsd:enumeration value="Disaster"/>
<xsd:enumeration value="Documentary"/>
<xsd:enumeration value="Drama"/>
<xsd:enumeration value="Epic / Historical"/>
<xsd:enumeration value="Erotic"/>
<xsd:enumeration value="Expressionism"/>
<xsd:enumeration value="Family"/>
<xsd:enumeration value="Fantasy"/>
<xsd:enumeration value="Film-Noir"/>
<xsd:enumeration value="GLBT"/>
<xsd:enumeration value="Horror"/>
<xsd:enumeration value="Independent Film"/>
<xsd:enumeration value="Martial-Arts / Eastern"/>
<xsd:enumeration value="Monumental"/>
<xsd:enumeration value="Musical / Dance"/>
<xsd:enumeration value="Music"/>
<xsd:enumeration value="Mystery"/>
<xsd:enumeration value="Reality-TV"/>
<xsd:enumeration value="Romantic"/>
<xsd:enumeration value="Science Fiction"/>
<xsd:enumeration value="Silent Movie"/>
<xsd:enumeration value="Sport"/>
<xsd:enumeration value="Thriller"/>
<xsd:enumeration value="TV-Series"/>
<xsd:enumeration value="Tragicomedy"/>
<xsd:enumeration value="War / Anti-War"/>
<xsd:enumeration value="Western"/>
<xsd:enumeration value="Youth"/>
<xsd:enumeration value="Country"/>
<xsd:enumeration value="United Arab Emirates (AE)"/>
<xsd:enumeration value="Afghanistan (AF)"/>
<xsd:enumeration value="Antigua and Barbuda (AG)"/>
<xsd:enumeration value="Anguilla (AI)"/>
<xsd:enumeration value="Albania (AL)"/>
<xsd:enumeration value="Armenia (AM)"/>
<xsd:enumeration value="Angola (AO)"/>
<xsd:enumeration value="Antarctica (AQ)"/>
<xsd:enumeration value="Argentina (AR)"/>
<xsd:enumeration value="American Samoa (AS)"/>
<xsd:enumeration value="Austria (AT)"/>
<xsd:enumeration value="Australia (AU)"/>
```

```
<xsd:enumeration value="Aruba (AW)" />
<xsd:enumeration value="Åland Islands (AX)" />
<xsd:enumeration value="Azerbaijan (AZ)" />
<xsd:enumeration value="Bosnia and Herzegovina (BA)" />
<xsd:enumeration value="Barbados (BB)" />
<xsd:enumeration value="Bangladesh (BD)" />
<xsd:enumeration value="Belgium (BE)" />
<xsd:enumeration value="Burkina Faso (BF)" />
<xsd:enumeration value="Bulgaria (BG)" />
<xsd:enumeration value="Bahrain (BH)" />
<xsd:enumeration value="Burundi (BI)" />
<xsd:enumeration value="Benin (BJ)" />
<xsd:enumeration value="Saint Barthélemy (BL)" />
<xsd:enumeration value="Bermuda (BM)" />
<xsd:enumeration value="Brunei Darussalam (BN)" />
<xsd:enumeration value="Bolivia Plurinational State of (BO)" />
<xsd:enumeration value="Bonaire Saint Eustatius and Saba (BQ)" />
<xsd:enumeration value="Brazil (BR)" />
<xsd:enumeration value="Bahamas (BS)" />
<xsd:enumeration value="Bhutan (BT)" />
<xsd:enumeration value="Bouvet Island (BV)" />
<xsd:enumeration value="Botswana (BW)" />
<xsd:enumeration value="Belarus (BY)" />
<xsd:enumeration value="Belize (BZ)" />
<xsd:enumeration value="Canada (CA)" />
<xsd:enumeration value="Cocos (Keeling) Islands (CC)" />
<xsd:enumeration value="Congo the Democratic Republic of the (CD)" />
<xsd:enumeration value="Central African Republic (CF)" />
<xsd:enumeration value="Congo (CG)" />
<xsd:enumeration value="Switzerland (CH)" />
<xsd:enumeration value="Côte d'Ivoire (CI)" />
<xsd:enumeration value="Cook Islands (CK)" />
<xsd:enumeration value="Chile (CL)" />
<xsd:enumeration value="Cameroon (CM)" />
<xsd:enumeration value="China (CN)" />
<xsd:enumeration value="Colombia (CO)" />
<xsd:enumeration value="Costa Rica (CR)" />
<xsd:enumeration value="Cuba (CU)" />
<xsd:enumeration value="Cape Verde (CV)" />
<xsd:enumeration value="Curaçao (CW)" />
<xsd:enumeration value="Christmas Island (CX)" />
<xsd:enumeration value="Cyprus (CY)" />
<xsd:enumeration value="Czech Republic (CZ)" />
<xsd:enumeration value="Germany (DE)" />
<xsd:enumeration value="Djibouti (DJ)" />
<xsd:enumeration value="Denmark (DK)" />
<xsd:enumeration value="Dominica (DM)" />
<xsd:enumeration value="Dominican Republic (DO)" />
<xsd:enumeration value="Algeria (DZ)" />
<xsd:enumeration value="Ecuador (EC)" />
<xsd:enumeration value="Estonia (EE)" />
<xsd:enumeration value="Egypt (EG)" />
<xsd:enumeration value="Western Sahara (EH)" />
<xsd:enumeration value="Eritrea (ER)" />
<xsd:enumeration value="Spain (ES)" />
<xsd:enumeration value="Ethiopia (ET)" />
<xsd:enumeration value="Finland (FI)" />
<xsd:enumeration value="Fiji (FJ)" />
<xsd:enumeration value="Falkland Islands (Malvinas) (FK)" />
<xsd:enumeration value="Micronesia Federated States of (FM)" />
<xsd:enumeration value="Faroe Islands (FO)" />
<xsd:enumeration value="France (FR)" />
<xsd:enumeration value="Gabon (GA)" />
<xsd:enumeration value="United Kingdom (GB)" />
<xsd:enumeration value="Grenada (GD)" />
<xsd:enumeration value="Georgia (GE)" />
<xsd:enumeration value="French Guiana (GF)" />
<xsd:enumeration value="Guernsey (GG)" />
<xsd:enumeration value="Ghana (GH)" />
<xsd:enumeration value="Gibraltar (GI)" />
<xsd:enumeration value="Greenland (GL)" />
<xsd:enumeration value="Gambia (GM)" />
<xsd:enumeration value="Guinea (GN)" />
<xsd:enumeration value="Guadeloupe (GP)" />
<xsd:enumeration value="Equatorial Guinea (GQ)" />
<xsd:enumeration value="Greece (GR)" />
<xsd:enumeration value="South Georgia and the South Sandwich Islands (GS)" />
<xsd:enumeration value="Guatemala (GT)" />
<xsd:enumeration value="Guam (GU)" />
<xsd:enumeration value="Guinea-Bissau (GW)" />
<xsd:enumeration value="Guyana (GY)" />
<xsd:enumeration value="Hong Kong (HK)" />
<xsd:enumeration value="Heard Island and McDonald Islands (HM)" />
```

```
<xsd:enumeration value="Honduras (HN)"/>
<xsd:enumeration value="Croatia (HR)"/>
<xsd:enumeration value="Haiti (HT)"/>
<xsd:enumeration value="Hungary (HU)"/>
<xsd:enumeration value="Indonesia (ID)"/>
<xsd:enumeration value="Ireland (IE)"/>
<xsd:enumeration value="Israel (IL)"/>
<xsd:enumeration value="Isle of Man (IM)"/>
<xsd:enumeration value="India (IN)"/>
<xsd:enumeration value="British Indian Ocean Territory (IO)"/>
<xsd:enumeration value="Iraq (IQ)"/>
<xsd:enumeration value="Iran Islamic Republic of (IR)"/>
<xsd:enumeration value="Iceland (IS)"/>
<xsd:enumeration value="Italy (IT)"/>
<xsd:enumeration value="Jersey (JE)"/>
<xsd:enumeration value="Jamaica (JM)"/>
<xsd:enumeration value="Jordan (JO)"/>
<xsd:enumeration value="Japan (JP)"/>
<xsd:enumeration value="Kenya (KE)"/>
<xsd:enumeration value="Kyrgyzstan (KG)"/>
<xsd:enumeration value="Cambodia (KH)"/>
<xsd:enumeration value="Kiribati (KI)"/>
<xsd:enumeration value="Comoros (KM)"/>
<xsd:enumeration value="Saint Kitts and Nevis (KN)"/>
<xsd:enumeration value="Korea Democratic People's Republic of (KP)"/>
<xsd:enumeration value="Korea Republic of (KR)"/>
<xsd:enumeration value="Kuwait (KW)"/>
<xsd:enumeration value="Cayman Islands (KY)"/>
<xsd:enumeration value="Kazakhstan (KZ)"/>
<xsd:enumeration value="Lao People's Democratic Republic (LA)"/>
<xsd:enumeration value="Lebanon (LB)"/>
<xsd:enumeration value="Saint Lucia (LC)"/>
<xsd:enumeration value="Liechtenstein (LI)"/>
<xsd:enumeration value="Sri Lanka (LK)"/>
<xsd:enumeration value="Liberia (LR)"/>
<xsd:enumeration value="Lesotho (LS)"/>
<xsd:enumeration value="Lithuania (LT)"/>
<xsd:enumeration value="Luxembourg (LU)"/>
<xsd:enumeration value="Latvia (LV)"/>
<xsd:enumeration value="Libyan Arab Jamahiriya (LY)"/>
<xsd:enumeration value="Morocco (MA)"/>
<xsd:enumeration value="Monaco (MC)"/>
<xsd:enumeration value="Moldova Republic of (MD)"/>
<xsd:enumeration value="Montenegro (ME)"/>
<xsd:enumeration value="Saint Martin (French part) (MF)"/>
<xsd:enumeration value="Madagascar (MG)"/>
<xsd:enumeration value="Marshall Islands (MH)"/>
<xsd:enumeration value="Macedonia the former Yugoslav Republic of (MK)"/>
<xsd:enumeration value="Mali (ML)"/>
<xsd:enumeration value="Myanmar (MM)"/>
<xsd:enumeration value="Mongolia (MN)"/>
<xsd:enumeration value="Macao (MO)"/>
<xsd:enumeration value="Northern Mariana Islands (MP)"/>
<xsd:enumeration value="Martinique (MQ)"/>
<xsd:enumeration value="Mauritania (MR)"/>
<xsd:enumeration value="Montserrat (MS)"/>
<xsd:enumeration value="Malta (MT)"/>
<xsd:enumeration value="Mauritius (MU)"/>
<xsd:enumeration value="Maldives (MV)"/>
<xsd:enumeration value="Malawi (MW)"/>
<xsd:enumeration value="Mexico (MX)"/>
<xsd:enumeration value="Malaysia (MY)"/>
<xsd:enumeration value="Mozambique (MZ)"/>
<xsd:enumeration value="Namibia (NA)"/>
<xsd:enumeration value="New Caledonia (NC)"/>
<xsd:enumeration value="Niger (NE)"/>
<xsd:enumeration value="Norfolk Island (NF)"/>
<xsd:enumeration value="Nigeria (NG)"/>
<xsd:enumeration value="Nicaragua (NI)"/>
<xsd:enumeration value="Netherlands (NL)"/>
<xsd:enumeration value="Norway (NO)"/>
<xsd:enumeration value="Nepal (NP)"/>
<xsd:enumeration value="Nauru (NR)"/>
<xsd:enumeration value="Niue (NU)"/>
<xsd:enumeration value="New Zealand (NZ)"/>
<xsd:enumeration value="Oman (OM)"/>
<xsd:enumeration value="Panama (PA)"/>
<xsd:enumeration value="Peru (PE)"/>
<xsd:enumeration value="French Polynesia (PF)"/>
<xsd:enumeration value="Papua New Guinea (PG)"/>
<xsd:enumeration value="Philippines (PH)"/>
<xsd:enumeration value="Pakistan (PK)"/>
<xsd:enumeration value="Poland (PL)"/>
```

```

<xsd:enumeration value="Saint Pierre and Miquelon (PM)" />
<xsd:enumeration value="Pitcairn (PN)" />
<xsd:enumeration value="Puerto Rico (PR)" />
<xsd:enumeration value="Palestinian Territory Occupied (PS)" />
<xsd:enumeration value="Portugal (PT)" />
<xsd:enumeration value="Palau (PW)" />
<xsd:enumeration value="Paraguay (PY)" />
<xsd:enumeration value="Qatar (QA)" />
<xsd:enumeration value="Réunion (RE)" />
<xsd:enumeration value="Romania (RO)" />
<xsd:enumeration value="Serbia (RS)" />
<xsd:enumeration value="Russian Federation (RU)" />
<xsd:enumeration value="Rwanda (RW)" />
<xsd:enumeration value="Saudi Arabia (SA)" />
<xsd:enumeration value="Solomon Islands (SB)" />
<xsd:enumeration value="Seychelles (SC)" />
<xsd:enumeration value="Sudan (SD)" />
<xsd:enumeration value="Sweden (SE)" />
<xsd:enumeration value="Singapore (SG)" />
<xsd:enumeration value="Saint Helena Ascension and Tristan da Cunha (SH)" />
<xsd:enumeration value="Slovenia (SI)" />
<xsd:enumeration value="Svalbard and Jan Mayen (SJ)" />
<xsd:enumeration value="Slovakia (SK)" />
<xsd:enumeration value="Sierra Leone (SL)" />
<xsd:enumeration value="San Marino (SM)" />
<xsd:enumeration value="Senegal (SN)" />
<xsd:enumeration value="Somalia (SO)" />
<xsd:enumeration value="Suriname (SR)" />
<xsd:enumeration value="Sao Tome and Principe (ST)" />
<xsd:enumeration value="El Salvador (SV)" />
<xsd:enumeration value="Sint Maarten (Dutch part) (SX)" />
<xsd:enumeration value="Syrian Arab Republic (SY)" />
<xsd:enumeration value="Swaziland (SZ)" />
<xsd:enumeration value="Turks and Caicos Islands (TC)" />
<xsd:enumeration value="Chad (TD)" />
<xsd:enumeration value="French Southern Territories (TF)" />
<xsd:enumeration value="Togo (TG)" />
<xsd:enumeration value="Thailand (TH)" />
<xsd:enumeration value="Tajikistan (TJ)" />
<xsd:enumeration value="Tokelau (TK)" />
<xsd:enumeration value="Timor-Leste (TL)" />
<xsd:enumeration value="Turkmenistan (TM)" />
<xsd:enumeration value="Tunisia (TN)" />
<xsd:enumeration value="Tonga (TO)" />
<xsd:enumeration value="Turkey (TR)" />
<xsd:enumeration value="Trinidad and Tobago (TT)" />
<xsd:enumeration value="Tuvalu (TV)" />
<xsd:enumeration value="Taiwan Province of China (TW)" />
<xsd:enumeration value="Tanzania United Republic of (TZ)" />
<xsd:enumeration value="Ukraine (UA)" />
<xsd:enumeration value="Uganda (UG)" />
<xsd:enumeration value="United States Minor Outlying Islands (UM)" />
<xsd:enumeration value="United States (US)" />
<xsd:enumeration value="Uruguay (UY)" />
<xsd:enumeration value="Uzbekistan (UZ)" />
<xsd:enumeration value="Holy See (Vatican City State) (VA)" />
<xsd:enumeration value="Saint Vincent and the Grenadines (VC)" />
<xsd:enumeration value="Venezuela Bolivarian Republic of (VE)" />
<xsd:enumeration value="Virgin Islands British (VG)" />
<xsd:enumeration value="Virgin Islands U.S. (VI)" />
<xsd:enumeration value="Viet Nam (VN)" />
<xsd:enumeration value="Vanuatu (VU)" />
<xsd:enumeration value="Wallis and Futuna (WF)" />
<xsd:enumeration value="Samoa (WS)" />
<xsd:enumeration value="Yemen (YE)" />
<xsd:enumeration value="Mayotte (YT)" />
<xsd:enumeration value="South Africa (ZA)" />
<xsd:enumeration value="Zambia (ZM)" />
<xsd:enumeration value="Zimbabwe (ZW)" />
</xsd:restriction>
</xsd:simpleType>
```

Namespace: "http://fnppl.org/opensdx/languages"

Schema(s)

Imported schema openSDX_languages.xsd

Namespace	http://fnppl.org/opensdx/languages
Properties	attribute form default: unqualified

element form default: unqualified

Simple Type(s)

Simple Type language

Namespace	http://fnppl.org/opensdx/languages																																																																																																																			
Annotations	This element includes a list of ISO 639-1 language codes.																																																																																																																			
Diagram	<pre> classDiagram class language { <<ISO 639-1 language codes>> } xsd:string language "1" -- "1" xsd:string </pre> <p>This element includes a list of ISO 639-1 language codes.</p> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>																																																																																																																			
Type	restriction of xsd:string																																																																																																																			
Facets	<table> <tr><td>enumeration</td><td>aa</td><td>Afar</td></tr> <tr><td>enumeration</td><td>ab</td><td>Abkhazian</td></tr> <tr><td>enumeration</td><td>af</td><td>Afrikaans</td></tr> <tr><td>enumeration</td><td>am</td><td>Amharic</td></tr> <tr><td>enumeration</td><td>ar</td><td>Arabic</td></tr> <tr><td>enumeration</td><td>as</td><td>Assamese</td></tr> <tr><td>enumeration</td><td>ay</td><td>Aymara</td></tr> <tr><td>enumeration</td><td>az</td><td>Azerbaijani</td></tr> <tr><td>enumeration</td><td>ba</td><td>Bashkir</td></tr> <tr><td>enumeration</td><td>be</td><td>Byelorussian</td></tr> <tr><td>enumeration</td><td>bg</td><td>Bulgarian</td></tr> <tr><td>enumeration</td><td>bh</td><td>Bihari</td></tr> <tr><td>enumeration</td><td>bi</td><td>Bislama</td></tr> <tr><td>enumeration</td><td>bn</td><td>Bengali; Bangla</td></tr> <tr><td>enumeration</td><td>bo</td><td>Tibetan</td></tr> <tr><td>enumeration</td><td>br</td><td>Breton</td></tr> <tr><td>enumeration</td><td>ca</td><td>Catalan</td></tr> <tr><td>enumeration</td><td>co</td><td>Corsican</td></tr> <tr><td>enumeration</td><td>cs</td><td>Czech</td></tr> <tr><td>enumeration</td><td>cy</td><td>Welsh</td></tr> <tr><td>enumeration</td><td>da</td><td>Danish</td></tr> <tr><td>enumeration</td><td>de</td><td>German</td></tr> <tr><td>enumeration</td><td>dz</td><td>Bhutani</td></tr> <tr><td>enumeration</td><td>el</td><td>Greek</td></tr> <tr><td>enumeration</td><td>en</td><td>English</td></tr> <tr><td>enumeration</td><td>eo</td><td>Esperanto</td></tr> <tr><td>enumeration</td><td>es</td><td>Spanish</td></tr> <tr><td>enumeration</td><td>et</td><td>Estonian</td></tr> <tr><td>enumeration</td><td>eu</td><td>Basque</td></tr> <tr><td>enumeration</td><td>fa</td><td>Persian</td></tr> <tr><td>enumeration</td><td>fi</td><td>Finnish</td></tr> <tr><td>enumeration</td><td>fj</td><td>Fiji</td></tr> <tr><td>enumeration</td><td>fo</td><td>Faroese</td></tr> <tr><td>enumeration</td><td>fr</td><td>French</td></tr> <tr><td>enumeration</td><td>fy</td><td>Frisian</td></tr> <tr><td>enumeration</td><td>ga</td><td>Irish</td></tr> <tr><td>enumeration</td><td>gd</td><td>Scots Gaelic</td></tr> <tr><td>enumeration</td><td>gl</td><td>Galician</td></tr> </table>		enumeration	aa	Afar	enumeration	ab	Abkhazian	enumeration	af	Afrikaans	enumeration	am	Amharic	enumeration	ar	Arabic	enumeration	as	Assamese	enumeration	ay	Aymara	enumeration	az	Azerbaijani	enumeration	ba	Bashkir	enumeration	be	Byelorussian	enumeration	bg	Bulgarian	enumeration	bh	Bihari	enumeration	bi	Bislama	enumeration	bn	Bengali; Bangla	enumeration	bo	Tibetan	enumeration	br	Breton	enumeration	ca	Catalan	enumeration	co	Corsican	enumeration	cs	Czech	enumeration	cy	Welsh	enumeration	da	Danish	enumeration	de	German	enumeration	dz	Bhutani	enumeration	el	Greek	enumeration	en	English	enumeration	eo	Esperanto	enumeration	es	Spanish	enumeration	et	Estonian	enumeration	eu	Basque	enumeration	fa	Persian	enumeration	fi	Finnish	enumeration	fj	Fiji	enumeration	fo	Faroese	enumeration	fr	French	enumeration	fy	Frisian	enumeration	ga	Irish	enumeration	gd	Scots Gaelic	enumeration	gl	Galician
enumeration	aa	Afar																																																																																																																		
enumeration	ab	Abkhazian																																																																																																																		
enumeration	af	Afrikaans																																																																																																																		
enumeration	am	Amharic																																																																																																																		
enumeration	ar	Arabic																																																																																																																		
enumeration	as	Assamese																																																																																																																		
enumeration	ay	Aymara																																																																																																																		
enumeration	az	Azerbaijani																																																																																																																		
enumeration	ba	Bashkir																																																																																																																		
enumeration	be	Byelorussian																																																																																																																		
enumeration	bg	Bulgarian																																																																																																																		
enumeration	bh	Bihari																																																																																																																		
enumeration	bi	Bislama																																																																																																																		
enumeration	bn	Bengali; Bangla																																																																																																																		
enumeration	bo	Tibetan																																																																																																																		
enumeration	br	Breton																																																																																																																		
enumeration	ca	Catalan																																																																																																																		
enumeration	co	Corsican																																																																																																																		
enumeration	cs	Czech																																																																																																																		
enumeration	cy	Welsh																																																																																																																		
enumeration	da	Danish																																																																																																																		
enumeration	de	German																																																																																																																		
enumeration	dz	Bhutani																																																																																																																		
enumeration	el	Greek																																																																																																																		
enumeration	en	English																																																																																																																		
enumeration	eo	Esperanto																																																																																																																		
enumeration	es	Spanish																																																																																																																		
enumeration	et	Estonian																																																																																																																		
enumeration	eu	Basque																																																																																																																		
enumeration	fa	Persian																																																																																																																		
enumeration	fi	Finnish																																																																																																																		
enumeration	fj	Fiji																																																																																																																		
enumeration	fo	Faroese																																																																																																																		
enumeration	fr	French																																																																																																																		
enumeration	fy	Frisian																																																																																																																		
enumeration	ga	Irish																																																																																																																		
enumeration	gd	Scots Gaelic																																																																																																																		
enumeration	gl	Galician																																																																																																																		

enumeration	gn	Guarani
enumeration	gu	Gujarati
enumeration	ha	Hausa
enumeration	he	Hebrew
enumeration	hi	Hindi
enumeration	hr	Croatian
enumeration	hu	Hungarian
enumeration	hy	Armenian
enumeration	ia	Interlingua
enumeration	id	Indonesian
enumeration	ie	Interlingue
enumeration	ik	Inupiak
enumeration	is	Icelandic
enumeration	it	Italian
enumeration	iu	Inuktitut
enumeration	ja	Japanese
enumeration	jw	Javanese
enumeration	ka	Georgian
enumeration	kk	Kazakh
enumeration	kl	Greenlandic
enumeration	km	Cambodian
enumeration	kn	Kannada
enumeration	ko	Korean
enumeration	ks	Kashmiri
enumeration	ku	Kurdish
enumeration	ky	Kirghiz
enumeration	la	Latin
enumeration	ln	Lingala
enumeration	lo	Laothian
enumeration	lt	Lithuanian
enumeration	lv	Latvian; Lettish
enumeration	mg	Malagasy
enumeration	mi	Maori
enumeration	mk	Macedonian
enumeration	ml	Malayalam
enumeration	mn	Mongolian
enumeration	mo	Moldavian
enumeration	mr	Marathi
enumeration	ms	Malay
enumeration	mt	Maltese
enumeration	my	Burmese
enumeration	na	Nauru
enumeration	ne	Nepali
enumeration	nl	Dutch
enumeration	no	Norwegian
enumeration	oc	Occitan
enumeration	om	(Afan) Oromo
enumeration	or	Oriya
enumeration	pa	Punjabi

enumeration	pl	Polish
enumeration	ps	Pashto, Pushto
enumeration	pt	Portuguese
enumeration	qu	Quechua
enumeration	rm	Rhaeto-Romance
enumeration	rn	Kirundi
enumeration	ro	Romanian
enumeration	ru	Russian
enumeration	rw	Kinyarwanda
enumeration	sa	Sanskrit
enumeration	sd	Sindhi
enumeration	sg	Sangho
enumeration	sh	Serbo-Croatian
enumeration	si	Singhalese
enumeration	sk	Slovak
enumeration	sl	Slovenian
enumeration	sm	Samoan
enumeration	sn	Shona
enumeration	so	Somali
enumeration	sq	Albanian
enumeration	sr	Serbian
enumeration	ss	Siswati
enumeration	st	Sesotho
enumeration	su	Sundanese
enumeration	sv	Swedish
enumeration	sw	Swahili
enumeration	ta	Tamil
enumeration	te	Telugu
enumeration	tg	Tajik
enumeration	th	Thai
enumeration	ti	Tigrinya
enumeration	tk	Turkmen
enumeration	tl	Tagalog
enumeration	tn	Setswana
enumeration	to	Tonga
enumeration	tr	Turkish
enumeration	ts	Tsonga
enumeration	tt	Tatar
enumeration	tw	Twi
enumeration	ug	Uigur
enumeration	uk	Ukrainian
enumeration	ur	Urdu
enumeration	uz	Uzbek
enumeration	vi	Vietnamese
enumeration	vo	Volapuk
enumeration	wo	Wolof
enumeration	xh	Xhosa
enumeration	yi	Yiddish
enumeration	yo	Yoruba

	enumeration	za	Zhuang
	enumeration	zh	Chinese
	enumeration	zu	Zulu
Used by	Element	information/main_language	
Source	<pre> <xsd:simpleType name="language"> <xsd:annotation> <xsd:documentation xml:lang="en">This element includes a list of ISO 639-1 language codes.</xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration value="aa"> <xsd:annotation> <xsd:documentation>Afar</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="ab"> <xsd:annotation> <xsd:documentation>Abkhazian</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="af"> <xsd:annotation> <xsd:documentation>Afrikaans</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="am"> <xsd:annotation> <xsd:documentation>Amharic</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="ar"> <xsd:annotation> <xsd:documentation>Arabic</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="as"> <xsd:annotation> <xsd:documentation>Assamese</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="ay"> <xsd:annotation> <xsd:documentation>Aymara</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="az"> <xsd:annotation> <xsd:documentation>Azerbaijani</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="ba"> <xsd:annotation> <xsd:documentation>Bashkir</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="be"> <xsd:annotation> <xsd:documentation>Byelorussian</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="bg"> <xsd:annotation> <xsd:documentation>Bulgarian</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="bh"> <xsd:annotation> <xsd:documentation>Bihari</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="bi"> <xsd:annotation> <xsd:documentation>Bislama</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="bn"> <xsd:annotation> <xsd:documentation>Bengali; Bangla</xsd:documentation> </xsd:annotation> </xsd:enumeration> </xsd:restriction> </xsd:simpleType></pre>		

```
</xsd:enumeration>
<xsd:enumeration value="bo">
  <xsd:annotation>
    <xsd:documentation>Tibetan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="br">
  <xsd:annotation>
    <xsd:documentation>Breton</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ca">
  <xsd:annotation>
    <xsd:documentation>Catalan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="co">
  <xsd:annotation>
    <xsd:documentation>Corsican</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="cs">
  <xsd:annotation>
    <xsd:documentation>Czech</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="cy">
  <xsd:annotation>
    <xsd:documentation>Welsh</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="da">
  <xsd:annotation>
    <xsd:documentation>Danish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="de">
  <xsd:annotation>
    <xsd:documentation>German</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="dz">
  <xsd:annotation>
    <xsd:documentation>Bhutani</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="el">
  <xsd:annotation>
    <xsd:documentation>Greek</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="en">
  <xsd:annotation>
    <xsd:documentation>English</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="eo">
  <xsd:annotation>
    <xsd:documentation>Esperanto</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="es">
  <xsd:annotation>
    <xsd:documentation>Spanish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="et">
  <xsd:annotation>
    <xsd:documentation>Estonian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="eu">
  <xsd:annotation>
    <xsd:documentation>Basque</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="fa">
  <xsd:annotation>
    <xsd:documentation>Persian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="fi">
  <xsd:annotation>
```

```
<xsd:documentation>Finnish</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="fj">
  <xsd:annotation>
    <xsd:documentation>Fiji</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="fo">
  <xsd:annotation>
    <xsd:documentation>Faroese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="fr">
  <xsd:annotation>
    <xsd:documentation>French</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="fy">
  <xsd:annotation>
    <xsd:documentation>Frisian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ga">
  <xsd:annotation>
    <xsd:documentation>Irish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="gd">
  <xsd:annotation>
    <xsd:documentation>Scots Gaelic</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="gl">
  <xsd:annotation>
    <xsd:documentation>Galician</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="gn">
  <xsd:annotation>
    <xsd:documentation>Guarani</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="gu">
  <xsd:annotation>
    <xsd:documentation>Gujarati</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ha">
  <xsd:annotation>
    <xsd:documentation>Hausa</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="he">
  <xsd:annotation>
    <xsd:documentation>Hebrew</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="hi">
  <xsd:annotation>
    <xsd:documentation>Hindi</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="hr">
  <xsd:annotation>
    <xsd:documentation>Croatian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="hu">
  <xsd:annotation>
    <xsd:documentation>Hungarian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="hy">
  <xsd:annotation>
    <xsd:documentation>Armenian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ia">
  <xsd:annotation>
    <xsd:documentation>Interlingua</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
```

```
<xsd:enumeration value="id">
  <xsd:annotation>
    <xsd:documentation>Indonesian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ie">
  <xsd:annotation>
    <xsd:documentation>Interlingue</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ik">
  <xsd:annotation>
    <xsd:documentation>Inupiak</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="is">
  <xsd:annotation>
    <xsd:documentation>Icelandic</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="it">
  <xsd:annotation>
    <xsd:documentation>Italian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="iu">
  <xsd:annotation>
    <xsd:documentation>Inuktitut</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ja">
  <xsd:annotation>
    <xsd:documentation>Japanese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="jw">
  <xsd:annotation>
    <xsd:documentation>Javanese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ka">
  <xsd:annotation>
    <xsd:documentation>Georgian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="kk">
  <xsd:annotation>
    <xsd:documentation>Kazakh</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="kl">
  <xsd:annotation>
    <xsd:documentation>Greenlandic</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="km">
  <xsd:annotation>
    <xsd:documentation>Cambodian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="kn">
  <xsd:annotation>
    <xsd:documentation>Kannada</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ko">
  <xsd:annotation>
    <xsd:documentation>Korean</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ks">
  <xsd:annotation>
    <xsd:documentation>Kashmiri</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ku">
  <xsd:annotation>
    <xsd:documentation>Kurdish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ky">
  <xsd:annotation>
    <xsd:documentation>Kirghiz</xsd:documentation>
```

```
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="la">
  <xsd:annotation>
    <xsd:documentation>Latin</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ln">
  <xsd:annotation>
    <xsd:documentation>Lingala</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="lo">
  <xsd:annotation>
    <xsd:documentation>Laothian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="lt">
  <xsd:annotation>
    <xsd:documentation>Lithuanian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="lv">
  <xsd:annotation>
    <xsd:documentation>Latvian; Lettish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="mg">
  <xsd:annotation>
    <xsd:documentation>Malagasy</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="mi">
  <xsd:annotation>
    <xsd:documentation>Maori</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="mk">
  <xsd:annotation>
    <xsd:documentation>Macedonian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ml">
  <xsd:annotation>
    <xsd:documentation>Malayalam</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="mn">
  <xsd:annotation>
    <xsd:documentation>Mongolian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="mo">
  <xsd:annotation>
    <xsd:documentation>Moldavian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="mr">
  <xsd:annotation>
    <xsd:documentation>Marathi</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ms">
  <xsd:annotation>
    <xsd:documentation>Malay</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="mt">
  <xsd:annotation>
    <xsd:documentation>Maltese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="my">
  <xsd:annotation>
    <xsd:documentation>Burmese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="na">
  <xsd:annotation>
    <xsd:documentation>Nauru</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ne">
```

```
<xsd:annotation>
  <xsd:documentation>Nepali</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="nl">
  <xsd:annotation>
    <xsd:documentation>Dutch</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="no">
  <xsd:annotation>
    <xsd:documentation>Norwegian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="oc">
  <xsd:annotation>
    <xsd:documentation>Occitan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="om">
  <xsd:annotation>
    <xsd:documentation>(Afan) Oromo</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="or">
  <xsd:annotation>
    <xsd:documentation>Oriya</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="pa">
  <xsd:annotation>
    <xsd:documentation>Punjabi</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="pl">
  <xsd:annotation>
    <xsd:documentation>Polish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ps">
  <xsd:annotation>
    <xsd:documentation>Pashto, Pushto</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="pt">
  <xsd:annotation>
    <xsd:documentation>Portuguese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="qu">
  <xsd:annotation>
    <xsd:documentation>Quechua</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="rm">
  <xsd:annotation>
    <xsd:documentation>Rhaeto-Romance</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="rn">
  <xsd:annotation>
    <xsd:documentation>Kirundi</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ro">
  <xsd:annotation>
    <xsd:documentation>Romanian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ru">
  <xsd:annotation>
    <xsd:documentation>Russian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="rw">
  <xsd:annotation>
    <xsd:documentation>Kinyarwanda</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sa">
  <xsd:annotation>
    <xsd:documentation>Sanskrit</xsd:documentation>
  </xsd:annotation>
```

```
</xsd:enumeration>
<xsd:enumeration value="sd">
  <xsd:annotation>
    <xsd:documentation>Sindhi</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sg">
  <xsd:annotation>
    <xsd:documentation>Sangho</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sh">
  <xsd:annotation>
    <xsd:documentation>Serbo-Croatian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="si">
  <xsd:annotation>
    <xsd:documentation>Singhalese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sk">
  <xsd:annotation>
    <xsd:documentation>Slovak</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sl">
  <xsd:annotation>
    <xsd:documentation>Slovenian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sm">
  <xsd:annotation>
    <xsd:documentation>Samoan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sn">
  <xsd:annotation>
    <xsd:documentation>Shona</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="so">
  <xsd:annotation>
    <xsd:documentation>Somali</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sq">
  <xsd:annotation>
    <xsd:documentation>Albanian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sr">
  <xsd:annotation>
    <xsd:documentation>Serbian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ss">
  <xsd:annotation>
    <xsd:documentation>Siswati</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="st">
  <xsd:annotation>
    <xsd:documentation>Sesotho</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="su">
  <xsd:annotation>
    <xsd:documentation>Sundanese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sv">
  <xsd:annotation>
    <xsd:documentation>Swedish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="sw">
  <xsd:annotation>
    <xsd:documentation>Swahili</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ta">
  <xsd:annotation>
```

```
<xsd:documentation>Tamil</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="te">
  <xsd:annotation>
    <xsd:documentation>Telugu</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="tg">
  <xsd:annotation>
    <xsd:documentation>Tajik</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="th">
  <xsd:annotation>
    <xsd:documentation>Thai</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ti">
  <xsd:annotation>
    <xsd:documentation>Tigrinya</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="tk">
  <xsd:annotation>
    <xsd:documentation>Turkmen</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="tl">
  <xsd:annotation>
    <xsd:documentation>Tagalog</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="tn">
  <xsd:annotation>
    <xsd:documentation>Setswana</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="to">
  <xsd:annotation>
    <xsd:documentation>Tonga</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="tr">
  <xsd:annotation>
    <xsd:documentation>Turkish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ts">
  <xsd:annotation>
    <xsd:documentation>Tsonga</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="tt">
  <xsd:annotation>
    <xsd:documentation>Tatar</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="tw">
  <xsd:annotation>
    <xsd:documentation>Twi</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ug">
  <xsd:annotation>
    <xsd:documentation>Uigur</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="uk">
  <xsd:annotation>
    <xsd:documentation>Ukrainian</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ur">
  <xsd:annotation>
    <xsd:documentation>Urdu</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="uz">
  <xsd:annotation>
    <xsd:documentation>Uzbek</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
```

```
<xsd:enumeration value="vi">
  <xsd:annotation>
    <xsd:documentation>Vietnamese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="vo">
  <xsd:annotation>
    <xsd:documentation>Volapuk</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="wo">
  <xsd:annotation>
    <xsd:documentation>Wolof</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="xh">
  <xsd:annotation>
    <xsd:documentation>Xhosa</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="yi">
  <xsd:annotation>
    <xsd:documentation>Yiddish</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="yo">
  <xsd:annotation>
    <xsd:documentation>Yoruba</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="za">
  <xsd:annotation>
    <xsd:documentation>Zhuang</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="zh">
  <xsd:annotation>
    <xsd:documentation>Chinese</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="zu">
  <xsd:annotation>
    <xsd:documentation>Zulu</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
```