

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

november 18, 2011

Table of Contents

Resource hierarchy:	6
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	10
Element receiver / type	10
Element receiver / servername	10
Element receiver / serveripv4	11
Element receiver / serveripv6	11
Element receiver / authtype	11
Element receiver / username	11
Element receiver / crypto	12
Element crypto / relatedemail	12
Element crypto / usedkeyid	12
Element crypto / usedpubkey	13
Element feedinfo / sender	13
Element sender / contractpartnerid	13
Element sender / ourcontractpartnerid	13
Element sender / email	14
Element sender / keyid	14
Element feedinfo / licensor	14
Element licensor / contractpartnerid	15
Element licensor / ourcontractpartnerid	15
Element licensor / email	15
Element licensor / keyid	15
Element feedinfo / licensee	16
Element licensee / contractpartnerid	16
Element licensee / ourcontractpartnerid	16
Element licensee / email	16
Element licensee / keyid	17
Element feedinfo / actions	17
Element actions / oninitialreceive	18
Element event / mailto	18
Element mailto / receiver	19
Element mailto / subject	19
Element mailto / text	19
Element event / http	19
Element http / url	20
Element http / type	20
Element http / addheader	20
Element http / addparams	21
Element event / fax	21
Element event / letter	21
Element letter / registered	22
Element letter / to	22
Element to / name	23
Element to / department	23
Element to / nameperson	23
Element to / street	23

Element to / postcode	24
Element to / country	24
Element to / additionaladdressinfo	24
Element letter / text	24
Element letter / costscoveredby	25
Element costscoveredby / contractpartnerid	25
Element costscoveredby / ourcontractpartnerid	25
Element costscoveredby / maxcostscovered	25
Element actions / onprocessstart	26
Element actions / onprocessend	26
Element actions / onfullsuccess	27
Element actions / onerror	27
Element feed / bundle	28
Element bundle / displayname	29
Element bundle / name	29
Element bundle / version	29
Element bundle / display_artistname	30
Element bundle / ids	30
Element ids / grid	31
Element ids / upc	31
Element ids / isrc	31
Element ids / contentauth	31
Element ids / labelordernum	32
Element ids / amzn	32
Element ids / isbn	32
Element ids / finetunes	32
Element ids / licensor	33
Element ids / licensee	33
Element ids / gvl	33
Element bundle / items	33
Element items / item	34
Element item / displayname	35
Element item / name	35
Element item / version	35
Element item / type	35
Element item / display_artistname	36
Element item / ids	36
Element item / contributors	37
Element contributors / contributor	37
Element contributor / name	38
Element contributor / type	38
Element contributor / year	39
Element contributor / ids	39
Element contributor / www	40
Element www / facebook	40
Element www / myspace	41
Element www / homepage	41
Element www / twitter	41
Element www / phone	42
Element item / information	42
Element information / texts	43
Element texts / promotext	44
Element texts / teasertext	44
Element information / physical_release_datetime	44
Element information / digital_release_datetime	45
Element information / playlength	45
Element information / num	45
Element information / setnum	45
Element information / suggested_prelistening_offset	46
Element information / origin_country	46
Element information / main_language	51
Element information / related	54
Element related / physical_distributor	55
Element related / utube	55
Element utube / url	56
Element utube / channel	56
Element related / bundle	56
Element bundle / contributors	58
Element bundle / information	58
Element bundle / license_basis	59
Element license_basis / territorial	59
Element territorial / territory	60

Element license_basis / timeframe	60
Element timeframe / from	60
Element timeframe / to	61
Element license_basis / pricing	61
Element pricing / pricecode	61
Element pricing / wholesale	62
Element license_basis / streaming_allowed	62
Element license_basis / channels	62
Element channels / channel	62
Element bundle / license_specifics	63
Element license_specifics / rules	63
Element rules / rule	64
Element rule / if	64
Element if / what	65
Element if / operator	65
Element if / value	65
Element rule / then	65
Element then / proclaim	66
Element proclaim / what	66
Element proclaim / for	66
Element then / echo	66
Element then / break	67
Element rule / else	67
Element else / proclaim	67
Element else / break	68
Element bundle / reporting	68
Element reporting / realtime	68
Element realtime / http	69
Element reporting / postponed	69
Element postponed / id	69
Element bundle / tags	70
Element tags / genres	70
Element genres / genre	71
Element tags / bundle_only	88
Element tags / explicit_lyrics	88
Element tags / live	88
Element tags / accoustic	89
Element tags / instrumental	89
Element bundle / files	89
Element files / file	90
Element file / location	92
Element fileLocation / path	92
Element fileLocation / http	93
Element fileHttp / url	93
Element fileHttp / user	93
Element fileHttp / pass	94
Element fileHttp / expiresdatetime	94
Element fileLocation / ftp	94
Element fileFtp / server	95
Element fileFtp / port	95
Element fileFtp / path	95
Element fileFtp / user	95
Element fileFtp / pass	96
Element fileFtp / expiresdatetime	96
Element file / type	96
Element file / filetype	97
Element file / samplerate	97
Element file / prelistening_offset	97
Element file / prelistening_length	98
Element file / samplesize	98
Element file / bitrate	98
Element file / bitratetype	99
Element file / codec	99
Element file / codecsettings	99
Element file / bytes	100
Element file / checksums	100
Element checksums / md5	100
Element checksums / sha1	101
Element checksums / sha256	101
Element file / channels	101
Element file / dimension	101
Element dimension / width	102

Element dimension / height	102
Element file / decryptinfo	102
Element decryptinfo / cipher	103
Element decryptinfo / initvector	103
Element decryptinfo / key	103
Element decryptinfo / bytes	104
Element decryptinfo / checksums	104
Element file / no_file_given	104
Element file / comment	105
Element bundle / purchase	105
Element purchase / pos	105
Element purchase / url	105
Element item / license_basis	106
Element license_basis_item / territorial	106
Element license_basis_item / timeframe	106
Element license_basis_item / pricing	107
Element license_basis_item / streaming_allowed	107
Element license_basis_item / channels	108
Element license_basis_item / as_on_bundle	108
Element item / license_specifics	108
Element license_specifics_item / rules	109
Element license_specifics_item / as_on_bundle	109
Element item / tags	109
Element item / fingerprint	110
Element fingerprint / echoprint	110
Element item / reporting	110
Element item / files	111
Element feed / item	111
Complex Type(s)	113
Complex Type feedinfo	113
Complex Type creator	114
Complex Type receiver	114
Complex Type crypto	115
Complex Type sender	115
Complex Type licensor	116
Complex Type licensee	116
Complex Type actions	117
Complex Type event	117
Complex Type mailto	118
Complex Type action	119
Complex Type http	119
Complex Type http_addheader	119
Complex Type action_instruction	120
Complex Type http_addparams	120
Complex Type fax	120
Complex Type letter	121
Complex Type to	121
Complex Type costscoveredby	122
Complex Type bundle	122
Complex Type ids	124
Complex Type items	124
Complex Type item	125
Complex Type contributors	126
Complex Type contributor	126
Complex Type www	127
Complex Type publishable	127
Complex Type information	128
Complex Type texts	129
Complex Type promotext	129
Complex Type teasertext	130
Complex Type related	130
Complex Type physical_distributor	130
Complex Type utube	131
Complex Type license_basis	131
Complex Type territorial	132
Complex Type territory	132
Complex Type timeframe	133
Complex Type pricing	133
Complex Type channels	134
Complex Type channel	134
Complex Type license_specifics	134
Complex Type rules	135

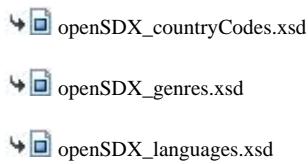
Complex Type rule	135
Complex Type if	136
Complex Type then	136
Complex Type proclaim	137
Complex Type else	137
Complex Type reporting	137
Complex Type realtime	138
Complex Type postponed	138
Complex Type tags	139
Complex Type genres	139
Complex Type files	139
Complex Type file	140
Complex Type fileLocation	143
Complex Type fileHttp	143
Complex Type fileFtp	144
Complex Type checksums	144
Complex Type dimension	145
Complex Type decryptinfo	145
Complex Type purchase	146
Complex Type license_basis_item	146
Complex Type license_specifics_item	147
Complex Type fingerprint	147
Complex Type oninitialreceive	147
Complex Type onprocessstart	148
Complex Type onprocessend	149
Complex Type onfullsuccess	149
Complex Type onerror	150
Simple Type(s)	150
Simple Type datetimeGMT	150
Simple Type email	151
Simple Type userid	151
Simple Type receivertypes	151
Simple Type iporhostname	151
Simple Type ipv4	152
Simple Type ipv6	152
Simple Type authtype	152
Simple Type keyid	153
Simple Type emaillist	153
Simple Type url	153
Simple Type httpmethods	153
Simple Type upc	154
Simple Type isrc	154
Simple Type finetunes	155
Simple Type contributorType	155
Simple Type allowance	156
Simple Type operator	156
Simple Type explicitLyrics	156
Simple Type fileType	157
Simple Type fileChannels	157
Simple Type isbn	158
Attribute(s)	158
Attribute publishable / @publishable	158
Attribute contributor / @num	158
Attribute promotext / @lang	158
Attribute teasertext / @lang	158
Attribute physical_distributor / @publishable	159
Attribute territory / @type	159
Attribute channel / @type	159
Attribute rule / @num	159
Namespace: "http://fnppl.org/opensdx/countrycodes"	159
Schema(s)	159
Imported schema openSDX_countryCodes.xsd	159
Simple Type(s)	160
Simple Type countryCode	160
Namespace: "http://fnppl.org/opensdx/genres"	180
Schema(s)	180
Imported schema openSDX_genres.xsd	180
Simple Type(s)	180
Simple Type genre	180
Namespace: "http://fnppl.org/opensdx/languages"	207
Schema(s)	207
Imported schema openSDX_languages.xsd	207

Simple Type(s)	207
Simple Type language	207

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" -- "0..infinity" feedinfo feed "0..infinity" -- "0..infinity" bundle feed "0..infinity" -- "0..infinity" 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
-----------	--------------

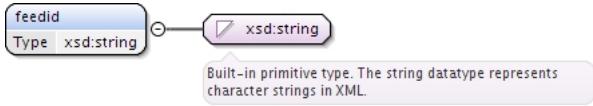
Diagram	<pre> classDiagram class feedinfo { onlytest feedid creationdatetime effectivedatetime creator receiver sender licensor licensee actions } feedinfo < -- feedinfo note over feedinfo: On feedinfo-level there are the global information needed or at least valuable for ingesting / identifying the content... </pre>
Type	feedinfo
Properties	content: complex
Model	ALL(onlytest feedid creationdatetime effectivedatetime creator{0,1} 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>{0,1}</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

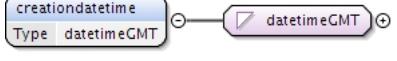
Namespace	No namespace
Diagram	<pre> classDiagram class onlytest { Type xsd:boolean } note over onlytest: Built-in primitive type. It defines the boolean values true and false. </pre>
Type	xsd:boolean
Properties	content: simple
Source	<code><xsd:element name="onlytest" type="xsd:boolean"/></code>

Element feedinfo / feedid

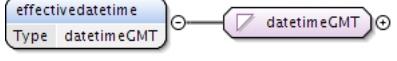
Namespace	No namespace
-----------	--------------

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

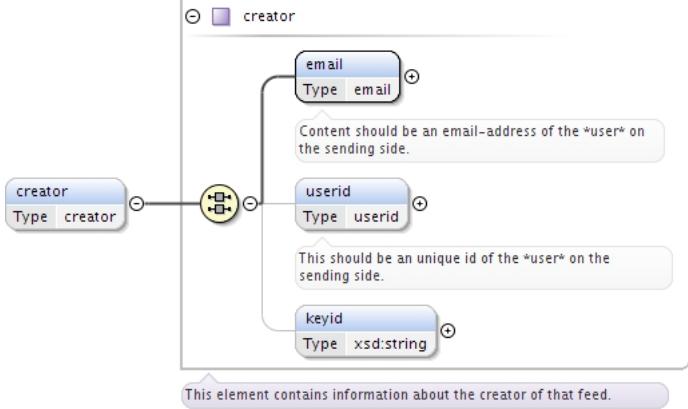
Element feedinfo / creationdatetime

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

Element feedinfo / effectivedatetime

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

Element feedinfo / creator

Namespace	No namespace
Diagram	
Type	creator
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(email userid{0,1} keyid{0,1})

Children	email, keyid, userid
Instance	<pre><creator> <email>{1,1}</email> <userid>{0,1}</userid> <keyid>{0,1}</keyid> </creator></pre>
Source	<pre><xsd:element name="creator" type="creator" maxOccurs="1" minOccurs="0" /></pre>

Element creator / email

Namespace	No namespace
Annotations	Content should be an email-address of the *user* on the sending side.
Diagram	<p>The diagram shows a node labeled "email" with a self-loop arrow, indicating it is a simple type. A callout box below the node states: "Content should be an email-address of the *user* on the sending side."</p>
Type	email
Properties	content: simple
Facets	pattern (([a-zA-Z0-9_\-\.\.]+@[a-zA-Z0-9-]+\.(\. [a-zA-Z0-9-]+)*(\.[a-zA-Z]{2,3}))?)
Source	<pre><xsd:element name="email" type="email"> <xsd:annotation> <xsd:documentation xml:lang="en">Content should be an email-address of the *user* on the sending side.</xsd:documentation> </xsd:annotation> </xsd:element></pre>

Element creator / userid

Namespace	No namespace						
Annotations	This should be an unique id of the *user* on the sending side.						
Diagram	<p>The diagram shows a node labeled "userid" with a self-loop arrow, indicating it is a simple type. A callout box below the node states: "This should be an unique id of the *user* on the sending side."</p>						
Type	userid						
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="userid" type="userid" maxOccurs="1" minOccurs="0" > <xsd:annotation> <xsd:documentation xml:lang="en">This should be an unique id of the *user* on the sending side.</xsd:documentation> </xsd:annotation> </xsd:element></pre>						

Element creator / keyid

Namespace	No namespace				
Diagram	<p>The diagram shows a node labeled "keyid" with a self-loop arrow, indicating it is a simple type. A callout box below the node states: "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> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				

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

Element feedinfo / receiver

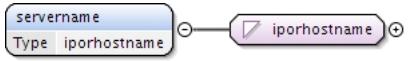
Namespace	No namespace
Diagram	<pre> classDiagram class receiver { type receivertypes servername iporhostname serveripv4 ipv4 serveripv6 ipv6 authtype authtype username xsd:string crypto crypto } receiver < -- receiver </pre> <p>This element contains information about the receiver of that feed.</p>
Type	receiver
Properties	content: complex
Model	ALL(type servername serveripv4 serveripv6{0,1} authtype username{0,1} crypto{0,1})
Children	authtype, crypto, serveripv4, serveripv6, servername, type, username
Instance	<pre> <receiver> <type>{1,1}</type> <servername>{1,1}</servername> <serveripv4>{1,1}</serveripv4> <serveripv6>{0,1}</serveripv6> <authtype>{1,1}</authtype> <username>{0,1}</username> <crypto>{0,1}</crypto> </receiver> </pre>
Source	<xsd:element name="receiver" type="receiver" />

Element receiver / type

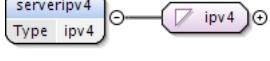
Namespace	No namespace										
Diagram	<pre> classDiagram class type { Type receivertypes } type --> receivertypes </pre>										
Type	receivertypes										
Properties	content: simple										
Facets	<table border="1"> <tr> <td>enumeration</td> <td>ftp</td> </tr> <tr> <td>enumeration</td> <td>ftps</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

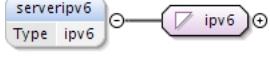
Namespace	No namespace
-----------	--------------

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

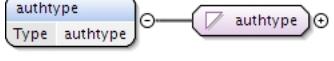
Element receiver / serveripv4

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

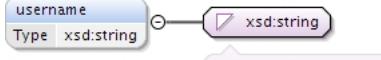
Element receiver / serveripv6

Namespace	No namespace
Diagram	
Type	ipv6
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="serveripv6" type="ipv6" maxOccurs="1" minOccurs="0" />

Element receiver / authtype

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

Element receiver / username

Namespace	No namespace
Diagram	 <small>Built-in primitive type. The string datatype represents character strings in XML.</small>
Type	xsd:string

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

Element receiver / crypto

Namespace	No namespace
Diagram	<p>This element contains crypto information for secure and authenticated transfer.</p>
Type	crypto
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(relatedemail{0,1} usedkeyid{0,1} usedpubkey{0,1})
Children	relatedemail, usedkeyid, usedpubkey
Instance	<crypto> <relatedemail>{0,1}</relatedemail> <usedkeyid>{0,1}</usedkeyid> <usedpubkey>{0,1}</usedpubkey> </crypto>
Source	<xsd:element name="crypto" type="crypto" maxOccurs="1" minOccurs="0" />

Element crypto / relatedemail

Namespace	No namespace
Diagram	
Type	email
Properties	content: simple minOccurs: 0 maxOccurs: 1
Facets	pattern (([a-zA-Z0-9_\-\.\.]+@[a-zA-Z0-9\-.]+\(\.[a-zA-Z0-9\-.]+\)*(\.\.[a-zA-Z]{2,3}\))?)?
Source	<xsd:element name="relatedemail" type="email" maxOccurs="1" minOccurs="0" />

Element crypto / usedkeyid

Namespace	No namespace
Diagram	
Type	keyid
Properties	content: simple minOccurs: 0

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

Element crypto / usedpubkey

Namespace	No namespace						
Diagram	<p>The diagram shows the <code>usedpubkey</code> element connected to the <code>xsd:base64Binary</code> type. A callout box indicates that <code>base64Binary</code> is a built-in primitive type representing Base64-encoded arbitrary binary data.</p>						
Type	xsd:base64Binary						
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="usedpubkey" type="xsd:base64Binary" maxOccurs="1" minOccurs="0" />						

Element feedinfo / sender

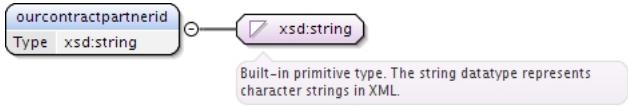
Namespace	No namespace
Diagram	<p>The diagram shows the <code>sender</code> element containing four child elements: <code>contractpartnerid</code>, <code>ourcontractpartnerid</code>, <code>email</code>, and <code>keyid</code>. A callout box states that this element contains information about the sender of that feed.</p>
Type	sender
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid email{0,1} keyid{0,1})
Children	contractpartnerid, email, keyid, ourcontractpartnerid
Instance	<pre><sender> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid> <email>{0,1}</email> <keyid>{0,1}</keyid> </sender></pre>
Source	<xsd:element name="sender" type="sender" />

Element sender / contractpartnerid

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

Element sender / ourcontractpartnerid

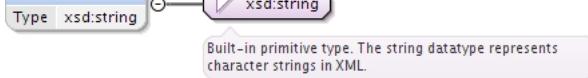
Namespace	No namespace
-----------	--------------

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

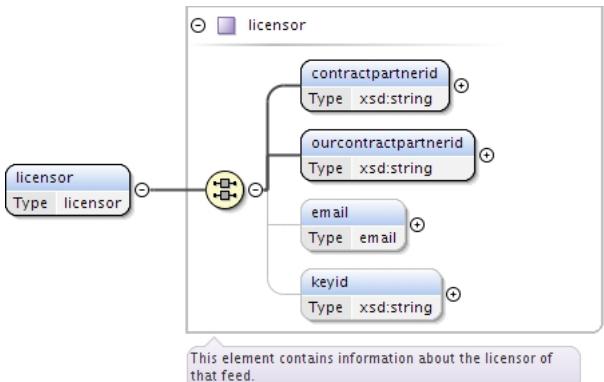
Element sender / email

Namespace	No namespace
Diagram	
Type	email
Properties	content: simple minOccurs: 0 maxOccurs: 1
Facets	pattern (([a-zA-Z0-9_\-\.\.]+@[a-zA-Z0-9\-.]+\(\.[a-zA-Z0-9\-.]+\)*(\.\.[a-zA-Z]{2,3}))?)
Source	<xsd:element name="email" type="email" maxOccurs="1" minOccurs="0"/>

Element sender / 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 / licensor

Namespace	No namespace
Diagram	
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	<code><xsd:element name="licensor" type="licensor" /></code>

Element licensor / contractpartnerid

Namespace	No namespace
Diagram	<p>The diagram shows the element 'contractpartnerid' represented by a blue rounded rectangle. To its right is a purple rounded rectangle containing the text 'xsd:string'. A line connects the two. Below this, a grey box contains the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="contractpartnerid" type="xsd:string" /></code>

Element licensor / ourcontractpartnerid

Namespace	No namespace
Diagram	<p>The diagram shows the element 'ourcontractpartnerid' represented by a blue rounded rectangle. To its right is a purple rounded rectangle containing the text 'xsd:string'. A line connects the two. Below this, a grey box contains the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="ourcontractpartnerid" type="xsd:string" /></code>

Element licensor / email

Namespace	No namespace
Diagram	<p>The diagram shows the element 'email' represented by a blue rounded rectangle. To its right is a purple rounded rectangle containing the text 'email'. A line connects the two. Below this, a grey box contains the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	email
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Facets	<p>pattern</p> <pre>(([a-zA-Z0-9_\- \.]*)@[a-zA-Z0-9-]+(\. [a-zA-Z0-9-]+)*(\.[a-zA-Z] {2,3}))?</pre>
Source	<code><xsd:element name="email" type="email" maxOccurs="1" minOccurs="0" /></code>

Element licensor / keyid

Namespace	No namespace
Diagram	<p>The diagram shows the element 'keyid' represented by a blue rounded rectangle. To its right is a purple rounded rectangle containing the text 'xsd:string'. A line connects the two. Below this, a grey box contains 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="keyid" type="xsd:string" maxOccurs="1" minOccurs="0"/>

Element feedinfo / licensee

Namespace	No namespace
Diagram	<pre> classDiagram class licensee { contractpartnerid : xsd:string ourcontractpartnerid : xsd:string email : email keyid : xsd:string } licensee < -- licensee </pre>
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

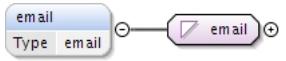
Namespace	No namespace
Diagram	<pre> classDiagram class contractpartnerid { <<Built-in primitive type. The string datatype represents character strings in XML.>> } contractpartnerid < -- xsd:string </pre>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="contractpartnerid" type="xsd:string"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

Element licensee / ourcontractpartnerid

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

Element licensee / email

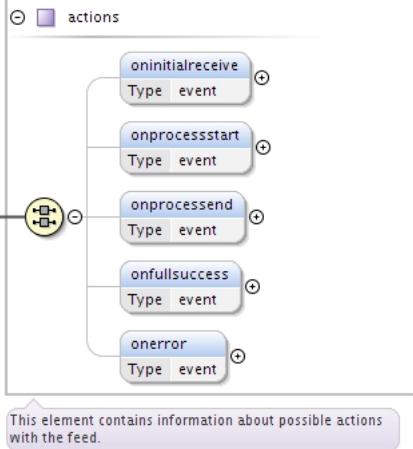
Namespace	No namespace
-----------	--------------

Diagram	
Type	email
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Facets	<p>pattern</p> <pre>(([a-zA-Z0-9_-] \.)+@[a-zA-Z0-9-]+(\. [a-zA-Z0-9-]+)*(\.[a-zA-Z0-9-]{2,3}))?</pre>
Source	<xsd:element name="email" type="email" maxOccurs="1" minOccurs="0"/>

Element licensee / keyid

Namespace	No namespace
Diagram	 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="keyid" type="xsd:string" maxOccurs="1" minOccurs="0"/>

Element feedinfo / actions

Namespace	No namespace
Diagram	 This element contains information about possible actions with the feed.
Type	actions
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	ALL(oninitialreceive{0,1} onprocessstart{0,1} onprocessend{0,1} onfullsuccess{0,1} onerror{0,1})
Children	onerror, onfullsuccess, oninitialreceive, onprocessend, onprocessstart
Instance	<pre><actions> <oninitialreceive>{0,1}</oninitialreceive> <onprocessstart>{0,1}</onprocessstart> <onprocessend>{0,1}</onprocessend> <onfullsuccess>{0,1}</onfullsuccess> <onerror>{0,1}</onerror> </actions></pre>

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

Element actions / oninitialreceive

Namespace	No namespace						
Diagram							
Type	event						
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	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	<code><xsd:element name="oninitialreceive" type="event" maxOccurs="1" minOccurs="0" /></code>						

Element event / mailto

Namespace	No namespace						
Diagram							
Type	mailto						
Type hierarchy	<ul style="list-style-type: none"> • action • mailto 						
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	receiver+, subject, text						
Children	receiver, subject, text						
Instance	<code><mailto></code>						

	<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	<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> </table>	content:	simple
content:	simple		
Source	<pre><xsd:element name="subject" type="xsd:string" /></pre>		

Element mailto / text

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> </table>	content:	simple
content:	simple		
Source	<pre><xsd:element name="text" type="xsd:string" /></pre>		

Element event / http

Namespace	No namespace
Diagram	<p>This element contains information about http-event.</p>

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 < -- action_instruction action_instruction < --> ##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 < -- action action < --> ##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 < -- letter letter < -- registered letter < -- to letter < -- text letter < -- costscoveredby </pre> <p>This element contains information about the letter event.</p>
Type	letter

Properties	content: complex minOccurs: 0 maxOccurs: unbounded
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
Annotations	This tells if letter must be registered or not.
Diagram	<p>The diagram shows a class named 'registered' with a note below it: 'This tells if letter must be registered or not.' A relationship arrow points from 'registered' to a 'xsd:boolean' class, with a note: 'Built-in primitive type. It defines the boolean values true and false.'</p>
Type	xsd:boolean
Properties	content: simple
Source	<pre><xsd:element name="registered" type="xsd:boolean"> <xsd:annotation> <xsd:documentation xml:lang="en">This tells if letter must be registered or not.</ xsd:documentation> </xsd:annotation> </xsd:element></pre>

Element letter / to

Namespace	No namespace
Diagram	<p>The diagram shows a class named 'to' with attributes: 'name' (xsd:string), 'department' (xsd:string), 'nameperson' (xsd:string), 'street' (xsd:string), 'postcode' (xsd:string), 'country' (xsd:string), and 'additionaladdressinfo' (xsd:string). A note below the class says: 'This element contains information about recipient.'</p>
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></pre>

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

Element to / department

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

Element to / nameperson

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

Element to / street

Namespace	No namespace
Diagram	
Type	xsd:string

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

Element to / postcode

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

Element to / country

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

Element to / additionaladdressinfo

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

Element letter / text

Namespace	No namespace
Annotations	This contains the content/text of letter.
Diagram	
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="text" type="xsd:string"> <xsd:annotation> <xsd:documentation xml:lang="en">This contains the content/text of letter.</xsd:documentation> </xsd:annotation> </xsd:element>

Element letter / costscoveredby

Namespace	No namespace
Diagram	<pre> classDiagram class costscoveredby { contractpartnerid ourcontractpartnerid maxcostscovered } costscoveredby < -- costscoveredby costscoveredby < -- costscoveredby costscoveredby < -- costscoveredby </pre> <p>This element contains information about who covered the costs of event.</p>
Type	costscoveredby
Properties	content: complex
Model	ALL(contractpartnerid ourcontractpartnerid maxcostscovered{0,1})
Children	contractpartnerid, maxcostscovered, ourcontractpartnerid
Instance	<costscoveredby> <contractpartnerid>{1,1}</contractpartnerid> <ourcontractpartnerid>{1,1}</ourcontractpartnerid> <maxcostscovered>{0,1}</maxcostscovered> </costscoveredby>
Source	<xsd:element name="costscoveredby" type="costscoveredby"/>

Element costscoveredby / contractpartnerid

Namespace	No namespace
Diagram	<pre> classDiagram class costscoveredby { contractpartnerid ... } costscoveredby < -- costscoveredby </pre> <p>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 costscoveredby / ourcontractpartnerid

Namespace	No namespace
Diagram	<pre> classDiagram class costscoveredby { ourcontractpartnerid ... } costscoveredby < -- costscoveredby </pre> <p>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 costscoveredby / maxcostscovered

Namespace	No namespace
Annotations	This contains the max amount that will be covered.
Diagram	<pre> classDiagram class costscoveredby { maxcostscovered ... } costscoveredby < -- costscoveredby </pre> <p>This contains the max amount that will be covered.</p> <p>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"> <xsd:annotation> <xsd:documentation xml:lang="en">This contains the max amount that will be covered.</ xsd:documentation> </xsd:annotation> </xsd:element></pre>

Element actions / onprocessstart

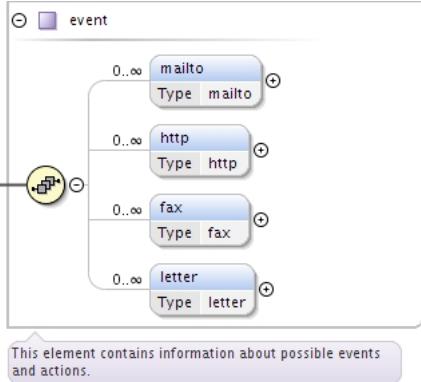
Namespace	No namespace
Diagram	<p>This element contains information about possible events and actions.</p>
Type	event
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Instance	<pre><onprocessstart> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </onprocessstart></pre>
Source	<pre><xsd:element name="onprocessstart" type="event" maxOccurs="1" minOccurs="0" /></pre>

Element actions / onprocessend

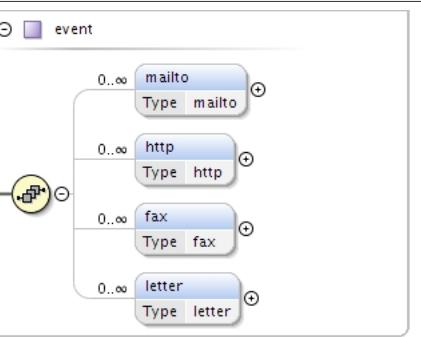
Namespace	No namespace
Diagram	<p>This element contains information about possible events and actions.</p>
Type	event
Properties	content: complex

	minOccurs: 0 maxOccurs: 1
Model	mailto*, http*, fax*, letter*
Children	fax, http, letter, mailto
Instance	<pre><onprocessend> <mailto>{0,unbounded}</mailto> <http>{0,unbounded}</http> <fax>{0,unbounded}</fax> <letter>{0,unbounded}</letter> </onprocessend></pre>
Source	<code><xsd:element name="onprocessend" type="event" maxOccurs="1" minOccurs="0" /></code>

Element actions / onfullsuccess

Namespace	No namespace
Diagram	 <p>This element contains information about possible events and actions.</p>
Type	event
Properties	content: complex minOccurs: 0 maxOccurs: 1
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" maxOccurs="1" minOccurs="0" /></code>

Element actions / onerror

Namespace	No namespace
Diagram	 <p>This element contains information about possible events and actions.</p>
Type	event

Properties	content: complex minOccurs: 0 maxOccurs: 1
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	<xsd:element name="onerror" type="event" maxOccurs="1" minOccurs="0"/>

Element feed / bundle

Namespace	No namespace
Diagram	<p>bundle</p> <p>displayname (xsd:string) name (xsd:string) version (xsd:string) display_artistname (xsd:string) ids (ids) items (items) contributors (contributors) information (information) license_basis (license_basis) license_specifics (license_specifics) reporting (reporting) tags (tags) files (files) purchase (purchase)</p> <p>On bundle level, there are information on how to handle a collection of "items". This is mainly an album/ep/single.</p>
Type	bundle
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ALL(displayname{0,1} name{0,1} version{0,1} display_artistname{0,1} ids items{0,1} contributors{0,1} information{0,1} license_basis{0,1} license_specifics{0,1} reporting{0,1} tags{0,1} files{0,1} purchase{0,1})

Children	contributors, display_artistname, displayname, files, ids, information, items, license_basis, license_specifics, name, purchase, reporting, tags, version
Instance	<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 / displayname

Namespace	No namespace						
Diagram	<p>The diagram shows the <code>displayname</code> element as a rounded rectangle with the label "displayname" above it and "Type xsd:string" below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A callout box below the connection line states: "Built-in primitive type. The string datatype represents character strings in XML."</p>						
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	<code><xsd:element name="displayname" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>						

Element bundle / name

Namespace	No namespace						
Diagram	<p>The diagram shows the <code>name</code> element as a rounded rectangle with the label "name" above it and "Type xsd:string" below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A callout box below the connection line states: "Built-in primitive type. The string datatype represents character strings in XML."</p>						
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	<code><xsd:element name="name" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>						

Element bundle / version

Namespace	No namespace						
Diagram	<p>The diagram shows the <code>version</code> element as a rounded rectangle with the label "version" above it and "Type xsd:string" below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A callout box below the connection line states: "Built-in primitive type. The string datatype represents character strings in XML."</p>						
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	<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	<pre><xsd:element name="display_artistname" type="xsd:string" maxOccurs="1" minOccurs="0" /></pre>

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></pre>

	<pre><licensor>{0,1}</licensor> <licensee>{0,1}</licensee> <gv1>{0,1}</gv1> </ids></pre>
Source	<code><xsd:element name="ids" type="ids" /></code>

Element ids / grid

Namespace	No namespace						
Diagram							
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="grid" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>						

Element ids / upc

Namespace	No namespace						
Diagram							
Type	upc						
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						
Facets	pattern <code>(\d{10,13})?</code>						
Source	<code><xsd:element name="upc" type="upc" maxOccurs="1" minOccurs="0" /></code>						

Element ids / isrc

Namespace	No namespace						
Diagram							
Type	isrc						
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						
Facets	pattern <code>([a-zA-Z]{2})(\-)? ([0-9a-zA-Z]{3})(\-)? \d{2}(\-)?\d{5})</code>						
Source	<code><xsd:element name="isrc" type="isrc" maxOccurs="1" minOccurs="0" /></code>						

Element ids / contentauth

Namespace	No namespace
Diagram	
Type	xsd:string

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

Element ids / labelordernum

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

Element ids / amzn

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

Element ids / isbn

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

Element ids / finetunes

Namespace	No namespace
Diagram	
Type	finetunes
Properties	content: simple minOccurs: 0

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

Element ids / licensor

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

Element ids / licensee

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

Element ids / gvl

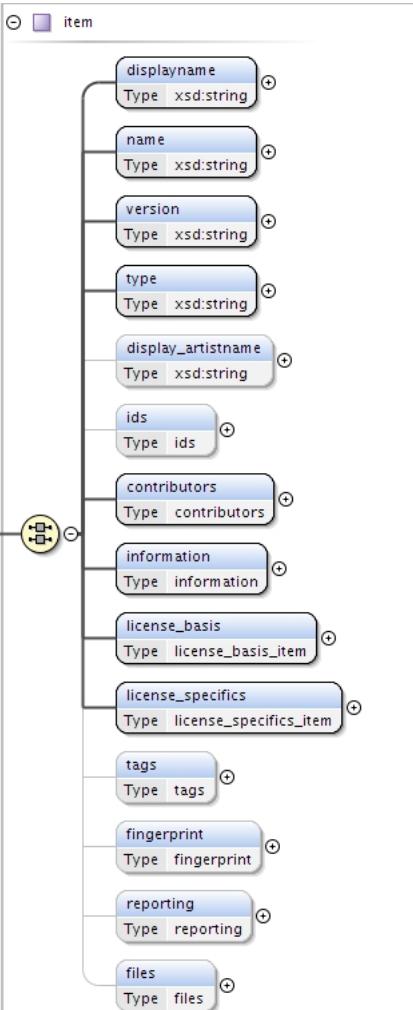
Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="gvl" type="xsd:string" maxOccurs="1" minOccurs="0" />

Element bundle / items

Namespace	No namespace
Diagram	
Type	items
Properties	content: complex

	minOccurs: 0
	maxOccurs: 1
Model	item+
Children	item
Instance	<items> <item>{1,unbounded}</item> </items>
Source	<xsd:element name="items" type="items" maxOccurs="1" minOccurs="0"/>

Element items / item

Namespace	No namespace						
Diagram	 <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>1</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	1	maxOccurs:	unbounded
content:	complex						
minOccurs:	1						
maxOccurs:	unbounded						
Model	ALL(displayname name version type display_artistname{0,1} ids{0,1} contributors information license_basis license_specifics tags{0,1} fingerprint{0,1} reporting{0,1} files{0,1})						
Children	contributors, display_artistname, displayname, files, fingerprint, ids, information, license_basis, license_specifics, name, reporting, tags, type, version						
Instance	<pre><item> <displayname>{1,1}</displayname> <name>{1,1}</name></pre>						

```

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

```

Source `<xsd:element name="item" type="item" maxOccurs="unbounded" minOccurs="1"/>`

Element item / displayname

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

Element item / name

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

Element item / version

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

Element item / type

Namespace	No namespace
Diagram	<p>The diagram shows a box labeled "type" with a line pointing to another box labeled "xsd:string". A callout bubble next to "xsd:string" states: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple

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

Element item / display_artistname

Namespace	No namespace						
Diagram	<p>The diagram shows the <code>display_artistname</code> element connected to the <code>xsd:string</code> type. A tooltip for <code>xsd:string</code> states: "Built-in primitive type. The string datatype represents 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="display_artistname" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>						

Element item / ids

Namespace	No namespace						
Diagram	<p>The diagram shows the <code>ids</code> element containing various ID types: <code>grid</code>, <code>upc</code>, <code>isrc</code>, <code>contentauth</code>, <code>labelordernum</code>, <code>amzn</code>, <code>isbn</code>, <code>finetunes</code>, <code>licensor</code>, <code>licensee</code>, and <code>gvl</code>. A tooltip for <code>ids</code> states: "This Element is a container for all IDs which are available for the associated element."</p>						
Type	<code>ids</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	<code>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})</code>						
Children	<code>amzn, contentauth, finetunes, grid, gvl, isbn, isrc, labelordernum, licensee, licensor, upc</code>						
Instance	<code><ids> <grid>{0,1}</grid> <upc>{0,1}</upc></code>						

	<pre><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	
Type	contributors
Properties	content: complex
Model	contributor*
Children	contributor
Instance	<pre><contributors> <contributor num="">{0,unbounded}</contributor> </contributors></pre>
Source	<xsd:element name="contributors" type="contributors"/>

Element contributors / contributor

Namespace	No namespace						
Diagram							
Type	contributor						
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(name type year{0,1} ids www{0,1})						
Children	ids, name, type, www, year						
Instance	<contributor num="">						

	<pre><name>{1,1}</name> <type>{1,1}</type> <year>{0,1}</year> <ids>{1,1}</ids> <www>{0,1}</www> </contributor></pre>				
Attributes	QName	Type	Fixed	Default	Use
	num	xsd:integer			optional
Source	<code><xsd:element name="contributor" type="contributor" maxOccurs="unbounded" minOccurs="0" /></code>				

Element contributor / name

Namespace	No namespace
Diagram	<p>The diagram shows a class named 'name' with a dependency arrow pointing to a class named 'xsd:string'. A callout bubble provides the definition: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="name" type="xsd:string" /></code>

Element contributor / type

Namespace	No namespace
Diagram	<p>The diagram shows a class named 'type' with a dependency arrow pointing to a class named 'contributorType'. A callout bubble provides the definition: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	contributorType
Properties	content: simple
Facets	enumeration label enumeration performer enumeration texter enumeration editor enumeration conductor enumeration orchestra enumeration display_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	<code><xsd:element name="type" type="contributorType" /></code>

Element contributor / year

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

Element contributor / ids

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

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

Element www / facebook

Namespace	No namespace										
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>										
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	<pre><xsd:element name="facebook" type="publishable" maxOccurs="1" minOccurs="0" /></pre>										

Element www / myspace

Namespace	No namespace										
Diagram	<pre> classDiagram class myspace { <<publishable>> <<xsd:string>> } class publishable { <<xsd:string>> <<attributes>> class publishable { <<xsd:boolean>> } } class xsdstring { <<Built-in primitive type. The string datatype represents character strings in XML.>> } class attributes { <<publishable>> } class xsdboolean { <<xsd:boolean>> } </pre>										
Type	publishable										
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										
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="myspace" type="publishable" maxOccurs="1" minOccurs="0" /></pre>										

Element www / homepage

Namespace	No namespace										
Diagram	<pre> classDiagram class homepage { <<publishable>> <<xsd:string>> } class publishable { <<xsd:string>> <<attributes>> class publishable { <<xsd:boolean>> } } class xsdstring { <<Built-in primitive type. The string datatype represents character strings in XML.>> } class attributes { <<publishable>> } class xsdboolean { <<xsd:boolean>> } </pre>										
Type	publishable										
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										
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="homepage" type="publishable" maxOccurs="1" minOccurs="0" /></pre>										

Element www / twitter

Namespace	No namespace
Diagram	<pre> classDiagram class twitter { <<publishable>> <<xsd:string>> } class publishable { <<xsd:string>> <<attributes>> class publishable { <<xsd:boolean>> } } class xsdstring { <<Built-in primitive type. The string datatype represents character strings in XML.>> } class attributes { <<publishable>> } class xsdboolean { <<xsd:boolean>> } </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="twitter" type="publishable" maxOccurs="1" minOccurs="0"/>														

Element www / phone

Namespace	No namespace										
Diagram	<pre> graph LR phone[phone Type: publishable] --> xsdString[xsd:string] xsdString --> publishableAttribute[publishable Type: xsd:boolean] </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 class information { texts physical_release_datetime digital_release_datetime playlength num setnum suggested_prelistening_offset origin_country main_language related } information < -- information </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 class texts { texts promotext teasertext } texts < -- texts </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	 Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...
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

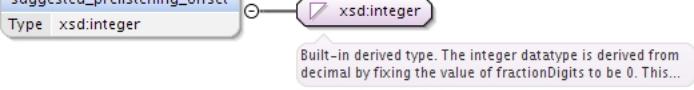
Namespace	No namespace
Diagram	 Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...
Type	xsd:integer
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="num" type="xsd:integer" maxOccurs="1" minOccurs="0"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

Element information / setnum

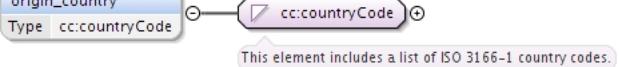
Namespace	No namespace
Diagram	 Built-in derived type. The integer datatype is derived from decimal by fixing the value of fractionDigits to be 0. This...
Type	xsd:integer

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' with a type of 'xsd:integer'. A tooltip indicates it is a built-in derived type derived from decimal by fixing fractionDigits to 0.</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' with a type of 'cc:countryCode'. A tooltip indicates it 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>AF</td><td>Afghanistan</td></tr> <tr><td>enumeration</td><td>AX</td><td>Aland Islands</td></tr> <tr><td>enumeration</td><td>AL</td><td>Albania</td></tr> <tr><td>enumeration</td><td>DZ</td><td>Algeria</td></tr> <tr><td>enumeration</td><td>AS</td><td>American Samoa</td></tr> <tr><td>enumeration</td><td>AD</td><td>Andorra</td></tr> <tr><td>enumeration</td><td>AO</td><td>Angola</td></tr> <tr><td>enumeration</td><td>AI</td><td>Anguilla</td></tr> <tr><td>enumeration</td><td>AQ</td><td>Antarctica</td></tr> <tr><td>enumeration</td><td>AG</td><td>Antigua and Barbuda</td></tr> <tr><td>enumeration</td><td>AR</td><td>Argentina</td></tr> <tr><td>enumeration</td><td>AM</td><td>Armenia</td></tr> <tr><td>enumeration</td><td>AW</td><td>Aruba</td></tr> <tr><td>enumeration</td><td>AU</td><td>Australia</td></tr> <tr><td>enumeration</td><td>AT</td><td>Austria</td></tr> <tr><td>enumeration</td><td>AZ</td><td>Azerbaijan</td></tr> <tr><td>enumeration</td><td>BS</td><td>Bahamas</td></tr> <tr><td>enumeration</td><td>BH</td><td>Bahrain</td></tr> <tr><td>enumeration</td><td>BD</td><td>Bangladesh</td></tr> <tr><td>enumeration</td><td>BB</td><td>Barbados</td></tr> <tr><td>enumeration</td><td>BY</td><td>Belarus</td></tr> <tr><td>enumeration</td><td>BE</td><td>Belgium</td></tr> <tr><td>enumeration</td><td>BZ</td><td>Belize</td></tr> </tbody> </table>	enumeration	AF	Afghanistan	enumeration	AX	Aland Islands	enumeration	AL	Albania	enumeration	DZ	Algeria	enumeration	AS	American Samoa	enumeration	AD	Andorra	enumeration	AO	Angola	enumeration	AI	Anguilla	enumeration	AQ	Antarctica	enumeration	AG	Antigua and Barbuda	enumeration	AR	Argentina	enumeration	AM	Armenia	enumeration	AW	Aruba	enumeration	AU	Australia	enumeration	AT	Austria	enumeration	AZ	Azerbaijan	enumeration	BS	Bahamas	enumeration	BH	Bahrain	enumeration	BD	Bangladesh	enumeration	BB	Barbados	enumeration	BY	Belarus	enumeration	BE	Belgium	enumeration	BZ	Belize
enumeration	AF	Afghanistan																																																																				
enumeration	AX	Aland Islands																																																																				
enumeration	AL	Albania																																																																				
enumeration	DZ	Algeria																																																																				
enumeration	AS	American Samoa																																																																				
enumeration	AD	Andorra																																																																				
enumeration	AO	Angola																																																																				
enumeration	AI	Anguilla																																																																				
enumeration	AQ	Antarctica																																																																				
enumeration	AG	Antigua and Barbuda																																																																				
enumeration	AR	Argentina																																																																				
enumeration	AM	Armenia																																																																				
enumeration	AW	Aruba																																																																				
enumeration	AU	Australia																																																																				
enumeration	AT	Austria																																																																				
enumeration	AZ	Azerbaijan																																																																				
enumeration	BS	Bahamas																																																																				
enumeration	BH	Bahrain																																																																				
enumeration	BD	Bangladesh																																																																				
enumeration	BB	Barbados																																																																				
enumeration	BY	Belarus																																																																				
enumeration	BE	Belgium																																																																				
enumeration	BZ	Belize																																																																				

enumeration	BJ	Benin
enumeration	BM	Bermuda
enumeration	BT	Bhutan
enumeration	BO	Bolivia, Plurinational State of
enumeration	BQ	Bonaire, Sint Eustatius and Saba
enumeration	BA	Bosnia and Herzegovina
enumeration	BW	Botswana
enumeration	BV	Bouvet Island
enumeration	BR	Brazil
enumeration	IO	British Indian Ocean Territory
enumeration	BN	Brunei Darussalam
enumeration	BG	Bulgaria
enumeration	BF	Burkina Faso
enumeration	BI	Burundi
enumeration	KH	Cambodia
enumeration	CM	Cameroon
enumeration	CA	Canada
enumeration	CV	Cape Verde
enumeration	KY	Cayman Islands
enumeration	CF	Central African Republic
enumeration	TD	Chad
enumeration	CL	Chile
enumeration	CN	China
enumeration	CX	Christmas Island
enumeration	CC	Cocos (Keeling) Islands
enumeration	CO	Colombia
enumeration	KM	Comoros
enumeration	CG	Congo
enumeration	CD	Congo, the Democratic Republic of the
enumeration	CK	Cook Islands
enumeration	CR	Costa Rica
enumeration	CI	Côte d'Ivoire
enumeration	HR	Croatia
enumeration	CU	Cuba
enumeration	CW	Curaçao
enumeration	CY	Cyprus
enumeration	CZ	Czech Republic
enumeration	DK	Denmark
enumeration	DJ	Djibouti
enumeration	DM	Dominica
enumeration	DO	Dominican Republic
enumeration	EC	Ecuador
enumeration	EG	Egypt
enumeration	SV	El Salvador
enumeration	GQ	Equatorial Guinea
enumeration	ER	Eritrea
enumeration	EE	Estonia
enumeration	ET	Ethiopia
enumeration	FK	Falkland Islands (Malvinas)

enumeration	FO	Faroe Islands
enumeration	FJ	Fiji
enumeration	FI	Finland
enumeration	FR	France
enumeration	GF	French Guiana
enumeration	PF	French Polynesia
enumeration	TF	French Southern Territories
enumeration	GA	Gabon
enumeration	GM	Gambia
enumeration	GE	Georgia
enumeration	DE	Germany
enumeration	GH	Ghana
enumeration	GI	Gibraltar
enumeration	GR	Greece
enumeration	GL	Greenland
enumeration	GD	Grenada
enumeration	GP	Guadeloupe
enumeration	GU	Guam
enumeration	GT	Guatemala
enumeration	GG	Guernsey
enumeration	GN	Guinea
enumeration	GW	Guinea-Bissau
enumeration	GY	Guyana
enumeration	HT	Haiti
enumeration	HM	Heard Island and McDonald Islands
enumeration	VA	Holy See (Vatican City State)
enumeration	HN	Honduras
enumeration	HK	Hong Kong
enumeration	HU	Hungary
enumeration	IS	Iceland
enumeration	IN	India
enumeration	ID	Indonesia
enumeration	IR	Iran, Islamic Republic of
enumeration	IQ	Iraq
enumeration	IE	Ireland
enumeration	IM	Isle of Man
enumeration	IL	Israel
enumeration	IT	Italy
enumeration	JM	Jamaica
enumeration	JP	Japan
enumeration	JE	Jersey
enumeration	JO	Jordan
enumeration	KZ	Kazakhstan
enumeration	KE	Kenya
enumeration	KI	Kiribati
enumeration	KP	Korea, Democratic People's Republic of
enumeration	KR	Korea, Republic of
enumeration	KW	Kuwait
enumeration	KG	Kyrgyzstan

enumeration	LA	Lao People's Democratic Republic
enumeration	LV	Latvia
enumeration	LB	Lebanon
enumeration	LS	Lesotho
enumeration	LR	Liberia
enumeration	LY	Libyan Arab Jamahiriya
enumeration	LI	Liechtenstein
enumeration	LT	Lithuania
enumeration	LU	Luxembourg
enumeration	MO	Macao
enumeration	MK	Macedonia, the former Yugoslav Republic of
enumeration	MG	Madagascar
enumeration	MW	Malawi
enumeration	MY	Malaysia
enumeration	MV	Maldives
enumeration	ML	Mali
enumeration	MT	Malta
enumeration	MH	Marshall Islands
enumeration	MQ	Martinique
enumeration	MR	Mauritania
enumeration	MU	Mauritius
enumeration	YT	Mayotte
enumeration	MX	Mexico
enumeration	FM	Micronesia, Federated States of
enumeration	MD	Moldova, Republic of
enumeration	MC	Monaco
enumeration	MN	Mongolia
enumeration	ME	Montenegro
enumeration	MS	Montserrat
enumeration	MA	Morocco
enumeration	MZ	Mozambique
enumeration	MM	Myanmar
enumeration	NA	Namibia
enumeration	NR	Nauru
enumeration	NP	Nepal
enumeration	NL	Netherlands
enumeration	NC	New Caledonia
enumeration	NZ	New Zealand
enumeration	NI	Nicaragua
enumeration	NE	Niger
enumeration	NG	Nigeria
enumeration	NU	Niue
enumeration	NF	Norfolk Island
enumeration	MP	Northern Mariana Islands
enumeration	NO	Norway
enumeration	OM	Oman
enumeration	PK	Pakistan
enumeration	PW	Palau
enumeration	PS	Palestinian Territory, Occupied

enumeration	PA	Panama
enumeration	PG	Papua New Guinea
enumeration	PY	Paraguay
enumeration	PE	Peru
enumeration	PH	Philippines
enumeration	PN	Pitcairn
enumeration	PL	Poland
enumeration	PT	Portugal
enumeration	PR	Puerto Rico
enumeration	QA	Qatar
enumeration	RE	Réunion
enumeration	RO	Romania
enumeration	RU	Russian Federation
enumeration	RW	Rwanda
enumeration	BL	Saint Barthélemy
enumeration	SH	Saint Helena, Ascension and Tristan da Cunha
enumeration	KN	Saint Kitts and Nevis
enumeration	LC	Saint Lucia
enumeration	MF	Saint Martin (French part)
enumeration	PM	Saint Pierre and Miquelon
enumeration	VC	Saint Vincent and the Grenadines
enumeration	WS	Samoa
enumeration	SM	San Marino
enumeration	ST	Sao Tome and Principe
enumeration	SA	Saudi Arabia
enumeration	SN	Senegal
enumeration	RS	Serbia
enumeration	SC	Seychelles
enumeration	SL	Sierra Leone
enumeration	SG	Singapore
enumeration	SX	Sint Maarten (Dutch part)
enumeration	SK	Slovakia
enumeration	SI	Slovenia
enumeration	SB	Solomon Islands
enumeration	SO	Somalia
enumeration	ZA	South Africa
enumeration	GS	South Georgia and the South Sandwich Islands
enumeration	SS	South Sudan
enumeration	ES	Spain
enumeration	LK	Sri Lanka
enumeration	SD	Sudan
enumeration	SR	Suriname
enumeration	SJ	Svalbard and Jan Mayen
enumeration	SZ	Swaziland
enumeration	SE	Sweden
enumeration	CH	Switzerland
enumeration	SY	Syrian Arab Republic
enumeration	TW	Taiwan, Province of China
enumeration	TJ	Tajikistan

enumeration	TZ	Tanzania, United Republic of
enumeration	TH	Thailand
enumeration	TL	Timor-Leste
enumeration	TG	Togo
enumeration	TK	Tokelau
enumeration	TO	Tonga
enumeration	TT	Trinidad and Tobago
enumeration	TN	Tunisia
enumeration	TR	Turkey
enumeration	TM	Turkmenistan
enumeration	TC	Turks and Caicos Islands
enumeration	TV	Tuvalu
enumeration	UG	Uganda
enumeration	UA	Ukraine
enumeration	AE	United Arab Emirates
enumeration	GB	United Kingdom
enumeration	US	United States
enumeration	UM	United States Minor Outlying Islands
enumeration	UY	Uruguay
enumeration	UZ	Uzbekistan
enumeration	VU	Vanuatu
enumeration	VE	Venezuela, Bolivarian Republic of
enumeration	VN	Viet Nam
enumeration	VG	Virgin Islands, British
enumeration	VI	Virgin Islands, U.S.
enumeration	WF	Wallis and Futuna
enumeration	WW	WorldWide
enumeration	EH	Western Sahara
enumeration	YE	Yemen
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	<pre> classDiagram class main_language { <<Type l:language>> } main_language "0..1" -- "1" l:language l:language <<This element includes a list of ISO 639-1 language codes.>> </pre>	
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

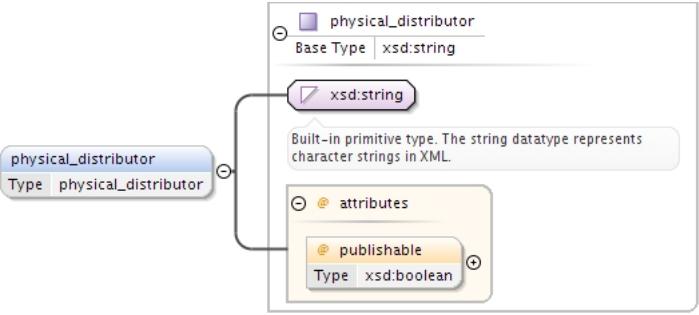
enumeration	so	Somali
enumeration	sq	Albanian
enumeration	sr	Serbian
enumeration	ss	Siswati
enumeration	st	Sesotho
enumeration	su	Sundanese
enumeration	sv	Swedish
enumeration	sw	Swahili
enumeration	ta	Tamil
enumeration	te	Telugu
enumeration	tg	Tajik
enumeration	th	Thai
enumeration	ti	Tigrinya
enumeration	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	<pre> classDiagram class related { <<related>> <<Type related>> } class physical_distributor { <<physical_distributor>> <<Type physical_distributor>> } class utube { <<utube>> <<Type utube>> } class bundle { <<bundle>> <<Type bundle>> } related "0..∞" --> physical_distributor : physical_distributor related "0..∞" --> utube : utube related "0..∞" --> bundle : bundle </pre> <p>This element contains informations of bundles which are related to the bundle of the actual feed. It may includes one...</p>
Type	related
Properties	content: complex

	minOccurs: 0 maxOccurs: 1
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	<code><xsd:element name="related" type="related" maxOccurs="1" minOccurs="0" /></code>

Element related / physical_distributor

Namespace	No namespace										
Diagram											
Type	physical_distributor										
Properties	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	<code><xsd:element name="physical_distributor" type="physical_distributor" maxOccurs="unbounded" minOccurs="0" /></code>										

Element related / utube

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

Element utube / url

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

Element utube / channel

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

Element related / 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 / contributors

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

Element bundle / information

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

Instance	<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 } territorial < -- license_basis channels < -- pricing channels < -- streaming_allowed channels < -- territorial channels < -- timeframe </pre> <p>This element includes the basic rules and information under which this bundle is provided. The optional element...</p>						
Type	license_basis						
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(territorial{0,1} timeframe{0,1} pricing{0,1} streaming_allowed{0,1} channels{0,1})						
Children	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> </license_basis></pre>						
Source	<pre><xsd:element name="license_basis" type="license_basis" maxOccurs="1" minOccurs="0"/></pre>						

Element license_basis / territorial

Namespace	No namespace						
Diagram	<pre> classDiagram class territorial { <<This Element is a container for territories. There should be a entry for all territories (ISO 3166-1 country code) with...>> territories } territory < -- territorial </pre>						
Type	territorial						
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	territory*
Children	territory
Instance	<pre><territorial> <territory type="">{0,unbounded}</territory> </territorial></pre>
Source	<code><xsd:element name="territorial" type="territorial" maxOccurs="1" minOccurs="0"/></code>

Element territorial / territory

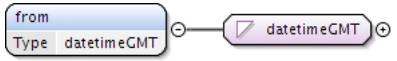
Namespace	No namespace										
Diagram											
Type	territory										
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded				
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>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	<code><xsd:element name="territory" type="territorial" maxOccurs="unbounded" minOccurs="0"/></code>										

Element license_basis / timeframe

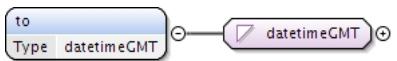
Namespace	No namespace						
Diagram							
Type	timeframe						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	from , to						
Children	from, to						
Instance	<pre><timeframe> <from>{1,1}</from> <to>{1,1}</to> </timeframe></pre>						
Source	<code><xsd:element name="timeframe" type="timeframe" maxOccurs="1" minOccurs="0"/></code>						

Element timeframe / from

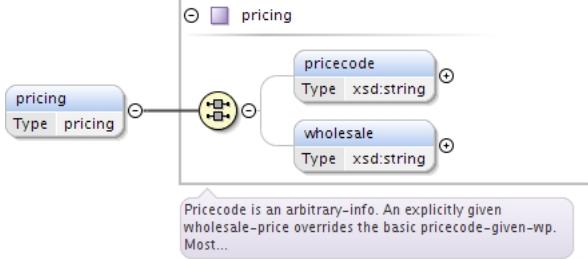
Namespace	No namespace
-----------	--------------

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="from" type="datetimeGMT" />

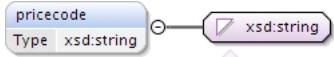
Element timeframe / to

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="to" type="datetimeGMT" />

Element license_basis / pricing

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

Element pricing / pricecode

Namespace	No namespace
Diagram	 A note states: '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="pricecode" type="xsd:string" maxOccurs="1" minOccurs="0" />

Element pricing / wholesale

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

Element license_basis / streaming_allowed

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

Element license_basis / channels

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

Element channels / channel

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

Diagram	<pre> classDiagram channel < -- xsd:string xsd:string --> note: Built-in primitive type. The string datatype represents character strings in XML. channel --> attribute: @type attribute --> allowance note: A channels can be either "all", "ad supported", "premium" or "ringtones". The required attribute "type" regards to the... </pre>										
Type	channel										
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										
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">QName</th><th style="text-align: left; padding: 2px;">Type</th><th style="text-align: left; padding: 2px;">Fixed</th><th style="text-align: left; padding: 2px;">Default</th><th style="text-align: left; padding: 2px;">Use</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;">type</td><td style="padding: 2px;">allowance</td><td style="padding: 2px;"></td><td style="padding: 2px;"></td><td style="padding: 2px;">required</td></tr> </tbody> </table>	QName	Type	Fixed	Default	Use	type	allowance			required
QName	Type	Fixed	Default	Use							
type	allowance			required							
Source	<pre><xsd:element name="channel" type="channel" maxOccurs="unbounded" minOccurs="0" /></pre>										

Element bundle / license_specifics

Namespace	No namespace						
Diagram	<pre> classDiagram license_specifics < -- rules note: This element includes specific rules which should be applied. </pre>						
Type	license_specifics						
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(rules{0,1})						
Children	rules						
Instance	<pre><license_specifics> <rules>{0,1}</rules> </license_specifics></pre>						
Source	<pre><xsd:element name="license_specifics" type="license_specifics" maxOccurs="1" minOccurs="0" /></pre>						

Element license_specifics / rules

Namespace	No namespace						
Diagram	<pre> classDiagram rules < -- rule note: This element is a container for rules. It needs an ordered mode here - first come first match. </pre>						
Type	rules						
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	rule*						

Children	rule
Instance	<pre><rules> <rule num="">{0,unbounded}</rule> </rules></pre>
Source	<pre><xsd:element name="rules" type="rules" maxOccurs="1" minOccurs="0" /></pre>

Element rules / rule

Namespace	No namespace										
Diagram	<p>A rule must include a "if"-element and a "then"-element to shape a legal instruction. It can also include a...</p>										
Type	rule										
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	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	<pre><xsd:element name="rule" type="rule" maxOccurs="unbounded" minOccurs="0" /></pre>										

Element rule / if

Namespace	No namespace		
Diagram	<p>This element must be the first element in a rule. It includes the information what is affected by the rule, an operator...</p>		
Type	if		
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
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 circle to its left, indicating it's a complex type. A line connects it to a purple rounded rectangle labeled "xsd:string" with a small circle to its left, indicating it's a primitive type. A callout bubble points to the "xsd:string" box with the text: "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 if / operator

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "operator" with a small circle to its left, indicating it's a complex type. A line connects it to another blue rounded rectangle labeled "operator" with a small circle to its left, indicating it's a primitive type. A callout bubble points to the second "operator" box with the text: "This element must be the second in a rule and includes information "echo" for debugging output and can include an..."</p>
Type	operator
Properties	content: simple
Facets	enumeration equals enumeration before enumeration after enumeration contains enumeration containedin
Source	<code><xsd:element name="operator" type="operator"/></code>

Element if / value

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "value" with a small circle to its left, indicating it's a complex type. A line connects it to a purple rounded rectangle labeled "xsd:string" with a small circle to its left, indicating it's a primitive type. A callout bubble points to the "xsd:string" box with the text: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="value" type="xsd:string"/></code>

Element rule / then

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "then" with a small circle to its left, indicating it's a complex type. A line connects it to another blue rounded rectangle labeled "then" with a small circle to its left, indicating it's a primitive type. This primitive "then" box is connected to a central yellow rounded rectangle with three outgoing lines. One line leads to a blue rounded rectangle labeled "proclaim" with a small circle to its left, indicating it's a complex type. Another line leads to a blue rounded rectangle labeled "echo" with a small circle to its left, indicating it's a complex type. The third line leads to a blue rounded rectangle labeled "break" with a small circle to its left, indicating it's a primitive type. A callout bubble points to the "then" box with the text: "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	proclaim*, echo{0,1}, break{0,1}
Children	break, echo, proclaim
Instance	<then> <proclaim>{0,unbounded}</proclaim> <echo>{0,1}</echo> <break>{0,1}</break> </then>
Source	<xsd:element name="then" type="then"/>

Element then / proclaim

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

Element proclaim / what

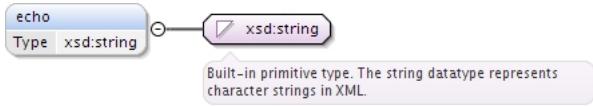
Namespace	No namespace
Diagram	<pre> classDiagram class what { <> xsd:string } what "0..1" -- "1..1" xsd:string </pre>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="what" type="xsd:string"/>

Element proclaim / for

Namespace	No namespace
Diagram	<pre> classDiagram class for { <> xsd:string } for "0..1" -- "1..1" xsd:string </pre>
Type	xsd:string
Properties	content: simple
Source	<xsd:element name="for" type="xsd:string"/>

Element then / echo

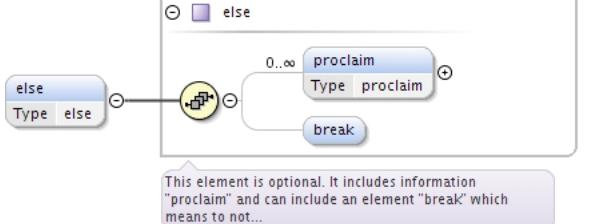
Namespace	No namespace
-----------	--------------

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

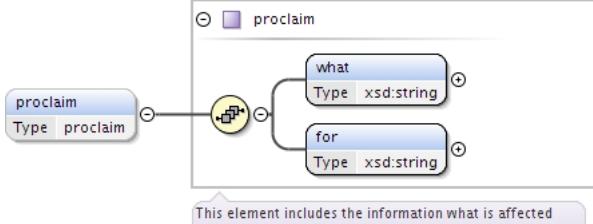
Element then / break

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

Element rule / else

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

Element else / proclaim

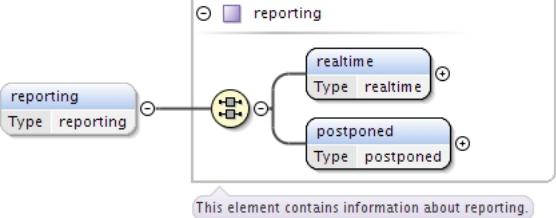
Namespace	No namespace
Diagram	
Type	proclaim
Properties	<p>content: complex</p> <p>minOccurs: 0</p>

	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>

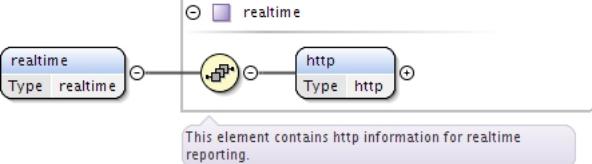
Element else / break

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	
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	
Type	postponed
Properties	content: complex
Model	id
Children	id
Instance	<pre><postponed> <id>{1,1}</id> </postponed></pre>
Source	<code><xsd:element name="postponed" type="postponed" /></code>

Element postponed / id

Namespace	No namespace
Diagram	

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

Element bundle / tags

Namespace	No namespace
Diagram	<pre> classDiagram class tags { genres bundle_only explicit_lyrics live acoustic instrumental } tags < -- bundle note over tags: This element contains information about genres and more. </pre>
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> <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	<xsd:element name="tags" type="tags" maxOccurs="1" minOccurs="0"/>

Element tags / genres

Namespace	No namespace
Diagram	<pre> classDiagram class genres { <-->[0..>] genre } note over genres: This element contains a list of genres. </pre>
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

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

Element genres / genre

Namespace	No namespace																																																																												
Diagram	<p>This element includes a list of openSDX-genres.</p>																																																																												
Type	genre																																																																												
Properties	<table> <tr> <td>content:</td><td>simple</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> <tr> <td>maxOccurs:</td><td>unbounded</td></tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded																																																																						
content:	simple																																																																												
minOccurs:	0																																																																												
maxOccurs:	unbounded																																																																												
Facets	<table> <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>Deutschrock</td></tr> <tr><td>enumeration</td><td>Emo</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 Rock</td></tr> <tr><td>enumeration</td><td>Psychedelic Rock</td></tr> <tr><td>enumeration</td><td>Psychobilly Rock</td></tr> <tr><td>enumeration</td><td>Rockabilly</td></tr> <tr><td>enumeration</td><td>Soft Rock</td></tr> <tr><td>enumeration</td><td>Southern Rock</td></tr> <tr><td>enumeration</td><td>Surf Rock</td></tr> <tr><td>enumeration</td><td>Alternative</td></tr> <tr><td>enumeration</td><td>Crossover</td></tr> <tr><td>enumeration</td><td>Dark Wave</td></tr> <tr><td>enumeration</td><td>Garage Rock</td></tr> <tr><td>enumeration</td><td>Goth / Industrial</td></tr> <tr><td>enumeration</td><td>Grunge</td></tr> <tr><td>enumeration</td><td>Hardcore</td></tr> <tr><td>enumeration</td><td>Indie Rock</td></tr> <tr><td>enumeration</td><td>New Wave</td></tr> <tr><td>enumeration</td><td>Punk</td></tr> <tr><td>enumeration</td><td>Funkpunk</td></tr> <tr><td>enumeration</td><td>Black Metal</td></tr> <tr><td>enumeration</td><td>Death Metal</td></tr> <tr><td>enumeration</td><td>Heavy Metal</td></tr> <tr><td>enumeration</td><td>Power Metal</td></tr> <tr><td>enumeration</td><td>Thrash / Speed Metal</td></tr> <tr><td>enumeration</td><td>Doom Metal</td></tr> <tr><td>enumeration</td><td>Grind Core</td></tr> <tr><td>enumeration</td><td>Pop</td></tr> </table>	enumeration	Rock	enumeration	Beat	enumeration	Blues Rock	enumeration	Rock "n" Roll	enumeration	Art Rock	enumeration	Classic Rock	enumeration	Deutschrock	enumeration	Emo	enumeration	Experimental Rock	enumeration	Glam Rock	enumeration	Hard Rock	enumeration	Krautrock	enumeration	Progressive Rock	enumeration	Psychedelic Rock	enumeration	Psychobilly Rock	enumeration	Rockabilly	enumeration	Soft Rock	enumeration	Southern Rock	enumeration	Surf Rock	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	Funkpunk	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	Rock																																																																												
enumeration	Beat																																																																												
enumeration	Blues Rock																																																																												
enumeration	Rock "n" Roll																																																																												
enumeration	Art Rock																																																																												
enumeration	Classic Rock																																																																												
enumeration	Deutschrock																																																																												
enumeration	Emo																																																																												
enumeration	Experimental Rock																																																																												
enumeration	Glam Rock																																																																												
enumeration	Hard Rock																																																																												
enumeration	Krautrock																																																																												
enumeration	Progressive Rock																																																																												
enumeration	Psychedelic Rock																																																																												
enumeration	Psychobilly Rock																																																																												
enumeration	Rockabilly																																																																												
enumeration	Soft Rock																																																																												
enumeration	Southern Rock																																																																												
enumeration	Surf Rock																																																																												
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	Funkpunk																																																																												
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	Britpop
enumeration	Dance Pop
enumeration	Deutschpop
enumeration	Disco
enumeration	Easy Listening
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	New Age
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	German Folk / Volksmusik
enumeration	Jazz
enumeration	Acid Jazz
enumeration	Avantgarde
enumeration	Bebop
enumeration	Big Band
enumeration	Classic Jazz
enumeration	Cool Jazz
enumeration	Dixieland music
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	Hip Hop
enumeration	Alternative Hip Hop
enumeration	Crunk
enumeration	Dirty South
enumeration	G-Funk
enumeration	Gangsta Rap
enumeration	Golden Era
enumeration	Grime
enumeration	Hyphy
enumeration	Instrumental Hip Hop
enumeration	Miami Bass
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	Doo-wop
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	Glitch hop
enumeration	House
enumeration	Acid House
enumeration	Deep House
enumeration	Disco House
enumeration	Electro House
enumeration	Fidget House
enumeration	Hard House
enumeration	Progressive House
enumeration	Soulful House
enumeration	Tech House
enumeration	Tribal
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	Ancient music
enumeration	Medieval music
enumeration	Renaissance
enumeration	Baroque
enumeration	Classical period
enumeration	Romantic
enumeration	Neoromanticism
enumeration	Neoclassicism
enumeration	New Music / Contemporary Music
enumeration	Modern, 20th / 21st Century
enumeration	Postmodern Music
enumeration	Music and other Media / Arts
enumeration	Music and Word
enumeration	12-Tone Composition
enumeration	Anthem
enumeration	Ballet
enumeration	Cantata
enumeration	Chamber Music
enumeration	Choral

enumeration	Crossover / Popular Classicism
enumeration	Electronic Music / Computer Music
enumeration	Madrigal
enumeration	March
enumeration	Minimal Music
enumeration	Motet
enumeration	Musical
enumeration	Opera Arias
enumeration	Opera Baroque
enumeration	Opera Classical
enumeration	Opera Renaissance
enumeration	Opera Romantic
enumeration	Operetta
enumeration	Oratorio
enumeration	Passion
enumeration	Requiem
enumeration	Serialism
enumeration	Sonata
enumeration	Suite
enumeration	Symphonic Music / Orchestral Music
enumeration	Symphony
enumeration	Waltz
enumeration	Brass Ensemble
enumeration	Concerto / Solo Instrument with Orchestra
enumeration	Mixed Ensemble (Strings / Wind)
enumeration	Mixed Wind Ensemble (Woodwind / Brass)
enumeration	Several Solo Instruments
enumeration	Solo Instrument
enumeration	String Ensemble
enumeration	String Orchestra
enumeration	String Quartet
enumeration	String Trio
enumeration	Woodwind Ensemble
enumeration	A cappella
enumeration	Vocal Ensemble
enumeration	Vocal Music
enumeration	Choir
enumeration	Boy's Choir
enumeration	Children's Choir
enumeration	Choir with Orchestra
enumeration	Women's Choir
enumeration	Men's Choir
enumeration	Mixed Choir
enumeration	Soprano

enumeration	Mezzosoprano
enumeration	Alto
enumeration	Tenor
enumeration	Baritone
enumeration	Bass
enumeration	Accordion
enumeration	Ancient Instruments
enumeration	Bassoon
enumeration	Cembalo
enumeration	Clarinet
enumeration	Double Bass
enumeration	Flute
enumeration	Guitar
enumeration	Harp
enumeration	Harpsichord
enumeration	Horn
enumeration	Lute
enumeration	Mandolin
enumeration	Oboe
enumeration	Organ
enumeration	Percussion (Vibraphone etc.)
enumeration	Piano
enumeration	Recorder / English Flute
enumeration	Saxophone
enumeration	Trombone
enumeration	Trumpet
enumeration	Tuba
enumeration	Viola
enumeration	Violin
enumeration	Violoncello
enumeration	Miscellaneous Lead Instrument
enumeration	Reggae
enumeration	Contemporary Reggae
enumeration	Dancehall
enumeration	Dub
enumeration	Lover's Rock
enumeration	Reggaeton
enumeration	Roots
enumeration	Ska
enumeration	World
enumeration	African Music
enumeration	Afro Beat
enumeration	Afro Pop
enumeration	Asian Music
enumeration	Austropop
enumeration	Calypso
enumeration	Caribbean Music

enumeration	Celtic Music
enumeration	Chanson
enumeration	Coupé Decalé
enumeration	Enka
enumeration	European Music
enumeration	Ghazal
enumeration	Griot
enumeration	Gypsy
enumeration	Highlife
enumeration	Judaica Music / Yiddish / Klezmer
enumeration	Kuduro
enumeration	Kwaito
enumeration	Makossa
enumeration	Marching Band
enumeration	Mento
enumeration	Middle Eastern Music
enumeration	Nordic / Scandinavia
enumeration	North American Music
enumeration	South American Music
enumeration	Parang
enumeration	Polka
enumeration	Rai
enumeration	Soca
enumeration	Soukous
enumeration	Zouk
enumeration	Zulu
enumeration	Latin
enumeration	Bachata
enumeration	Banda
enumeration	Bhangra
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	Gregorian Music
enumeration	Hymn
enumeration	Mass
enumeration	Spiritual
enumeration	Worship
enumeration	Miscellaneous
enumeration	Anime / Video Game Soundtracks
enumeration	Bollywood
enumeration	Instrumental
enumeration	Vocal
enumeration	Acoustic
enumeration	Unplugged
enumeration	Live
enumeration	Traditional
enumeration	Karaoke
enumeration	Movie Scores
enumeration	Movie Soundtracks
enumeration	Sound Effects
enumeration	Soundtrack
enumeration	TV Soundtrack
enumeration	Wedding Music
enumeration	Holiday
enumeration	Mashup
enumeration	unclassifiable
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 & Entrepreneurship

enumeration	Children's Audiobooks
enumeration	Popular Characters
enumeration	Animal Stories
enumeration	Children's Book Classics
enumeration	Children's 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	Esoteric
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	Literature
enumeration	Novels
enumeration	Erotica
enumeration	Romance
enumeration	Contemporary Literature
enumeration	Contemporary German Literature
enumeration	Entertainment
enumeration	Youth
enumeration	Youth Detective Stories
enumeration	Fantasy
enumeration	For Girls
enumeration	Knowledge for Teenagers
enumeration	Mystery
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	Afrikaans
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	Time
enumeration	Middle Ages
enumeration	20"s
enumeration	30"s
enumeration	40"s
enumeration	50"s
enumeration	60"s
enumeration	70"
enumeration	80"s
enumeration	90"s
enumeration	2000"s
enumeration	2010"s
enumeration	2020"s
enumeration	Adult
enumeration	Children
enumeration	Age: up to 6 years
enumeration	Age: 6 years +
enumeration	Age: 8 years +
enumeration	Kids & Family
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	South Sudan (SS)
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

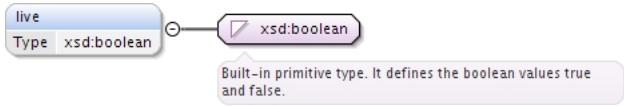
Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="bundle_only" type="xsd:boolean" maxOccurs="1" minOccurs="0" />

Element tags / explicit_lyrics

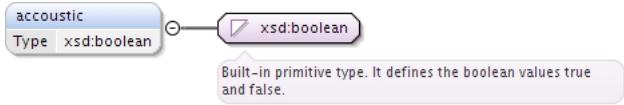
Namespace	No namespace
Diagram	
Type	explicitLyrics
Properties	content: simple minOccurs: 0 maxOccurs: 1
Facets	enumeration true enumeration false enumeration cleaned
Source	<xsd:element name="explicit_lyrics" type="explicitLyrics" maxOccurs="1" minOccurs="0" />

Element tags / live

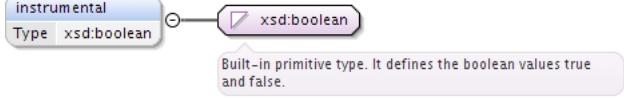
Namespace	No namespace
-----------	--------------

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

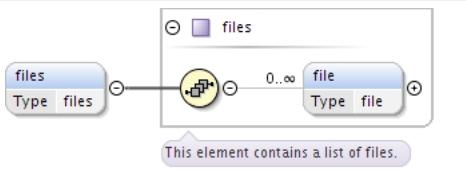
Element tags / accoustic

Namespace	No namespace
Diagram	
Type	xsd:boolean
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Source	<xsd:element name="accoustic" 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	<files> <file>{0..unbounded}</file> </files>
Source	<xsd:element name="files" type="files" maxOccurs="1" minOccurs="0"/>

Element files / file

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

Diagram



Type	file
Properties	<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} no_file_given{0,1} comment{0,1})
Children	bitrate, bitratetype, bytes, channels, checksums, codec, codecsettings, comment, decryptinfo, dimension, filetype, location, no_file_given, 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> <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> <no_file_given>{0,1}</no_file_given> <comment>{0,1}</comment> </file></pre>
Source	<code><xsd:element name="file" type="file" maxOccurs="unbounded" minOccurs="0"/></code>

Element file / location

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

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

Element fileLocation / http

Namespace	No namespace
Diagram	<pre> classDiagram fileHttp < -- action fileHttp < -- http : Type=fileHttp http < -- url : Type=url http < -- user : Type=xsd:string http < -- pass : Type=xsd:string http < -- expiresdatetime : Type=datetimeGMT note over http: This element contains information about http access to file. </pre>
Type	fileHttp
Type hierarchy	<ul style="list-style-type: none"> • action • fileHttp
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(url user{0,1} pass{0,1} expiresdatetime)
Children	expiresdatetime, pass, url, user
Instance	<pre> <http> <url>{1,1}</url> <user>{0,1}</user> <pass>{0,1}</pass> <expiresdatetime>{1,1}</expiresdatetime> </http> </pre>
Source	<xsd:element name="http" type="fileHttp" maxOccurs="1" minOccurs="0" />

Element fileHttp / url

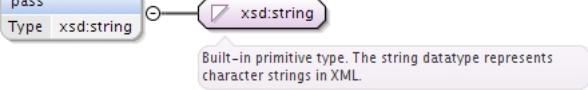
Namespace	No namespace
Diagram	<pre> url < -- xsd:string http : Type=url < -- url </pre>
Type	url
Properties	content: simple
Source	<xsd:element name="url" type="url" />

Element fileHttp / user

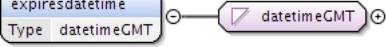
Namespace	No namespace
Diagram	<pre> user < -- xsd:string http : Type=user < -- user </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>

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

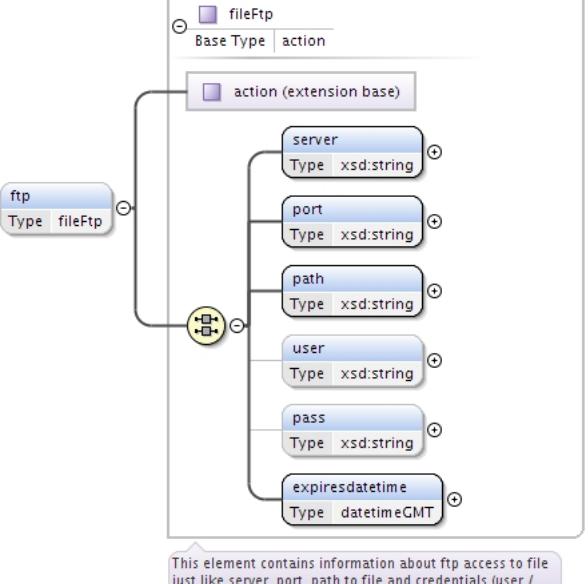
Element fileHttp / pass

Namespace	No namespace
Diagram	 <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="pass" type="xsd:string" maxOccurs="1" minOccurs="0" />

Element fileHttp / 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 fileLocation / ftp

Namespace	No namespace
Diagram	 <p>This element contains information about ftp access to file just like server, port, path to file and credentials (user / ...)</p>
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> <user>{0,1}</user> <pass>{0,1}</pass> <expiresdatetime>{1,1}</expiresdatetime> </ftp></pre>
Source	<code><xsd:element name="ftp" type="fileFtp" maxOccurs="1" minOccurs="0" /></code>

Element fileFtp / server

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "server" with a small circle icon to its right, indicating it is a complex type. A line connects this to a purple rounded rectangle labeled "xsd:string". A callout box points to the "xsd:string" box with the text: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="server" type="xsd:string" /></code>

Element fileFtp / port

Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "port" with a small circle icon to its right, indicating it is a complex type. A line connects this to a purple rounded rectangle labeled "xsd:string". A callout box points to the "xsd:string" box with the text: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="port" type="xsd:string" /></code>

Element fileFtp / path

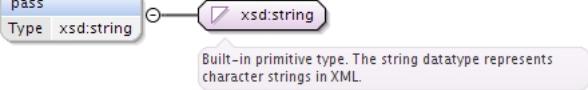
Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "path" with a small circle icon to its right, indicating it is a complex type. A line connects this to a purple rounded rectangle labeled "xsd:string". A callout box points to the "xsd:string" box with the text: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xsd:string
Properties	content: simple
Source	<code><xsd:element name="path" type="xsd:string" /></code>

Element fileFtp / user

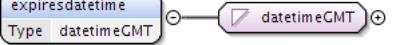
Namespace	No namespace
Diagram	<p>The diagram shows a blue rounded rectangle labeled "user" with a small circle icon to its right, indicating it is a complex type. A line connects this to a purple rounded rectangle labeled "xsd:string". A callout box points to the "xsd:string" box with the text: "Built-in 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="user" type="xsd:string" maxOccurs="1" minOccurs="0" />

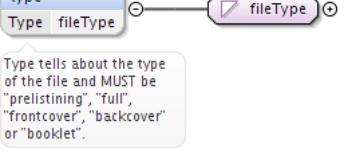
Element fileFtp / pass

Namespace	No namespace
Diagram	 <p>The diagram shows a class named 'pass' with a multiplicity of 0..1. It has a directed association labeled 'Type' pointing to a class named 'xsd:string' with a multiplicity of 1..1. A callout box from 'xsd:string' contains 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	 <p>The diagram shows a class named 'expiresdatetime' with a multiplicity of 0..1. It has a directed association labeled 'Type' pointing to a class named 'datetimeGMT' with a multiplicity of 1..1.</p>
Type	datetimeGMT
Properties	content: simple
Facets	pattern \d{4}-\d{2}-\d{2}\n \d{2}:\d{2}:\d{2} GMT\n +\d{2}:\d{2}
Source	<xsd:element name="expiresdatetime" type="datetimeGMT" />

Element file / type

Namespace	No namespace
Annotations	Type tells about the type of the file and MUST be "prelistining", "full", "frontcover", "backcover" or "booklet".
Diagram	 <p>The diagram shows a class named 'type' with a multiplicity of 0..1. It has a directed association labeled 'Type' pointing to a class named 'fileType' with a multiplicity of 1..1. A callout box from 'fileType' contains the text: 'Type tells about the type of the file and MUST be "prelistining", "full", "frontcover", "backcover" or "booklet".'</p>
Type	fileType
Properties	content: simple minOccurs: 0 maxOccurs: 1
Facets	enumeration full enumeration prelistining enumeration frontcover enumeration backcover enumeration booklet
Source	<xsd:element name="type" type="fileType" maxOccurs="1" minOccurs="0" > <xsd:annotation>

```

<xsd:documentation xml:lang="en">Type tells about the type of the file and MUST be
"prelistining", "full", "frontcover", "backcover" or "booklet".</xsd:documentation>
</xsd:annotation>
</xsd:element>

```

Element file / filetype

Namespace	No namespace						
Annotations	Filetype should accept e.g. "MP3" also as "MPEG-1 Audio Layer III".						
Diagram	<p>The diagram shows a class named 'filetype' with a dependency arrow pointing to the 'xsd:string' type. A callout box next to 'filetype' states: 'Filetype should accept e.g. "MP3" also as "MPEG-1 Audio Layer III".' A callout box next to 'xsd:string' states: '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="filetype" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Filetype should accept e.g. "MP3" also as "MPEG-1 Audio Layer III".</xsd:documentation> </xsd:annotation> </xsd:element> </pre>						

Element file / samplerate

Namespace	No namespace						
Annotations	Samplerate in khz.						
Diagram	<p>The diagram shows a class named 'samplerate' with a dependency arrow pointing to the 'xsd:string' type. A callout box next to 'samplerate' states: 'Samplerate in khz.' A callout box next to 'xsd:string' states: '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="samplerate" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Samplerate in khz.</xsd:documentation> </xsd:annotation> </xsd:element> </pre>						

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	<p>The diagram shows a class named 'prelistening_offset' with a dependency arrow pointing to the 'xsd:int' type. A callout box next to 'prelistening_offset' states: 'prelistening_offset is in seconds. prelistening_offset is a "SHOULD" for filetype "prelistening". prelistening_offset...'. A callout box next to 'xsd:int' states: 'Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...'</p>				
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	<pre> <xsd:element minOccurs="0" name="prelistening_offset" type="xsd:int"> <xsd:annotation> </pre>				

```

<xsd:documentation xml:lang="en">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	<p>The diagram shows a class named 'prelistening_length' with a multiplicity of 0..1. It has a directed association to a class named 'xsd:int' with a multiplicity of 0..1. A callout box points to the 'xsd:int' class with the text: 'Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...'. Another callout box points to the 'prelistening_length' class with the text: 'prelistening_length is in seconds. prelistening_offset is a "SHOULD" for filetype "prelistening". prelistening_length...'. The 'xsd:int' class is represented by a rectangle with a purple border and a small icon.</p>				
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	<pre> <xsd:element minOccurs="0" name="prelistening_length" type="xsd:int"> <xsd:annotation> <xsd:documentation xml:lang="en">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> </pre>				

Element file / samplesize

Namespace	No namespace						
Annotations	Samplesize in bit.						
Diagram	<p>The diagram shows a class named 'samplesize' with a multiplicity of 0..1. It has a directed association to a class named 'xsd:string' with a multiplicity of 0..1. A callout box points to the 'xsd:string' class with the text: 'Built-in primitive type. The string datatype represents character strings in XML.'. Another callout box points to the 'samplesize' class with the text: 'Samplesize in bit.'. The 'xsd:string' class is represented by a rectangle with a purple border and a small icon.</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="samplesize" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Samplesize in bit.</xsd:documentation> </xsd:annotation> </xsd:element> </pre>						

Element file / bitrate

Namespace	No namespace						
Annotations	Bitrate in kilobitpersecond e.g. 256 (on VBR) or 96/320 on (MBR).						
Diagram	<p>The diagram shows a class named 'bitrate' with a multiplicity of 0..1. It has a directed association to a class named 'xsd:string' with a multiplicity of 0..1. A callout box points to the 'xsd:string' class with the text: 'Built-in primitive type. The string datatype represents character strings in XML.'. Another callout box points to the 'bitrate' class with the text: 'Bitrate in kilobitpersecond e.g. 256 (on VBR) or 96/320 on (MBR).'. The 'xsd:string' class is represented by a rectangle with a purple border and a small icon.</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="bitrate" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Bitrate in kilobitpersecond e.g. 256 (on VBR) or 96/320 on (MBR).</xsd:documentation> </xsd:annotation> </xsd:element></pre>
--------	--

Element file / bitratetype

Namespace	No namespace						
Annotations	Type of bitrate such as CBR/VBR/MBR.						
Diagram	<p>The diagram shows a rounded rectangle labeled "bitratetype" with a blue border. To its right is a small circle with a minus sign, indicating it's a complex type. A line connects "bitratetype" to another rounded rectangle labeled "xsd:string" with a purple border. A tooltip below "bitratetype" states: "Type of bitrate such as CBR/VBR/MBR." A tooltip next to the connection line states: "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="bitratetype" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Type of bitrate such as CBR/VBR/MBR.</xsd:documentation> </xsd:annotation> </xsd:element></pre>						

Element file / codec

Namespace	No namespace						
Annotations	Name of used codec.						
Diagram	<p>The diagram shows a rounded rectangle labeled "codec" with a blue border. To its right is a small circle with a minus sign, indicating it's a complex type. A line connects "codec" to another rounded rectangle labeled "xsd:string" with a purple border. A tooltip below "codec" states: "Name of used codec." A tooltip next to the connection line states: "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="codec" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Name of used codec.</xsd:documentation> </xsd:annotation> </xsd:element></pre>						

Element file / codecsettings

Namespace	No namespace						
Annotations	Some arbitrary settings info in reference to codec - such as "insane" or "quality=9".						
Diagram	<p>The diagram shows a rounded rectangle labeled "codecsettings" with a blue border. To its right is a small circle with a minus sign, indicating it's a complex type. A line connects "codecsettings" to another rounded rectangle labeled "xsd:string" with a purple border. A tooltip below "codecsettings" states: "Some arbitrary settings info in reference to codec - such as \"insane\" or \"quality=9\"." A tooltip next to the connection line states: "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="codecsettings" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation></pre>						

```
<xsd:documentation xml:lang="en">Some arbitrary settings info in reference to codec - such as "insane" or "quality=9".</xsd:documentation>
</xsd:annotation>
</xsd:element>
```

Element file / bytes

Namespace	No namespace						
Annotations	Length of file in bytes.						
Diagram							
Type	xsd:integer						
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="bytes" type="xsd:integer" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Length of file in bytes.</xsd:documentation> </xsd:annotation> </xsd:element></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="checksums" /></pre>

Element checksums / md5

Namespace	No namespace						
Diagram							
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="md5" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>
--------	---

Element checksums / sha1

Namespace	No namespace						
Diagram							
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="sha1" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>						

Element checksums / sha256

Namespace	No namespace						
Diagram							
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="sha256" type="xsd:string" maxOccurs="1" minOccurs="0" /></code>						

Element file / channels

Namespace	No namespace								
Diagram									
Type	fileChannels								
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								
Facets	<table border="1"> <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								
Source	<code><xsd:element name="channels" type="fileChannels" maxOccurs="1" minOccurs="0" /></code>								

Element file / dimension

Namespace	No namespace
Diagram	

Type	dimension
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	width , height
Children	height, width
Instance	<pre><dimension> <width>{1,1}</width> <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	<p>The diagram shows the 'width' element from the schema. It is a rounded rectangle labeled 'width' with a blue border. To its right is a small circle with a minus sign, indicating it is a child element. Next to it is a purple rounded rectangle labeled 'xsd:integer' with a blue border. A line connects the two.</p> <p>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
Source	<code><xsd:element name="width" type="xsd:integer" /></code>

Element dimension / height

Namespace	No namespace
Diagram	<p>The diagram shows the 'height' element from the schema. It is a rounded rectangle labeled 'height' with a blue border. To its right is a small circle with a minus sign, indicating it is a child element. Next to it is a purple rounded rectangle labeled 'xsd:integer' with a blue border. A line connects the two.</p> <p>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
Source	<code><xsd:element name="height" type="xsd:integer" /></code>

Element file / decryptinfo

Namespace	No namespace
Diagram	<p>The diagram shows the 'decryptinfo' element from the schema. It is a rounded rectangle labeled 'decryptinfo' with a blue border. To its right is a small circle with a minus sign, indicating it is a child element. Next to it is a purple rounded rectangle labeled 'decryptinfo' with a blue border. A line connects the two.</p> <p>This element contains information about decryption of corresponding file.</p> <p>cipher Type xsd:string Contains info about the cipher for decryption like AES, RUNDael, XOR, Arcfour, whatever – should be "convenient".</p> <p>initvector Type xsd:string</p> <p>key Type xsd:string</p> <p>bytes Type xsd:string</p> <p>checksums Type checksums</p>
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	<pre><xsd:element name="decryptinfo" type="decryptinfo" maxOccurs="1" minOccurs="0"/></pre>

Element decryptinfo / cipher

Namespace	No namespace
Annotations	Contains info about the cipher for decryption like AES, RIJNDAEL, XOR, Arcfour, whatever - should be "convenient".
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<pre><xsd:element name="cipher" type="xsd:string" minOccurs="0" maxOccurs="1"> <xsd:annotation> <xsd:documentation xml:lang="en">Contains info about the cipher for decryption like AES, RIJNDAEL, XOR, Arcfour, whatever - should be "convenient".</xsd:documentation> </xsd:annotation> </xsd:element></pre>

Element decryptinfo / initvector

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

Element decryptinfo / key

Namespace	No namespace
Diagram	
Type	xsd:string
Properties	content: simple minOccurs: 0

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

Element decryptinfo / bytes

Namespace	No namespace
Diagram	<p>The diagram shows the 'bytes' element with its type 'xsd:string'. A callout bubble 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	<p>The diagram shows the 'checksums' element which contains three sub-elements: 'md5', 'sha1', and 'sha256', all of type 'xsd:string'. A callout bubble indicates that 'This element contains checksums for the file.'</p>
Type	checksums
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	ALL(md5{0,1} sha1{0,1} sha256{0,1})
Children	md5, sha1, sha256
Instance	<checksums> <md5>{0,1}</md5> <sha1>{0,1}</sha1> <sha256>{0,1}</sha256> </checksums>
Source	<xsd:element name="checksums" type="checksums" minOccurs="0" maxOccurs="1"/>

Element file / no_file_given

Namespace	No namespace
Diagram	<p>The diagram shows the 'no_file_given' element with its type 'xsd:boolean'. A callout bubble indicates that 'xsd:boolean' is a 'Built-in primitive type. It defines the boolean values true and false.'</p>
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<xsd:element name="no_file_given" type="xsd:boolean" maxOccurs="1" minOccurs="0"/>

Element file / comment

Namespace	No namespace				
Diagram	<p>The diagram shows a blue rounded rectangle labeled "comment" with a small "Type" label below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A callout bubble below "xsd:string" states: "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> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xsd:element minOccurs="0" name="comment" type="xsd:string"/>				

Element bundle / purchase

Namespace	No namespace						
Diagram	<p>The diagram shows a blue rounded rectangle labeled "purchase" with a small "Type" label below it. Inside the "purchase" box, there are two other elements: "pos" and "url", each with its own "Type" label ("xsd:string"). Lines connect "purchase" to both "pos" and "url". A callout bubble below "purchase" states: "This element contains information about purchase. Mostly when this feeds recipient is a POS."</p>						
Type	purchase						
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(pos url)						
Children	pos, url						
Instance	<pre><purchase> <pos>{1,1}</pos> <url>{1,1}</url> </purchase></pre>						
Source	<xsd:element name="purchase" type="purchase" maxOccurs="1" minOccurs="0"/>						

Element purchase / pos

Namespace	No namespace		
Diagram	<p>The diagram shows a blue rounded rectangle labeled "pos" with a small "Type" label below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A callout bubble below "xsd:string" states: "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> </table>	content:	simple
content:	simple		
Source	<xsd:element name="pos" type="xsd:string"/>		

Element purchase / url

Namespace	No namespace		
Diagram	<p>The diagram shows a blue rounded rectangle labeled "url" with a small "Type" label below it. A line connects it to a purple rounded rectangle labeled "xsd:string". A callout bubble below "xsd:string" states: "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> </table>	content:	simple
content:	simple		
Source	<xsd:element name="url" type="xsd:string"/>		

Element item / license_basis

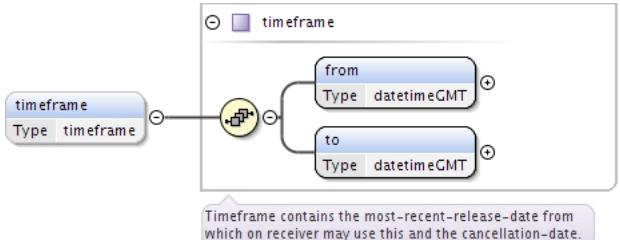
Namespace	No namespace
Diagram	<pre> classDiagram class license_basis_item { territorial timeframe pricing streaming_allowed channels as_on_bundle } license_basis_item < -- license_basis </pre>
Type	license_basis_item
Properties	content: complex
Model	ALL(territorial{0,1} timeframe{0,1} pricing{0,1}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

Element license_basis_item / territorial

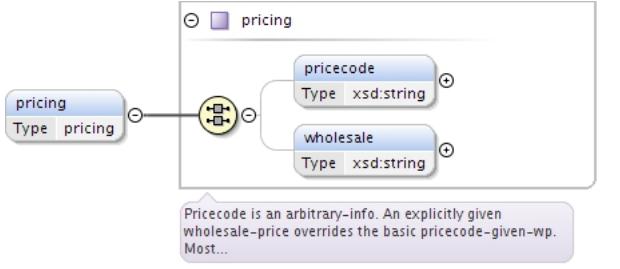
Namespace	No namespace						
Diagram	<pre> classDiagram class territorial { <> territory } territorial < -- territorial </pre>						
Type	territorial						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	territory*						
Children	territory						
Instance	<territorial> <territory type="">{0,unbounded}</territory> </territorial>						
Source	<xsd:element name="territorial" type="territorial" maxOccurs="1" minOccurs="0"/>						

Element license_basis_item / timeframe

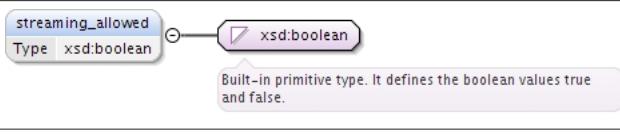
Namespace	No namespace
-----------	--------------

Diagram	 <p>Timeframe contains the most-recent-release-date from which on receiver may use this and the cancellation-date.</p>
Type	timeframe
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	from , to
Children	from, to
Instance	<pre><timeframe> <from>{1,1}</from> <to>{1,1}</to> </timeframe></pre>
Source	<code><xsd:element name="timeframe" type="timeframe" maxOccurs="1" minOccurs="0" /></code>

Element license_basis_item / pricing

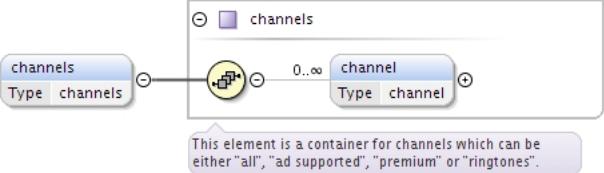
Namespace	No namespace
Diagram	 <p>Pricecode is an arbitrary-info. An explicitly given wholesale-price overrides the basic pricecode-given-wp. Most...</p>
Type	pricing
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Model	ALL(pricecode{0,1} wholesale{0,1})
Children	pricecode,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

Element license_basis_item / streaming_allowed

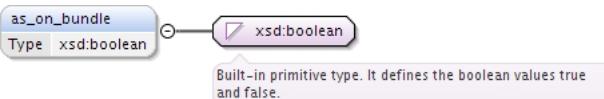
Namespace	No namespace
Diagram	 <p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xsd:boolean
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>

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

Element license_basis_item / channels

Namespace	No namespace						
Diagram	 <pre> classDiagram class channels { <<Type channels>> <<0..∞ channel>> <<Type channel>> } channels < --> channel note over channels: This element is a container for channels which can be either "all", "ad supported", "premium" or "ringtones". </pre>						
Type	channels						
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	channel*						
Children	channel						
Instance	<pre> <channels> <channel type="">{0,unbounded}</channel> </channels> </pre>						
Source	<code><xsd:element name="channels" type="channels" maxOccurs="1" minOccurs="0" /></code>						

Element license_basis_item / as_on_bundle

Namespace	No namespace						
Diagram	 <pre> classDiagram class as_on_bundle { <<xsd:boolean>> } note over as_on_bundle: Built-in primitive type. It defines the boolean values true and false. </pre>						
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	<code><xsd:element name="as_on_bundle" type="xsd:boolean" maxOccurs="1" minOccurs="0" /></code>						

Element item / license_specifics

Namespace	No namespace		
Diagram	<pre> classDiagram class license_specifics_item { <<rules>> <<0..1 as_on_bundle>> } class rules { <<Type rules>> } class as_on_bundle { <<xsd:boolean>> } note over license_specifics_item: This element includes specific rules which should be applied. </pre>		
Type	license_specifics_item		
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
content:	complex		
Model	rules{0,1} as_on_bundle{0,1}		
Children	as_on_bundle, rules		
Instance	<pre> <license_specifics> <rules>{0,1}</rules> <as_on_bundle>{0,1}</as_on_bundle> </license_specifics> </pre>		
Source	<code><xsd:element name="license_specifics" type="license_specifics_item"/></code>		

Element license_specifics_item / rules

Namespace	No namespace
Diagram	<p>This element is a container for rules. It needs an ordered mode here – first come first match.</p>
Type	rules
Properties	content: complex minOccurs: 0 maxOccurs: 1
Model	rule*
Children	rule
Instance	<pre><rules> <rule num=" ">{0,unbounded}</rule> </rules></pre>
Source	<pre><xsd:element name="rules" type="rules" maxOccurs="1" minOccurs="0" /></pre>

Element license_specifics_item / as_on_bundle

Namespace	No namespace
Diagram	<p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xsd:boolean
Properties	content: simple minOccurs: 0 maxOccurs: 1
Source	<pre><xsd:element name="as_on_bundle" type="xsd:boolean" maxOccurs="1" minOccurs="0" /></pre>

Element item / tags

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

	maxOccurs: 1
Model	ALL(genres{0,1} bundle_only{0,1} explicit_lyrics{0,1} live{0,1} accoustic{0,1} instrumental{0,1})
Children	accoustic, bundle_only, explicit_lyrics, genres, instrumental, live
Instance	<pre><tags> <genres>{0,1}</genres> <bundle_only>{0,1}</bundle_only> <explicit_lyrics>{0,1}</explicit_lyrics> <live>{0,1}</live> <accoustic>{0,1}</accoustic> <instrumental>{0,1}</instrumental> </tags></pre>
Source	<code><xsd:element name="tags" type="tags" maxOccurs="1" minOccurs="0" /></code>

Element item / fingerprint

Namespace	No namespace						
Diagram	<pre> classDiagram class fingerprint { echoprint } class echoprint { xsd:string } fingerprint "1" -- "0..1" 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

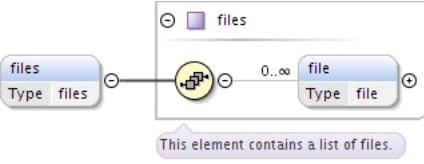
Namespace	No namespace						
Diagram	<pre> classDiagram class echoprint { xsd:string } </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>						
Type	xsd:string						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs: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

Element item / reporting

Namespace	No namespace
Diagram	<pre> classDiagram class reporting { realtime postponed } class realtime { xsd:string } class postponed { xsd:string } realtime "1" -- "0..1" realtime realtime "1" -- "0..1" postponed </pre> <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>

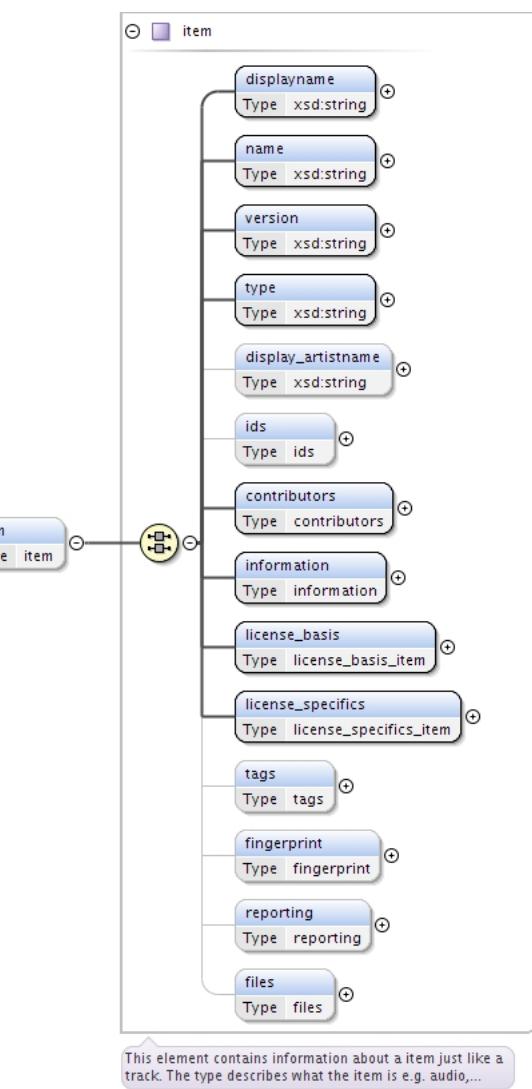
Element item / files

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

Element feed / item

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

Diagram



Type	item
Properties	<p>content: complex</p> <p>minOccurs: 0</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="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 feedid creationdatetime effectivedatetime creator receiver sender licensor licensee actions } feedinfo --> onlytest feedinfo --> feedid feedinfo --> creationdatetime feedinfo --> effectivedatetime feedinfo --> creator feedinfo --> receiver feedinfo --> sender feedinfo --> licensor feedinfo --> licensee feedinfo --> actions note over feedinfo: On feedinfo-level there are the global information needed or at least valuable for ingesting / identifying the content... note over onlytest: On feedinfo-level there are the global information needed or at least valuable for ingesting / identifying the content... </pre>
Used by	Element feed/feedinfo
Model	ALL(onlytest feedid creationdatetime effectivedatetime creator{0,1} 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, 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"</pre>

```

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" maxOccurs="1" minOccurs="0"/>
  <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	<p>The diagram illustrates the structure of the 'creator' complex type. It consists of a central node labeled 'creator' connected to three other nodes: 'email', 'userid', and 'keyid'. Each connection is labeled with its type ('Type email', 'Type userid', and 'Type xsd:string' respectively) and has a multiplicity of 0..1 indicated by a circle with a plus sign. A callout box points to the 'creator' node with the text: '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:annotation> <xsd:documentation xml:lang="en">Content should be an email-address of the *user* on the sending side.</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="userid" type="userid" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">This should be an unique id of the *user* on the sending side.</xsd:documentation> </xsd:annotation> </xsd:element> <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	
Used by	Element feedinfo/receiver
Model	ALL(type servername serverip4 serveripv6{0,1} authtype username{0,1} crypto{0,1})
Children	authtype, crypto, serverip4, 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="serverip4" type="ipv4"/> <xsd:element name="serveripv6" type="ipv6" maxOccurs="1" minOccurs="0"/> <xsd:element name="authtype" type="authtype"/> <xsd:element name="username" type="xsd:string" maxOccurs="1" minOccurs="0"/> <xsd:element name="crypto" type="crypto" maxOccurs="1" minOccurs="0"/> </xsd:all> </xsd:complexType></pre>

Complex Type crypto

Namespace	No namespace
Annotations	This element contains crypto information for secure and authenticated transfer.
Diagram	
Used by	Element receiver/crypto
Model	ALL(relatedemail{0,1} usedkeyid{0,1} usedpubkey{0,1})
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" maxOccurs="1" minOccurs="0"/> <xsd:element name="usedkeyid" type="keyid" maxOccurs="1" minOccurs="0"/> <xsd:element name="usedpubkey" type="xsd:base64Binary" maxOccurs="1" minOccurs="0"/> </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 { contractpartnerid : xsd:string ourcontractpartnerid : xsd:string email : email keyid : 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> <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> </pre>

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

Complex Type actions

Namespace	No namespace
Annotations	This element contains information about possible actions with the feed.
Diagram	
Used by	Element feedinfo/actions
Model	ALL(oninitialreceive{0,1} onprocessstart{0,1} onprocessend{0,1} onfullsuccess{0,1} onerror{0,1})
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" maxOccurs="1" minOccurs="0" /> <xsd:element name="onprocessstart" type="event" maxOccurs="1" minOccurs="0" /> <xsd:element name="onprocessend" type="event" maxOccurs="1" minOccurs="0" /> <xsd:element name="onfullsuccess" type="event" maxOccurs="1" minOccurs="0" /> <xsd:element name="onerror" type="event" maxOccurs="1" minOccurs="0" /> </xsd:all> </xsd:complexType></pre>

Complex Type event

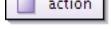
Namespace	No namespace
Annotations	This element contains information about possible events and actions.

Diagram	
Used by	Elements actions/onerror, actions/onfullsuccess, actions/oninitialreceive, actions/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>

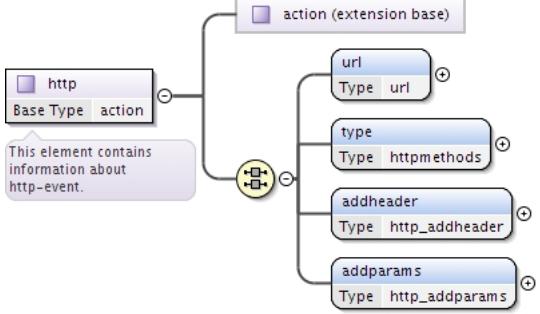
Complex Type mailto

Namespace	No namespace
Annotations	This element contains information about mailto-event.
Diagram	
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>

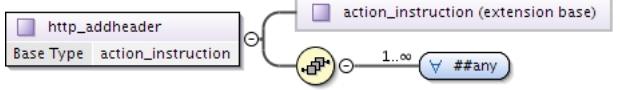
Complex Type action

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

Complex Type http

Namespace	No namespace
Annotations	This element contains information about http-event.
Diagram	
Type	extension of action
Type hierarchy	<ul style="list-style-type: none"> • action • 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

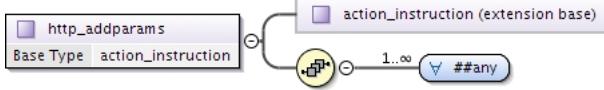
Namespace	No namespace
Diagram	
Type	extension of action_instruction
Type hierarchy	<ul style="list-style-type: none"> • action_instruction • http_addheader
Used by	Element http/addheader
Model	ANY element from ANY namespace

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>
--------	--

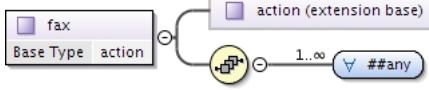
Complex Type action_instruction

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

Complex Type http_addparams

Namespace	No namespace
Diagram	
Type	extension of action_instruction
Type hierarchy	<ul style="list-style-type: none"> • action_instruction • 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	
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 class letter { <<This element contains information about the letter event.>> } letter "0..1" *-- "1" registered : registered letter "0..1" *-- "1" to : to letter "0..1" *-- "1" text : text letter "0..1" *-- "1" costscoveredby : costscoveredby </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:annotation> <xsd:documentation xml:lang="en">This tells if letter must be registered or not.</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="to" type="to"/> <xsd:element name="text" type="xsd:string"> <xsd:annotation> <xsd:documentation xml:lang="en">This contains the content/text of letter.</xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="costscoveredby" type="costscoveredby"/> </xsd:all> </xsd:complexType> </pre>

Complex Type to

Namespace	No namespace
Annotations	This element contains information about recipient.
Diagram	<pre> classDiagram class to { <<This element contains information about recipient.>> } to "0..1" *-- "1" name : name to "0..1" *-- "1" department : department to "0..1" *-- "1" nameperson : nameperson to "0..1" *-- "1" street : street to "0..1" *-- "1" postcode : postcode to "0..1" *-- "1" country : country to "0..1" *-- "1" additionaladdressinfo : additionaladdressinfo </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	
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:annotation> <xsd:documentation xml:lang="en">This contains the max amount that will be covered.</xsd:documentation> </xsd:annotation> </xsd:element> </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 { +displayname +name +version +display_artistname +ids +items +contributors +information +license_basis +license_specifics +reporting +tags +files +purchase } bundle < --> ids bundle < --> items bundle < --> contributors bundle < --> information bundle < --> license_basis bundle < --> license_specifics bundle < --> reporting bundle < --> tags bundle < --> files bundle < --> purchase </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"/> <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> </pre>

Complex Type ids

Namespace	No namespace
Annotations	This Element is a container for all IDs which are available for the associated element.
Diagram	<pre> graph TD ids[ids] --> grid["grid
Type xsd:string"] ids --> upc["upc
Type upc"] ids --> isrc["isrc
Type isrc"] ids --> contentauth["contentauth
Type xsd:string"] ids --> labelordernum["labelordernum
Type xsd:string"] ids --> amzn["amzn
Type xsd:string"] ids --> isbn["isbn
Type xsd:string"] ids --> finetunes["finetunes
Type finetunes"] ids --> licensor["licensor
Type xsd:string"] ids --> licensee["licensee
Type xsd:string"] ids --> gvl["gvl
Type xsd:string"] </pre> <p>This Element is a container for all IDs which are available for the associated element.</p>
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="upc" maxOccurs="1" minOccurs="0"/> <xsd:element name="isrc" type="isrc" 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="finetunes" 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	<pre> graph TD items[items] --> item["item
Type item"] item -- "1..oo" --> item </pre> <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 : xsd:string name : xsd:string version : xsd:string type : xsd:string display_artistname : xsd:string ids : ids contributors : contributors information : information license_basis : license_basis_item license_specifics : license_specifics_item tags : tags fingerprint : fingerprint reporting : reporting files : files } item < -- items </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</pre>

```
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_item"/>
<xsd:element name="license_specifics" type="license_specifics_item"/>
<xsd:element name="tags" type="tags" maxOccurs="1" minOccurs="0"/>
<xsd:element name="fingerprint" type="fingerprint" maxOccurs="1" minOccurs="0"/>
<xsd:element name="reporting" type="reporting" maxOccurs="1" minOccurs="0"/>
<xsd:element name="files" type="files" maxOccurs="1" minOccurs="0"/>
</xsd:all>
</xsd:complexType>
```

Complex Type contributors

Namespace	No namespace
Annotations	This element contains a list of contributor.
Diagram	<pre> classDiagram class contributors { <<This element contains a list of contributor.>> } class contributor { <<Type contributor>> } contributors "0..infinity" --> "0..infinity" contributor </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="0"/> </xsd:sequence> </xsd:complexType> </pre>

Complex Type contributor

Namespace	No namespace
Annotations	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.
Diagram	<pre> classDiagram class contributor { <<This element contains information of one contributor. A contributor can be a label, performer, texter, editor,...>> } class attributes { num xsd:integer name xsd:string type contributorType year xsd:string ids ids www } contributor "0..1" --> "0..1" attributes </pre>

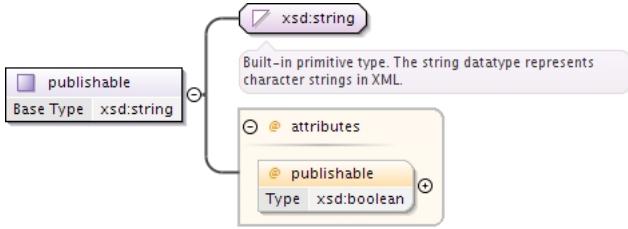
Used by	Element	contributors/contributor			
Model	ALL(name type year{0,1} ids www{0,1})				
Children	ids, name, type, www, year				
Attributes	QName	Type	Fixed	Default	Use
	num	xsd:integer			optional
Source	<pre><xsd:complexType name="contributor"> <xsd:annotation> <xsd:documentation xml:lang="en">This 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

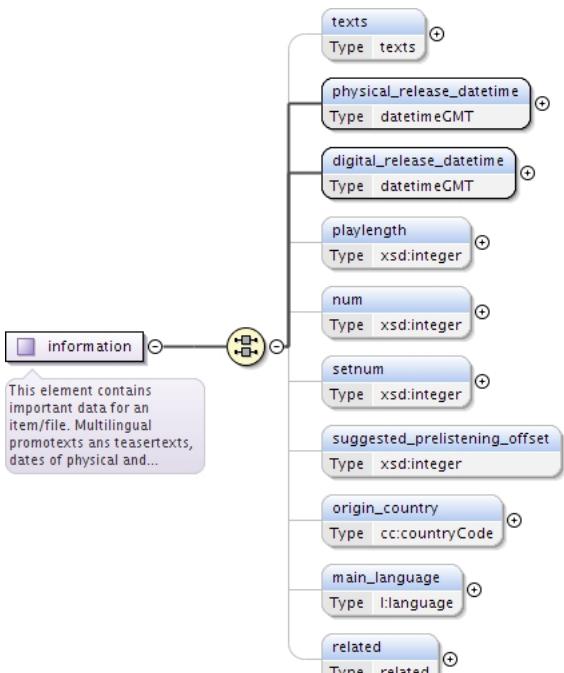
Namespace	No namespace				
Annotations	<p>This Element is a container for the important web addresses and phone of the associated element (contributor e.g.). Phone should be in international format.</p> <p>Every single information-entry cold be tagged "publishable" which would then mean wether customers 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 +myspace +homepage +twitter +phone } www < -- publishable facebook < -- publishable myspace < -- publishable homepage < -- publishable twitter < -- publishable phone < -- publishable www --> Note: This Element is a container for the important web addresses and phone of the associated element (contributor e.g....) </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											
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	This element contains important data for an item/file. Multilingual promotexts ans teasertexts, dates of physical and digital release, playlength of file, position of the file in relation to other file of bundle, number of set (e.g. 2 for cd 2), the suggested prelistining offset if the file not starts e.g. with significant content, origin country and main language of file and information about related bundles.
Diagram	
Used by	Elements bundle/information, item/information
Model	ALL(texts{0,1} physical_release_datetime digital_release_datetime playlength{0,1} num{0,1} setnum{0,1} suggested_prelistening_offset{0,1} origin_country{0,1} main_language{0,1} related{0,1})
Children	digital_release_datetime, main_language, num, origin_country, physical_release_datetime, playlength, related, setnum, suggested_prelistening_offset, texts
Source	<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,</pre>

```

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>
```

Complex Type texts

Namespace	No namespace
Annotations	This element contains multilingual promotexts ans teasertexts.
Diagram	<pre> graph LR texts[texts] --> promotext[promotext] texts --> teasertext[teasertext] </pre> <p>This element contains multilingual promotexts ans teasertexts.</p>
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	<pre> graph LR promotext[promotext Base Type: xsd:string] --> xsdstring[xsd:string] xsdstring --> attributes[attributes] attributes --> lang[lang Type: xsd:string] </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>										
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

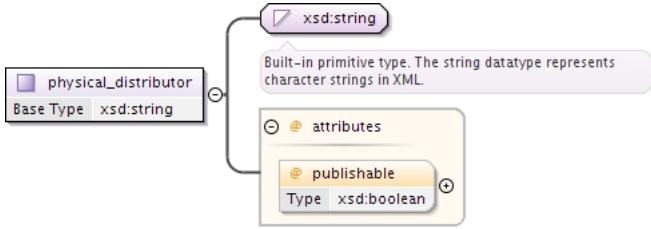
Namespace	No namespace										
Diagram	<pre> graph LR teasertext[xsd:string teasertext] --> lang[xsd:string @ lang] </pre>										
Type	extension of xsd:string										
Used by	Element texts/teasertext										
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <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"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="lang" type="xsd:string"/> </xsd:extension> </xsd:simpleContent> </xsd:complexType> </pre>										

Complex Type related

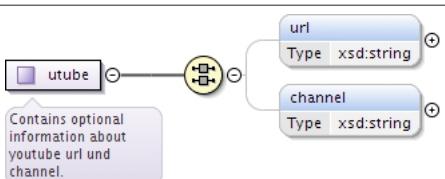
Namespace	No namespace
Annotations	This element contains informations of bundles which are related to the bundle of the actual feed. It may includes one or more physical distributors and one element "utube" which could include information about channel and url at youtube.
Diagram	<pre> graph LR related[xsd:string related] --> pd[physical_distributor 0..∞] related --> utube[utube 0..1] related --> b[bundle 0..∞] </pre>
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. It may includes one or more physical distributors and one element "utube" which could include information about channel and url at youtube.</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

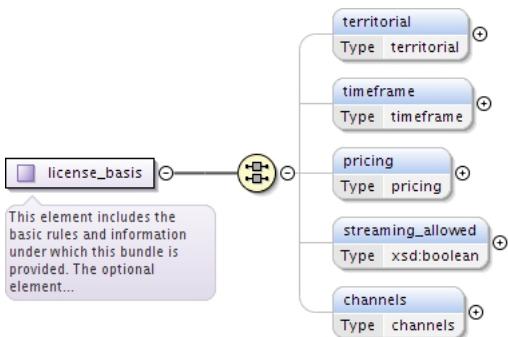
Namespace	No namespace
-----------	--------------

Diagram											
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

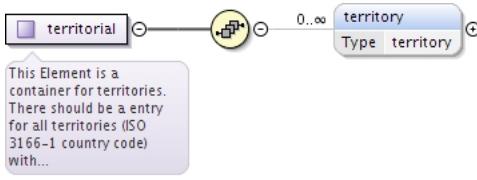
Namespace	No namespace
Annotations	Contains optional information about youtube url und channel.
Diagram	
Used by	Element related/utube
Model	ALL(url{0,1} channel{0,1})
Children	channel, url
Source	<pre><xsd:complexType name="utube"> <xsd:annotation> <xsd:documentation xml:lang="en">Contains optional 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

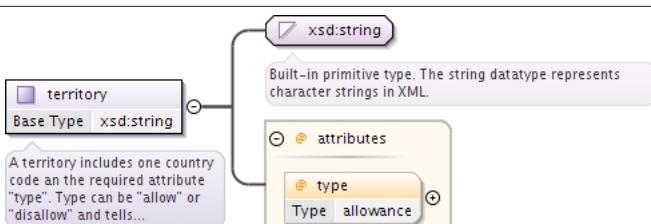
Namespace	No namespace
Annotations	This element includes the basic rules and information under which this bundle is provided. The optional element "streaming_allowed" tells if streaming is allowed or not.
Diagram	

Used by	Element bundle/license_basis
Model	ALL(territorial{0,1} timeframe{0,1} pricing{0,1} streaming_allowed{0,1} channels{0,1})
Children	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. The optional element "streaming_allowed" tells if streaming is allowed or not.</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: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 (ISO 3166-1 country code) with a attribute if distribution is allowed or not. "WW" means "World Wide" and is a wildcard for all territories.
Diagram	
Used by	Elements license_basis/territorial, license_basis_item/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 (ISO 3166-1 country code) with a attribute if distribution is allowed or not. "WW" means "World Wide" and is a wildcard for all territories.</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" and tells about the allowance of corresponding territory.										
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" and tells about the allowance of corresponding territory.</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	<pre> graph LR timeframe[timeframe] --> from[from Type datetimeGMT] timeframe --> to[to Type datetimeGMT] </pre> <p>Timeframe contains the most-recent-release-date from which on receiver may use this and the cancellation-date.</p>
Used by	Elements license_basis/timeframe, license_basis_item/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> graph LR pricing[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>
Used by	Elements license_basis/pricing, license_basis_item/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", "premium" or "ringtones".
Diagram	<p>This element is a container for channels which can be either "all", "ad supported", "premium" or "ringtones".</p>
Used by	Elements license_basis/channels, license_basis_item/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", "premium" or "ringtones".</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", "premium" or "ringtones". The required attribute "type" regards to the allowance in reference to the channel. Type can be "allow" or "disallow".										
Diagram	<p>This element is a container for channels which can be either "all", "ad supported", "premium" or "ringtones". The required attribute "type" regards to the allowance in reference to the channel. Type can be "allow" or "disallow".</p> <p>Built-in primitive type. The string datatype represents character strings in XML.</p> <p>@ attributes</p> <p>@ type</p> <p>Type allowance</p>										
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", "premium" or "ringtones". 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.
Diagram	<p>This element includes specific rules which should be applied.</p>

Used by	Element	bundle/license_specifics
Model	ALL(rules{0,1})	
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.</xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="rules" type="rules" maxOccurs="1" minOccurs="0" /> </xsd:all> </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		
Used by	Elements license_specifics/rules, license_specifics_item/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											
Used by	Element rules/rule										
Model	if, then, else{0,1}										
Children	else, if, then										
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="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></pre>									

```
</xsd:annotation>
<xsd:sequence>
  <xsd:element name="if" type="if"/>
  <xsd:element name="then" type="then"/>
  <xsd:element name="else" type="else" maxOccurs="1" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="num" type="xsd:integer"/>
</xsd:complexType>
```

Complex Type if

Namespace	No namespace
Annotations	This element must be the first element in a rule. It includes the information what is affected by the rule, an operator like "equals", "before", "after", "contains" or "containedin" and a value which will be compared.
Diagram	
Used by	Element rule/if
Model	what , operator , value
Children	operator, value, what
Source	<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: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. It also could include information "proclaim".
Diagram	
Used by	Element rule/then
Model	proclaim*, echo{0,1} , break{0,1}
Children	break, echo, proclaim
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. It also could include information "proclaim".</xsd:documentation></pre>

```
</xsd:annotation>
<xsd:sequence>
  <xsd:element name="proclaim" type="proclaim" maxOccurs="unbounded" minOccurs="0" />
  <xsd:element name="echo" type="xsd:string" maxOccurs="1" minOccurs="0" />
  <xsd:element name="break" maxOccurs="1" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
```

Complex Type proclaim

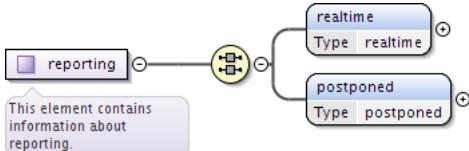
Namespace	No namespace
Annotations	This element includes the information what is affected and the corresponding value.
Diagram	
Used by	Elements else/proclaim, then/proclaim
Model	what , for
Children	for, what
Source	<pre><xsd:complexType name="proclaim"> <xsd:annotation> <xsd:documentation xml:lang="en">This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

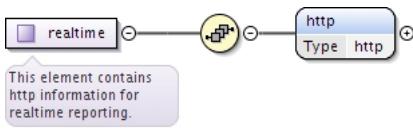
Namespace	No namespace
Annotations	This element is optional. It includes information "proclaim" and can include an element "break" which means to not process any more rules.
Diagram	
Used by	Element rule/else
Model	proclaim*, break{0,1}
Children	break, proclaim
Source	<pre><xsd:complexType name="else">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

Complex Type reporting

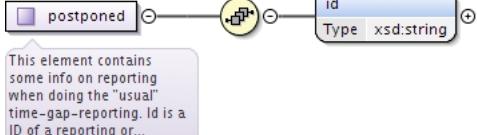
Namespace	No namespace
Annotations	This element contains information about reporting.

Diagram	
Used by	Elements bundle/reporting, item/reporting
Model	ALL(realtime postponed)
Children	postponed, realtime
Source	<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	
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. Id is a ID of a reporting or similar.
Diagram	
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. Id is a ID of a reporting or similar.</xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="id" type="xsd:string"/> </xsd:sequence> </xsd:complexType></pre>

</xsd:complexType>

Complex Type tags

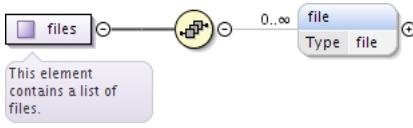
Namespace	No namespace
Annotations	This element contains information about genres and more.
Diagram	<pre> classDiagram class tags { <<This element contains information about genres and more.>> } class genres class bundle_only class explicit_lyrics class live class acoustic class instrumental tags "0..1" *-- "1..1" genres tags "0..1" *-- "1..1" bundle_only tags "0..1" *-- "1..1" explicit_lyrics tags "0..1" *-- "1..1" live tags "0..1" *-- "1..1" acoustic tags "0..1" *-- "1..1" instrumental </pre>
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	acoustic, 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="acoustic" 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> classDiagram class genres { <<This element contains a list of genres.>> } class genre genres "0..*" *-- "0..*" genre </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	
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. In the case of an update or takedown with no extra file given, set "no_file_given" to "true"

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 no_file_given <--> xsd:boolean comment <--> xsd:string } </pre> <p>This element contains information and location of a file. In the case of an update or takedown with no extra file...</p>
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} no_file_given{0,1} comment{0,1})
Children	bitrate, bitratetype, bytes, channels, checksums, codec, codecsettings, comment, decryptinfo, dimension, filetype, location, no_file_given, 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. In the case of an update or takedown with no extra file given, set "no_file_given" to "true" </xsd:documentation> </xsd:annotation> <xsd:all> <xsd:element name="location" type="fileLocation"/> <xsd:element name="type" type="fileType" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Type tells about the type of the file and MUST be "prelistining", "full", "frontcover", "backcover" or "booklet". </xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="filetype" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Filetype should accept e.g. "MP3" also as "MPEG-1 Audio Layer III". </xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="samplerate" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Samplerate in khz. </xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element minOccurs="0" name="prelistening_offset" type="xsd:int"> <xsd:annotation> <xsd:documentation xml:lang="en">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"> <xsd:annotation> <xsd:documentation xml:lang="en">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:annotation> <xsd:documentation xml:lang="en">Samplesize in bit. </xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="bitrate" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Bitrate in kilobitpersecond e.g. 256 (on VBR) or 96/320 on (MBR). </xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="bitratetype" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Type of bidrate such as CBR/VBR/MBR. </xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="codec" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Name of used codec. </xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="codecsettings" type="xsd:string" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Some arbitrary settings info in reference to codec - such as "insane" or "quality=9". </xsd:documentation> </xsd:annotation> </xsd:element> <xsd:element name="bytes" type="xsd:integer" maxOccurs="1" minOccurs="0"> <xsd:annotation> <xsd:documentation xml:lang="en">Length of file in bytes. </xsd:documentation> </xsd:annotation> </xsd:element> <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> </pre>

```

<xsd:element name="no_file_given" type="xsd:boolean" maxOccurs="1" minOccurs="0" />
<xsd:element minOccurs="0" name="comment" type="xsd:string" />
</xsd:all>
</xsd:complexType>

```

Complex Type fileLocation

Namespace	No namespace
Annotations	This element contains the path to the corresponding file. File can be accessible via path, http or ftp.
Diagram	<pre> classDiagram class fileLocation { <<This element contains the path to the corresponding file. File can be accessible via path, http or ftp.>> } class path { <<path</>> <<Type xsd:string</>> } class http { <<http</>> <<Type fileHttp</>> } class ftp { <<ftp</>> <<Type fileFtp</>> } fileLocation < -- path fileLocation < -- http fileLocation < -- 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. File can be accessible via path, http or ftp.</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 { <<action (extension base)>> <<Base Type action</>> <<This element contains information about http access to file.>> } class action { <<action (extension base)>> } class url { <<url</>> <<Type url</>> } class user { <<user</>> <<Type xsd:string</>> } class pass { <<pass</>> <<Type xsd:string</>> } class expiresdatetime { <<expiresdatetime</>> <<Type datetimeGMT</>> } fileHttp < -- action action < -- url action < -- user action < -- pass action < -- 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:all> </xsd:extension> </xsd:complexContent> </xsd:complexType> </pre>

```

<xsd:element name="expiresdatetime" type="datetimeGMT" />
</xsd:all>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

Complex Type fileFtp

Namespace	No namespace
Annotations	This element contains information about ftp access to file just like server, port, path to file and credentials (user / password). The expiredate tells until when this file is definitely available to be called.
Diagram	<pre> classDiagram class fileFtp { <<Base Type action>> } class action { <<extension base>> } fileFtp --o action action <--> server :xsd:string action <--> port :xsd:string action <--> path :xsd:string action <--> user :xsd:string action <--> pass :xsd:string action <--> 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 just like server, port, path to file and credentials (user / password). The expiredate tells until when this file is definitely available to be called.</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> classDiagram class checksums { <<This element contains checksums for the file.>> } class Base { <<Base>> } checksums --o Base Base <--> md5 :xsd:string Base <--> sha1 :xsd:string Base <--> sha256 :xsd:string </pre>

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 the dimension (width and height) of the file.
Diagram	<pre> classDiagram class dimension { <<This element contains entries for the dimension (width and height) of the file.>> } class width { <<Type xsd:integer>> } class height { <<Type xsd:integer>> } dimension "3" -- "2" width dimension "3" -- "2" height </pre>
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 the dimension (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> classDiagram class decryptinfo { <<This element contains information about decryption of corresponding file.>> } class cipher { <<Type xsd:string>> } class initvector { <<Type xsd:string>> } class key { <<Type xsd:string>> } class bytes { <<Type xsd:string>> } class checksums { <<Type checksums>> } decryptinfo "3" -- "5" cipher decryptinfo "3" -- "5" initvector decryptinfo "3" -- "5" key decryptinfo "3" -- "5" bytes decryptinfo "3" -- "5" checksums </pre> <p>Contains info about the cipher for decryption like AES, RUMDAEL, XOR, Arcfour, whatever - should be "convenient".</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></pre>

```

<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:annotation>
      <xsd:documentation xml:lang="en">Contains info about the cipher for decryption
like AES, RIJNDAEL, XOR, Arcfour, whatever - should be "convenient".</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="initvector" type="xsd:string" minOccurs="0" maxOccurs="1"/>
  <xsd:element name="key" type="xsd:string" minOccurs="0" maxOccurs="1"/>
  <xsd:element name="bytes" type="xsd:string" minOccurs="0" maxOccurs="1"/>
  <xsd:element name="checksums" type="checksums" minOccurs="0" maxOccurs="1"/>
</xsd:all>
</xsd:complexType>

```

Complex Type purchase

Namespace	No namespace
Annotations	This element contains information about purchase. Mostly when this feeds recipient is a POS.
Diagram	<pre> graph LR purchase[purchase] --> pos[pos
Type xsd:string] purchase --> url[url
Type xsd:string] </pre> <p>This element contains information about purchase. Mostly when this feeds recipient is a POS.</p>
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 license_basis_item

Namespace	No namespace
Annotations	This element includes the basic rules and information under which this bundle is provided. The optional element "streaming_allowed" tells if streaming is allowed or not
Diagram	<pre> graph LR license_basis_item[license_basis_item] --> territorial[territorial
Type territorial] license_basis_item --> timeframe[timeframe
Type timeframe] license_basis_item --> pricing[pricing
Type pricing] license_basis_item --> streaming_allowed[streaming_allowed
Type xsd:boolean] license_basis_item --> channels[channels
Type channels] license_basis_item --> as_on_bundle[as_on_bundle
Type xsd:boolean] </pre> <p>This element includes the basic rules and information under which this bundle is provided. The optional element...</p>
Used by	Element 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_item"> <xsd:annotation> <xsd:documentation xml:lang="en">This element includes the basic rules and information under which this bundle is provided. The optional element "streaming_allowed" tells if streaming is allowed or not</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 license_specifics_item

Namespace	No namespace
Annotations	This element includes specific rules which should be applied.
Diagram	<pre> classDiagram class license_specifics_item { <<This element includes specific rules which should be applied.>> } class rules { <<Type xsd:boolean>> } class as_on_bundle { <<Type xsd:boolean>> } license_specifics_item "1..2" -- "0..1" rules : Type xsd:boolean license_specifics_item "1..2" -- "0..1" as_on_bundle : Type xsd:boolean </pre>
Used by	Element item/license_specifics
Model	rules{0,1} as_on_bundle{0,1}
Children	as_on_bundle, rules
Source	<pre><xsd:complexType name="license_specifics_item"> <xsd:annotation> <xsd:documentation xml:lang="en">This element includes specific rules which should be applied.</xsd:documentation> </xsd:annotation> <xsd:choice minOccurs="1" maxOccurs="2"> <xsd:element name="rules" type="rules" maxOccurs="1" minOccurs="0"/> <xsd:element name="as_on_bundle" type="xsd:boolean" maxOccurs="1" minOccurs="0"/> </xsd:choice> </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	http://echoprint.me https://github.com/echonest/echoprint-codegen).'"/> <pre> classDiagram class fingerprint { <<This element includes an element "echoprint" (http://echoprint.me https://github.com/echonest/echoprint-codegen).>> } class echoprint { <<Type xsd:string>> } fingerprint "1..2" -- "0..1" echoprint : Type xsd:string </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:complexContent> <xsd:extension base="event" /> </xsd:complexContent> </xsd:complexType> </pre>

</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	<pre> classDiagram event < -- onprocessend event < -- Base Type event event --> mailto : 0..oo event --> http : 0..oo event --> fax : 0..oo event --> letter : 0..oo </pre> <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	<pre> classDiagram event < -- onfullsuccess event < -- Base Type event event --> mailto : 0..oo event --> http : 0..oo event --> fax : 0..oo event --> letter : 0..oo </pre> <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"> </pre>

```

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

```

Complex Type onerror

Namespace	No namespace
Annotations	This element contains information about what should be done on error processing the feed.
Diagram	<p>This element contains information about what should be done on error 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 • 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 datetimeGMT

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

</xsd:simpleType>

Simple Type email

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	<p>pattern</p> <pre>(([a-zA-Z0-9_]-[.]+)@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*(\.[a-zA-Z]{2,3}))?</pre>
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:pattern value="(([a-zA-Z0-9_]-[.]+)@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*(\.[a-zA-Z]{2,3}))?" /> </xsd:restriction> </xsd:simpleType></pre>

Simple Type userid

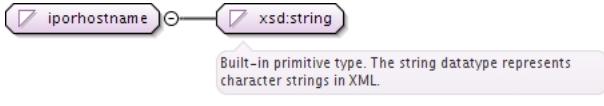
Namespace	No namespace
Diagram	
Type	xsd:string
Used by	Element creator/userid
Source	<pre><xsd:simpleType name="userid"> <xsd:restriction base="xsd:string"/> </xsd:simpleType></pre>

Simple Type receivertypes

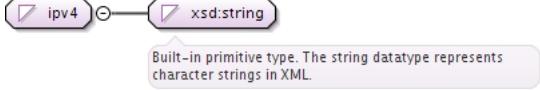
Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	<p>enumeration ftp</p> <p>enumeration ftps</p> <p>enumeration sftp</p> <p>enumeration webdav</p> <p>enumeration openSDX fileserver</p>
Used by	Element receiver/type
Source	<pre><xsd:simpleType name="receivertypes"> <xsd:restriction 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

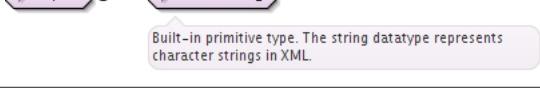
Namespace	No namespace
-----------	--------------

Diagram	
Type	xsd:string
Used by	Element receiver/servername
Source	<pre><xsd:simpleType name="iporhostname"> <xsd:restriction base="xsd:string"/> </xsd:simpleType></pre>

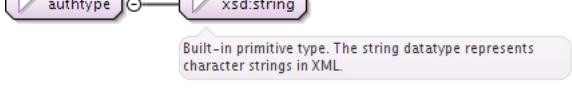
Simple Type ipv4

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	<p>pattern</p> <pre>(25[0-5] 2[0-4][0-9] 1[0-9][0-9] 1[0-9] {1,2})(\.(25[0-5] 2[0-4][0-9] 1[0-9] [0-9] 1[0-9]{1,2})) {3}</pre>
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[0-9] {1,2})(\.(25[0-5] 2[0-4][0-9] 1[0-9] [0-9] 1[0-9]{1,2})) {3}"> <xsd:annotation> <xsd:documentation xml:lang="en">Internet Protocol version 4 (IPv4) is the fourth revision in the development of the Internet Protocol (IP) and the first version of the protocol to be widely deployed. Valide ipv4-addresses includes four dotted separated blocks with digits between 0 and 255.</xsd:documentation> </xsd:annotation> </xsd:pattern> </xsd:restriction> </xsd:simpleType></pre>

Simple Type ipv6

Namespace	No namespace
Diagram	
Type	xsd:string
Used by	Element receiver/serverip6
Source	<pre><xsd:simpleType name="ipv6"> <xsd:restriction base="xsd:string"> <!-- not pattern defined yet... --> </xsd:restriction> </xsd:simpleType></pre>

Simple Type authtype

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	<p>enumeration</p> <p>login</p>

	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>	

Simple Type keyid

Namespace	No namespace
Diagram	<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

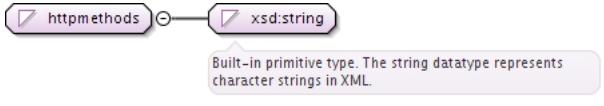
Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Element mailto/receiver
Source	<pre><xsd:simpleType name="emaillist"> <xsd:restriction base="xsd:string"> <!-- make to NMOKENS or such... --> </xsd:restriction>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

Simple Type url

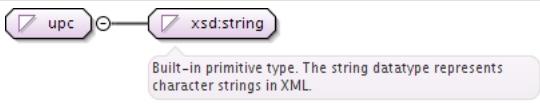
Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xsd:string
Used by	Elements fileHttp/url, http/url
Source	<pre><xsd:simpleType name="url"> <xsd:restriction base="xsd:string"> <!-- not pattern defined yet... --> </xsd:restriction>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

Simple Type httpmethods

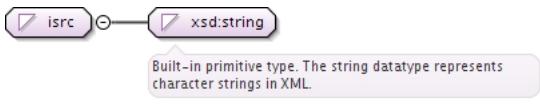
Namespace	No namespace
-----------	--------------

Diagram							
Type	restriction of xsd:string						
Facets	<table> <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 upc

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	pattern $(\d{10,13})?$
Used by	Element ids/upc
Source	<pre><xsd:simpleType name="upc"> <xsd:restriction base="xsd:string"> <xsd:pattern value="(\d{10,13})?"> <xsd:annotation> <xsd:documentation xml:lang="en">The Universal Product Code (UPC) is a barcode symbology (i.e., a specific type of barcode), that is widely used in North America, and in countries including the UK, Australia, and New Zealand for tracking trade items in stores. Its most common form, the UPC-A, consists of 12 numerical digits, which are uniquely assigned to each trade item. Along with the related EAN barcode, the UPC is the only barcode allowed for scanning trade items at the point of sale, per GS1 standards.</xsd:documentation> </xsd:annotation> </xsd:pattern> </xsd:restriction> </xsd:simpleType></pre>

Simple Type isrc

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	pattern $([a-zA-Z]\{2\})(\-\?)$ $[0-9a-zA-Z]\{3\}(\-\?)$ $\d\{2\}(\-\?)\d\{5\}$
Used by	Element ids/isrc
Source	<pre><xsd:simpleType name="isrc"> <xsd:restriction base="xsd:string"> <xsd:pattern value="([a-zA-Z]\{2\})(\-\?)[0-9a-zA-Z]\{3\}(\-\?)\d\{2\}(\-\?)\d\{5\}"> <xsd:annotation> <xsd:documentation xml:lang="en">The International Standard Recording Code (ISRC), defined by ISO 3901, is an international standard code for uniquely identifying sound recordings and music video recordings.</xsd:documentation> </xsd:annotation> </xsd:pattern> </xsd:restriction> </xsd:simpleType></pre>

</xsd:simpleType>

Simple Type finetunes

Namespace	No namespace
Diagram	<p>Built-in derived type. The long datatype is derived from integer by setting the value of maxInclusive to be...</p>
Type	xsd:long
Used by	Element ids/finetunes
Source	<pre><xsd:simpleType name="finetunes"> <xsd:restriction base="xsd:long"> <xsd:annotation> <xsd:documentation xml:lang="en">The 1-13 digits long identifier of a item at finetunes.</xsd:documentation> </xsd:annotation> </xsd:restriction> </xsd:simpleType></pre>

Simple Type contributorType

Namespace	No namespace
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	restriction of xsd:string
Facets	enumeration label enumeration performer enumeration texter enumeration editor enumeration conductor enumeration orchestra enumeration display_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"/></pre>

```

<xsd:enumeration value="conductor" />
<xsd:enumeration value="orchestra" />
<xsd:enumeration value="display_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>

```

Simple Type allowance

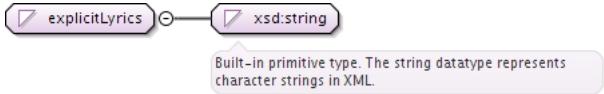
Namespace	No namespace				
Diagram					
Type	restriction of xsd:string				
Facets	<table> <tr> <td>enumeration</td><td>allow</td></tr> <tr> <td>enumeration</td><td>disallow</td></tr> </table>	enumeration	allow	enumeration	disallow
enumeration	allow				
enumeration	disallow				
Used by	Attributes channel/@type, 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

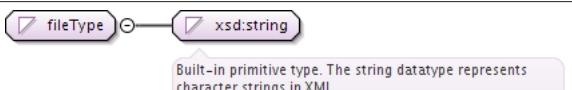
Namespace	No namespace										
Diagram											
Type	restriction of xsd:string										
Facets	<table> <tr> <td>enumeration</td><td>equals</td></tr> <tr> <td>enumeration</td><td>before</td></tr> <tr> <td>enumeration</td><td>after</td></tr> <tr> <td>enumeration</td><td>contains</td></tr> <tr> <td>enumeration</td><td>containedin</td></tr> </table>	enumeration	equals	enumeration	before	enumeration	after	enumeration	contains	enumeration	containedin
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

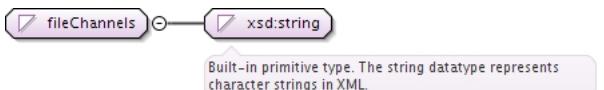
Namespace	No namespace
-----------	--------------

Diagram							
Type	restriction of xsd:string						
Facets	<table> <tr> <td>enumeration</td><td>true</td></tr> <tr> <td>enumeration</td><td>false</td></tr> <tr> <td>enumeration</td><td>cleaned</td></tr> </table>	enumeration	true	enumeration	false	enumeration	cleaned
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"/></pre>								

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

Simple Type isbn

Namespace	No namespace
Diagram	
Type	restriction of xsd:string
Facets	<p>pattern</p> $(\backslash d\{1\}-\backslash d\{5\}-\backslash d\{3\}-\backslash d\{1\} \backslash d\{1\}-\backslash d\{3\}-\backslash d\{5\}-\backslash d\{1\} \backslash d\{1\}-\backslash d\{2\}-\backslash d\{6\}-\backslash d\{1\})?$
Source	<pre><xsd:simpleType name="isbn"> <xsd:restriction base="xsd:string"> <xsd:pattern value="(\backslash d\{1\}-\backslash d\{5\}-\backslash d\{3\}-\backslash d\{1\} \backslash d\{1\}-\backslash d\{3\}-\backslash d\{5\}-\backslash d\{1\} \backslash d\{1\}-\backslash d\{2\}-\backslash d\{6\}-\backslash d\{1\})?"> <xsd:annotation> <xsd:documentation xml:lang="en">The International Standard Book Number (ISBN) is a unique numeric commercial book identifier based upon the 9-digit Standard Book Numbering (SBN) code.</xsd:documentation> </xsd:annotation> </xsd:pattern> </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	<pre><xsd:attribute name="num" type="xsd:integer" /></pre>

Attribute promotext / @lang

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

Attribute teasertext / @lang

Namespace	No namespace
Type	xsd:string

Properties	content: simple
Used by	Complex Type teasertext
Source	<xsd:attribute name="lang" type="xsd:string" />

Attribute physical_distributor / @publishable

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

Attribute territory / @type

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

Attribute channel / @type

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

Attribute rule / @num

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

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://fnpl.org/opensdx/countrycodes																																																																																																																									
Annotations	This element includes a list of ISO 3166-1 country codes.																																																																																																																									
Diagram	<pre> classDiagram class countryCode { <<ISO 3166-1 country codes>> } class xsdString { <<Built-in primitive type. The string datatype represents character strings in XML.>> } countryCode < -- xsdString </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>AF</td><td>Afghanistan</td></tr> <tr><td>enumeration</td><td>AX</td><td>Åland Islands</td></tr> <tr><td>enumeration</td><td>AL</td><td>Albania</td></tr> <tr><td>enumeration</td><td>DZ</td><td>Algeria</td></tr> <tr><td>enumeration</td><td>AS</td><td>American Samoa</td></tr> <tr><td>enumeration</td><td>AD</td><td>Andorra</td></tr> <tr><td>enumeration</td><td>AO</td><td>Angola</td></tr> <tr><td>enumeration</td><td>AI</td><td>Anguilla</td></tr> <tr><td>enumeration</td><td>AQ</td><td>Antarctica</td></tr> <tr><td>enumeration</td><td>AG</td><td>Antigua and Barbuda</td></tr> <tr><td>enumeration</td><td>AR</td><td>Argentina</td></tr> <tr><td>enumeration</td><td>AM</td><td>Armenia</td></tr> <tr><td>enumeration</td><td>AW</td><td>Aruba</td></tr> <tr><td>enumeration</td><td>AU</td><td>Australia</td></tr> <tr><td>enumeration</td><td>AT</td><td>Austria</td></tr> <tr><td>enumeration</td><td>AZ</td><td>Azerbaijan</td></tr> <tr><td>enumeration</td><td>BS</td><td>Bahamas</td></tr> <tr><td>enumeration</td><td>BH</td><td>Bahrain</td></tr> <tr><td>enumeration</td><td>BD</td><td>Bangladesh</td></tr> <tr><td>enumeration</td><td>BB</td><td>Barbados</td></tr> <tr><td>enumeration</td><td>BY</td><td>Belarus</td></tr> <tr><td>enumeration</td><td>BE</td><td>Belgium</td></tr> <tr><td>enumeration</td><td>BZ</td><td>Belize</td></tr> <tr><td>enumeration</td><td>BJ</td><td>Benin</td></tr> <tr><td>enumeration</td><td>BM</td><td>Bermuda</td></tr> <tr><td>enumeration</td><td>BT</td><td>Bhutan</td></tr> <tr><td>enumeration</td><td>BO</td><td>Bolivia, Plurinational State of</td></tr> <tr><td>enumeration</td><td>BQ</td><td>Bonaire, Sint Eustatius and Saba</td></tr> <tr><td>enumeration</td><td>BA</td><td>Bosnia and Herzegovina</td></tr> <tr><td>enumeration</td><td>BW</td><td>Botswana</td></tr> <tr><td>enumeration</td><td>BV</td><td>Bouvet Island</td></tr> <tr><td>enumeration</td><td>BR</td><td>Brazil</td></tr> <tr><td>enumeration</td><td>IO</td><td>British Indian Ocean Territory</td></tr> <tr><td>enumeration</td><td>BN</td><td>Brunei Darussalam</td></tr> <tr><td>enumeration</td><td>BG</td><td>Bulgaria</td></tr> <tr><td>enumeration</td><td>BF</td><td>Burkina Faso</td></tr> <tr><td>enumeration</td><td>BI</td><td>Burundi</td></tr> <tr><td>enumeration</td><td>KH</td><td>Cambodia</td></tr> <tr><td>enumeration</td><td>CM</td><td>Cameroon</td></tr> <tr><td>enumeration</td><td>CA</td><td>Canada</td></tr> </table>		enumeration	AF	Afghanistan	enumeration	AX	Åland Islands	enumeration	AL	Albania	enumeration	DZ	Algeria	enumeration	AS	American Samoa	enumeration	AD	Andorra	enumeration	AO	Angola	enumeration	AI	Anguilla	enumeration	AQ	Antarctica	enumeration	AG	Antigua and Barbuda	enumeration	AR	Argentina	enumeration	AM	Armenia	enumeration	AW	Aruba	enumeration	AU	Australia	enumeration	AT	Austria	enumeration	AZ	Azerbaijan	enumeration	BS	Bahamas	enumeration	BH	Bahrain	enumeration	BD	Bangladesh	enumeration	BB	Barbados	enumeration	BY	Belarus	enumeration	BE	Belgium	enumeration	BZ	Belize	enumeration	BJ	Benin	enumeration	BM	Bermuda	enumeration	BT	Bhutan	enumeration	BO	Bolivia, Plurinational State of	enumeration	BQ	Bonaire, Sint Eustatius and Saba	enumeration	BA	Bosnia and Herzegovina	enumeration	BW	Botswana	enumeration	BV	Bouvet Island	enumeration	BR	Brazil	enumeration	IO	British Indian Ocean Territory	enumeration	BN	Brunei Darussalam	enumeration	BG	Bulgaria	enumeration	BF	Burkina Faso	enumeration	BI	Burundi	enumeration	KH	Cambodia	enumeration	CM	Cameroon	enumeration	CA	Canada
enumeration	AF	Afghanistan																																																																																																																								
enumeration	AX	Åland Islands																																																																																																																								
enumeration	AL	Albania																																																																																																																								
enumeration	DZ	Algeria																																																																																																																								
enumeration	AS	American Samoa																																																																																																																								
enumeration	AD	Andorra																																																																																																																								
enumeration	AO	Angola																																																																																																																								
enumeration	AI	Anguilla																																																																																																																								
enumeration	AQ	Antarctica																																																																																																																								
enumeration	AG	Antigua and Barbuda																																																																																																																								
enumeration	AR	Argentina																																																																																																																								
enumeration	AM	Armenia																																																																																																																								
enumeration	AW	Aruba																																																																																																																								
enumeration	AU	Australia																																																																																																																								
enumeration	AT	Austria																																																																																																																								
enumeration	AZ	Azerbaijan																																																																																																																								
enumeration	BS	Bahamas																																																																																																																								
enumeration	BH	Bahrain																																																																																																																								
enumeration	BD	Bangladesh																																																																																																																								
enumeration	BB	Barbados																																																																																																																								
enumeration	BY	Belarus																																																																																																																								
enumeration	BE	Belgium																																																																																																																								
enumeration	BZ	Belize																																																																																																																								
enumeration	BJ	Benin																																																																																																																								
enumeration	BM	Bermuda																																																																																																																								
enumeration	BT	Bhutan																																																																																																																								
enumeration	BO	Bolivia, Plurinational State of																																																																																																																								
enumeration	BQ	Bonaire, Sint Eustatius and Saba																																																																																																																								
enumeration	BA	Bosnia and Herzegovina																																																																																																																								
enumeration	BW	Botswana																																																																																																																								
enumeration	BV	Bouvet Island																																																																																																																								
enumeration	BR	Brazil																																																																																																																								
enumeration	IO	British Indian Ocean Territory																																																																																																																								
enumeration	BN	Brunei Darussalam																																																																																																																								
enumeration	BG	Bulgaria																																																																																																																								
enumeration	BF	Burkina Faso																																																																																																																								
enumeration	BI	Burundi																																																																																																																								
enumeration	KH	Cambodia																																																																																																																								
enumeration	CM	Cameroon																																																																																																																								
enumeration	CA	Canada																																																																																																																								

enumeration	CV	Cape Verde
enumeration	KY	Cayman Islands
enumeration	CF	Central African Republic
enumeration	TD	Chad
enumeration	CL	Chile
enumeration	CN	China
enumeration	CX	Christmas Island
enumeration	CC	Cocos (Keeling) Islands
enumeration	CO	Colombia
enumeration	KM	Comoros
enumeration	CG	Congo
enumeration	CD	Congo, the Democratic Republic of the
enumeration	CK	Cook Islands
enumeration	CR	Costa Rica
enumeration	CI	Côte d'Ivoire
enumeration	HR	Croatia
enumeration	CU	Cuba
enumeration	CW	Curaçao
enumeration	CY	Cyprus
enumeration	CZ	Czech Republic
enumeration	DK	Denmark
enumeration	DJ	Djibouti
enumeration	DM	Dominica
enumeration	DO	Dominican Republic
enumeration	EC	Ecuador
enumeration	EG	Egypt
enumeration	SV	El Salvador
enumeration	GQ	Equatorial Guinea
enumeration	ER	Eritrea
enumeration	EE	Estonia
enumeration	ET	Ethiopia
enumeration	FK	Falkland Islands (Malvinas)
enumeration	FO	Faroe Islands
enumeration	FJ	Fiji
enumeration	FI	Finland
enumeration	FR	France
enumeration	GF	French Guiana
enumeration	PF	French Polynesia
enumeration	TF	French Southern Territories
enumeration	GA	Gabon
enumeration	GM	Gambia
enumeration	GE	Georgia
enumeration	DE	Germany
enumeration	GH	Ghana
enumeration	GI	Gibraltar
enumeration	GR	Greece
enumeration	GL	Greenland
enumeration	GD	Grenada
enumeration	GP	Guadeloupe

enumeration	GU	Guam
enumeration	GT	Guatemala
enumeration	GG	Guernsey
enumeration	GN	Guinea
enumeration	GW	Guinea-Bissau
enumeration	GY	Guyana
enumeration	HT	Haiti
enumeration	HM	Heard Island and McDonald Islands
enumeration	VA	Holy See (Vatican City State)
enumeration	HN	Honduras
enumeration	HK	Hong Kong
enumeration	HU	Hungary
enumeration	IS	Iceland
enumeration	IN	India
enumeration	ID	Indonesia
enumeration	IR	Iran, Islamic Republic of
enumeration	IQ	Iraq
enumeration	IE	Ireland
enumeration	IM	Isle of Man
enumeration	IL	Israel
enumeration	IT	Italy
enumeration	JM	Jamaica
enumeration	JP	Japan
enumeration	JE	Jersey
enumeration	JO	Jordan
enumeration	KZ	Kazakhstan
enumeration	KE	Kenya
enumeration	KI	Kiribati
enumeration	KP	Korea, Democratic People's Republic of
enumeration	KR	Korea, Republic of
enumeration	KW	Kuwait
enumeration	KG	Kyrgyzstan
enumeration	LA	Lao People's Democratic Republic
enumeration	LV	Latvia
enumeration	LB	Lebanon
enumeration	LS	Lesotho
enumeration	LR	Liberia
enumeration	LY	Libyan Arab Jamahiriya
enumeration	LI	Liechtenstein
enumeration	LT	Lithuania
enumeration	LU	Luxembourg
enumeration	MO	Macao
enumeration	MK	Macedonia, the former Yugoslav Republic of
enumeration	MG	Madagascar
enumeration	MW	Malawi
enumeration	MY	Malaysia
enumeration	MV	Maldives
enumeration	ML	Mali
enumeration	MT	Malta

enumeration	MH	Marshall Islands
enumeration	MQ	Martinique
enumeration	MR	Mauritania
enumeration	MU	Mauritius
enumeration	YT	Mayotte
enumeration	MX	Mexico
enumeration	FM	Micronesia, Federated States of
enumeration	MD	Moldova, Republic of
enumeration	MC	Monaco
enumeration	MN	Mongolia
enumeration	ME	Montenegro
enumeration	MS	Montserrat
enumeration	MA	Morocco
enumeration	MZ	Mozambique
enumeration	MM	Myanmar
enumeration	NA	Namibia
enumeration	NR	Nauru
enumeration	NP	Nepal
enumeration	NL	Netherlands
enumeration	NC	New Caledonia
enumeration	NZ	New Zealand
enumeration	NI	Nicaragua
enumeration	NE	Niger
enumeration	NG	Nigeria
enumeration	NU	Niue
enumeration	NF	Norfolk Island
enumeration	MP	Northern Mariana Islands
enumeration	NO	Norway
enumeration	OM	Oman
enumeration	PK	Pakistan
enumeration	PW	Palau
enumeration	PS	Palestinian Territory, Occupied
enumeration	PA	Panama
enumeration	PG	Papua New Guinea
enumeration	PY	Paraguay
enumeration	PE	Peru
enumeration	PH	Philippines
enumeration	PN	Pitcairn
enumeration	PL	Poland
enumeration	PT	Portugal
enumeration	PR	Puerto Rico
enumeration	QA	Qatar
enumeration	RE	Réunion
enumeration	RO	Romania
enumeration	RU	Russian Federation
enumeration	RW	Rwanda
enumeration	BL	Saint Barthélemy
enumeration	SH	Saint Helena, Ascension and Tristan da Cunha
enumeration	KN	Saint Kitts and Nevis

enumeration	LC	Saint Lucia
enumeration	MF	Saint Martin (French part)
enumeration	PM	Saint Pierre and Miquelon
enumeration	VC	Saint Vincent and the Grenadines
enumeration	WS	Samoa
enumeration	SM	San Marino
enumeration	ST	Sao Tome and Principe
enumeration	SA	Saudi Arabia
enumeration	SN	Senegal
enumeration	RS	Serbia
enumeration	SC	Seychelles
enumeration	SL	Sierra Leone
enumeration	SG	Singapore
enumeration	SX	Sint Maarten (Dutch part)
enumeration	SK	Slovakia
enumeration	SI	Slovenia
enumeration	SB	Solomon Islands
enumeration	SO	Somalia
enumeration	ZA	South Africa
enumeration	GS	South Georgia and the South Sandwich Islands
enumeration	SS	South Sudan
enumeration	ES	Spain
enumeration	LK	Sri Lanka
enumeration	SD	Sudan
enumeration	SR	Suriname
enumeration	SJ	Svalbard and Jan Mayen
enumeration	SZ	Swaziland
enumeration	SE	Sweden
enumeration	CH	Switzerland
enumeration	SY	Syrian Arab Republic
enumeration	TW	Taiwan, Province of China
enumeration	TJ	Tajikistan
enumeration	TZ	Tanzania, United Republic of
enumeration	TH	Thailand
enumeration	TL	Timor-Leste
enumeration	TG	Togo
enumeration	TK	Tokelau
enumeration	TO	Tonga
enumeration	TT	Trinidad and Tobago
enumeration	TN	Tunisia
enumeration	TR	Turkey
enumeration	TM	Turkmenistan
enumeration	TC	Turks and Caicos Islands
enumeration	TV	Tuvalu
enumeration	UG	Uganda
enumeration	UA	Ukraine
enumeration	AE	United Arab Emirates
enumeration	GB	United Kingdom
enumeration	US	United States

	enumeration	UM	United States Minor Outlying Islands
	enumeration	UY	Uruguay
	enumeration	UZ	Uzbekistan
	enumeration	VU	Vanuatu
	enumeration	VE	Venezuela, Bolivarian Republic of
	enumeration	VN	Viet Nam
	enumeration	VG	Virgin Islands, British
	enumeration	VI	Virgin Islands, U.S.
	enumeration	WF	Wallis and Futuna
	enumeration	WW	WorldWide
	enumeration	EH	Western Sahara
	enumeration	YE	Yemen
	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="AF"> <xsd:annotation> <xsd:documentation>Afghanistan</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="AL"> <xsd:annotation> <xsd:documentation>Albania</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="DZ"> <xsd:annotation> <xsd:documentation>Algeria</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="AD"> <xsd:annotation> <xsd:documentation>Andorra</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AO"> <xsd:annotation> <xsd:documentation>Angola</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AI"> <xsd:annotation> <xsd:documentation>Anguilla</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AQ"> <xsd:annotation> <xsd:documentation>Antarctica</xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="AG"> <xsd:annotation> <xsd:documentation>Antigua and Barbuda</xsd:documentation> </xsd:annotation> </xsd:enumeration> </xsd:restriction> </xsd:simpleType></pre>		

```
<xsd:enumeration value="AR">
  <xsd:annotation>
    <xsd:documentation>Argentina</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AM">
  <xsd:annotation>
    <xsd:documentation>Armenia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AW">
  <xsd:annotation>
    <xsd:documentation>Aruba</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AU">
  <xsd:annotation>
    <xsd:documentation>Australia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AT">
  <xsd:annotation>
    <xsd:documentation>Austria</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="AZ">
  <xsd:annotation>
    <xsd:documentation>Azerbaijan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BS">
  <xsd:annotation>
    <xsd:documentation>Bahamas</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BH">
  <xsd:annotation>
    <xsd:documentation>Bahrain</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BD">
  <xsd:annotation>
    <xsd:documentation>Bangladesh</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BB">
  <xsd:annotation>
    <xsd:documentation>Barbados</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BY">
  <xsd:annotation>
    <xsd:documentation>Belarus</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BE">
  <xsd:annotation>
    <xsd:documentation>Belgium</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BZ">
  <xsd:annotation>
    <xsd:documentation>Belize</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BJ">
  <xsd:annotation>
    <xsd:documentation>Benin</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BM">
  <xsd:annotation>
    <xsd:documentation>Bermuda</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BT">
  <xsd:annotation>
    <xsd:documentation>Bhutan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BO">
  <xsd:annotation>
    <xsd:documentation>Bolivia, Plurinational State of</xsd:documentation>
```

```
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BQ">
  <xsd:annotation>
    <xsd:documentation>Bonaire, Sint Eustatius and Saba</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="BW">
  <xsd:annotation>
    <xsd:documentation>Botswana</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="BR">
  <xsd:annotation>
    <xsd:documentation>Brazil</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="BN">
  <xsd:annotation>
    <xsd:documentation>Brunei Darussalam</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="BG">
  <xsd:annotation>
    <xsd:documentation>Bulgaria</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="BI">
  <xsd:annotation>
    <xsd:documentation>Burundi</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KH">
  <xsd:annotation>
    <xsd:documentation>Cambodia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CM">
  <xsd:annotation>
    <xsd:documentation>Cameroon</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CA">
  <xsd:annotation>
    <xsd:documentation>Canada</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="KY">
  <xsd:annotation>
    <xsd:documentation>Cayman Islands</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="TD">
```

```
<xsd:annotation>
  <xsd:documentation>Chad</xsd:documentation>
</xsd:annotation>
<xsd:enumeration value="CL">
  <xsd:annotation>
    <xsd:documentation>Chile</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CN">
  <xsd:annotation>
    <xsd:documentation>China</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="CC">
  <xsd:annotation>
    <xsd:documentation>Cocos (Keeling) Islands</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CO">
  <xsd:annotation>
    <xsd:documentation>Colombia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KM">
  <xsd:annotation>
    <xsd:documentation>Comoros</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CG">
  <xsd:annotation>
    <xsd:documentation>Congo</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="CK">
  <xsd:annotation>
    <xsd:documentation>Cook Islands</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="CI">
  <xsd:annotation>
    <xsd:documentation>Côte d'Ivoire</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HR">
  <xsd:annotation>
    <xsd:documentation>Croatia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CU">
  <xsd:annotation>
    <xsd:documentation>Cuba</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CW">
  <xsd:annotation>
    <xsd:documentation>Curaçao</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="DK">
  <xsd:annotation>
    <xsd:documentation>Denmark</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DJ">
  <xsd:annotation>
    <xsd:documentation>Djibouti</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="EC">
  <xsd:annotation>
    <xsd:documentation>Ecuador</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="EG">
  <xsd:annotation>
    <xsd:documentation>Egypt</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="GQ">
  <xsd:annotation>
    <xsd:documentation>Equatorial Guinea</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ER">
  <xsd:annotation>
    <xsd:documentation>Eritrea</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="EE">
  <xsd:annotation>
    <xsd:documentation>Estonia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ET">
  <xsd:annotation>
    <xsd:documentation>Ethiopia</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="FO">
  <xsd:annotation>
    <xsd:documentation>Faroe Islands</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FJ">
  <xsd:annotation>
    <xsd:documentation>Fiji</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FI">
  <xsd:annotation>
    <xsd:documentation>Finland</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="FR">
  <xsd:annotation>
    <xsd:documentation>France</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="PF">
  <xsd:annotation>
    <xsd:documentation>French Polynesia</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="GA">
  <xsd:annotation>
    <xsd:documentation>Gabon</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GM">
  <xsd:annotation>
    <xsd:documentation>Gambia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GE">
  <xsd:annotation>
    <xsd:documentation>Georgia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="DE">
  <xsd:annotation>
    <xsd:documentation>Germany</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="GR">
  <xsd:annotation>
    <xsd:documentation>Greece</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GL">
  <xsd:annotation>
    <xsd:documentation>Greenland</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GD">
  <xsd:annotation>
    <xsd:documentation>Grenada</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GP">
  <xsd:annotation>
    <xsd:documentation>Guadeloupe</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GU">
  <xsd:annotation>
    <xsd:documentation>Guam</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GT">
  <xsd:annotation>
    <xsd:documentation>Guatemala</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GG">
  <xsd:annotation>
    <xsd:documentation>Guernsey</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="GN">
  <xsd:annotation>
    <xsd:documentation>Guinea</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="HT">
  <xsd:annotation>
    <xsd:documentation>Haiti</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="VA">
  <xsd:annotation>
    <xsd:documentation>Holy See (Vatican City State)</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="HN">
  <xsd:annotation>
    <xsd:documentation>Honduras</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="HU">
  <xsd:annotation>
    <xsd:documentation>Hungary</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IS">
  <xsd:annotation>
    <xsd:documentation>Iceland</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IN">
  <xsd:annotation>
    <xsd:documentation>India</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ID">
  <xsd:annotation>
    <xsd:documentation>Indonesia</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="IQ">
  <xsd:annotation>
    <xsd:documentation>Iraq</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IE">
  <xsd:annotation>
    <xsd:documentation>Ireland</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="IL">
  <xsd:annotation>
    <xsd:documentation>Israel</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="IT">
  <xsd:annotation>
    <xsd:documentation>Italy</xsd:documentation>
```

```
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="JM">
  <xsd:annotation>
    <xsd:documentation>Jamaica</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="JP">
  <xsd:annotation>
    <xsd:documentation>Japan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="JE">
  <xsd:annotation>
    <xsd:documentation>Jersey</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="JO">
  <xsd:annotation>
    <xsd:documentation>Jordan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KZ">
  <xsd:annotation>
    <xsd:documentation>Kazakhstan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KE">
  <xsd:annotation>
    <xsd:documentation>Kenya</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="KI">
  <xsd:annotation>
    <xsd:documentation>Kiribati</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="KG">
  <xsd:annotation>
    <xsd:documentation>Kyrgyzstan</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="LV">
  <xsd:annotation>
    <xsd:documentation>Latvia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LB">
  <xsd:annotation>
    <xsd:documentation>Lebanon</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LS">
  <xsd:annotation>
    <xsd:documentation>Lesotho</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="LR">
  <xsd:annotation>
    <xsd:documentation>Liberia</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="LI">
  <xsd:annotation>
    <xsd:documentation>Liechtenstein</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="MO">
  <xsd:annotation>
    <xsd:documentation>Macao</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="MG">
  <xsd:annotation>
    <xsd:documentation>Madagascar</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MW">
  <xsd:annotation>
    <xsd:documentation>Malawi</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MY">
  <xsd:annotation>
    <xsd:documentation>Malaysia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MV">
  <xsd:annotation>
    <xsd:documentation>Maldives</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ML">
  <xsd:annotation>
    <xsd:documentation>Mali</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MT">
  <xsd:annotation>
    <xsd:documentation>Malta</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="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="MU">
  <xsd:annotation>
    <xsd:documentation>Mauritius</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="YT">
  <xsd:annotation>
    <xsd:documentation>Mayotte</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
```

```
</xsd:enumeration>
<xsd:enumeration value="MX">
  <xsd:annotation>
    <xsd:documentation>Mexico</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="MD">
  <xsd:annotation>
    <xsd:documentation>Moldova, Republic of</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MC">
  <xsd:annotation>
    <xsd:documentation>Monaco</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MN">
  <xsd:annotation>
    <xsd:documentation>Mongolia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ME">
  <xsd:annotation>
    <xsd:documentation>Montenegro</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MS">
  <xsd:annotation>
    <xsd:documentation>Montserrat</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MA">
  <xsd:annotation>
    <xsd:documentation>Morocco</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MZ">
  <xsd:annotation>
    <xsd:documentation>Mozambique</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="MM">
  <xsd:annotation>
    <xsd:documentation>Myanmar</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NA">
  <xsd:annotation>
    <xsd:documentation>Namibia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NR">
  <xsd:annotation>
    <xsd:documentation>Nauru</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NP">
  <xsd:annotation>
    <xsd:documentation>Nepal</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NL">
  <xsd:annotation>
    <xsd:documentation>Netherlands</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="NZ">
  <xsd:annotation>
    <xsd:documentation>New Zealand</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NI">
  <xsd:annotation>
```

```
<xsd:documentation>Nicaragua</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NE">
  <xsd:annotation>
    <xsd:documentation>Niger</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NG">
  <xsd:annotation>
    <xsd:documentation>Nigeria</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NU">
  <xsd:annotation>
    <xsd:documentation>Niue</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="MP">
  <xsd:annotation>
    <xsd:documentation>Northern Mariana Islands</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NO">
  <xsd:annotation>
    <xsd:documentation>Norway</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="OM">
  <xsd:annotation>
    <xsd:documentation>Oman</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PK">
  <xsd:annotation>
    <xsd:documentation>Pakistan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PW">
  <xsd:annotation>
    <xsd:documentation>Palau</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="PA">
  <xsd:annotation>
    <xsd:documentation>Panama</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="PY">
  <xsd:annotation>
    <xsd:documentation>Paraguay</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PE">
  <xsd:annotation>
    <xsd:documentation>Peru</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PH">
  <xsd:annotation>
    <xsd:documentation>Philippines</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PN">
  <xsd:annotation>
    <xsd:documentation>Pitcairn</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
```

```
<xsd:enumeration value="PL">
  <xsd:annotation>
    <xsd:documentation>Poland</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="PT">
  <xsd:annotation>
    <xsd:documentation>Portugal</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="QA">
  <xsd:annotation>
    <xsd:documentation>Qatar</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="RE">
  <xsd:annotation>
    <xsd:documentation>Réunion</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="RO">
  <xsd:annotation>
    <xsd:documentation>Romania</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="BL">
  <xsd:annotation>
    <xsd:documentation>Saint Barthélemy</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SH">
  <xsd:annotation>
    <xsd:documentation>Saint Helena, Ascension and Tristan da Cunha</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="LC">
  <xsd:annotation>
    <xsd:documentation>Saint Lucia</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="PM">
  <xsd:annotation>
    <xsd:documentation>Saint Pierre and Miquelon</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="WS">
  <xsd:annotation>
    <xsd:documentation>Samoa</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="ST">
  <xsd:annotation>
    <xsd:documentation>Sao Tome and Principe</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="SN">
  <xsd:annotation>
    <xsd:documentation>Senegal</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="RS">
  <xsd:annotation>
    <xsd:documentation>Serbia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SC">
  <xsd:annotation>
    <xsd:documentation>Seychelles</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="SG">
  <xsd:annotation>
    <xsd:documentation>Singapore</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SX">
  <xsd:annotation>
    <xsd:documentation>Sint Maarten (Dutch part)</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SK">
  <xsd:annotation>
    <xsd:documentation>Slovakia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SI">
  <xsd:annotation>
    <xsd:documentation>Slovenia</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="SO">
  <xsd:annotation>
    <xsd:documentation>Somalia</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="GS">
  <xsd:annotation>
    <xsd:documentation>South Georgia and the South Sandwich Islands</
xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SS">
  <xsd:annotation>
    <xsd:documentation>South Sudan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="ES">
  <xsd:annotation>
    <xsd:documentation>Spain</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="SD">
  <xsd:annotation>
    <xsd:documentation>Sudan</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SR">
  <xsd:annotation>
    <xsd:documentation>Suriname</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="SZ">
  <xsd:annotation>
    <xsd:documentation>Swaziland</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="SE">
  <xsd:annotation>
    <xsd:documentation>Sweden</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="CH">
  <xsd:annotation>
    <xsd:documentation>Switzerland</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="TW">
  <xsd:annotation>
    <xsd:documentation>Taiwan, Province of China</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TJ">
  <xsd:annotation>
    <xsd:documentation>Tajikistan</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="TH">
  <xsd:annotation>
    <xsd:documentation>Thailand</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="TG">
  <xsd:annotation>
    <xsd:documentation>Togo</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TK">
  <xsd:annotation>
    <xsd:documentation>Tokelau</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TO">
  <xsd:annotation>
    <xsd:documentation>Tonga</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="TN">
  <xsd:annotation>
    <xsd:documentation>Tunisia</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TR">
  <xsd:annotation>
    <xsd:documentation>Turkey</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="TM">
  <xsd:annotation>
    <xsd:documentation>Turkmenistan</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="TV">
  <xsd:annotation>
    <xsd:documentation>Tuvalu</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="UG">
  <xsd:annotation>
    <xsd:documentation>Uganda</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="UA">
  <xsd:annotation>
    <xsd:documentation>Ukraine</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="GB">
  <xsd:annotation>
    <xsd:documentation>United Kingdom</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="UM">
  <xsd:annotation>
    <xsd:documentation>United States Minor Outlying Islands</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="VU">
  <xsd:annotation>
    <xsd:documentation>Vanuatu</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="VE">
  <xsd:annotation>
    <xsd:documentation>Venezuela, Bolivarian Republic of</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="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="WF">
  <xsd:annotation>
    <xsd:documentation>Wallis and Futuna</xsd:documentation>
  </xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="WW">
  <xsd:annotation>
    <xsd:documentation>WorldWide</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="YE">
  <xsd:annotation>
    <xsd:documentation>Yemen</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 openSDX-genres.												
Diagram	<p>This element includes a list of openSDX-genres.</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> </table>	enumeration	Rock	enumeration	Beat	enumeration	Blues Rock	enumeration	Rock "n" Roll	enumeration	Art Rock	enumeration	Classic Rock
enumeration	Rock												
enumeration	Beat												
enumeration	Blues Rock												
enumeration	Rock "n" Roll												
enumeration	Art Rock												
enumeration	Classic Rock												

enumeration	Deutschrock
enumeration	Emo
enumeration	Experimental Rock
enumeration	Glam Rock
enumeration	Hard Rock
enumeration	Krautrock
enumeration	Progressive Rock
enumeration	Psychedelic Rock
enumeration	Psychobilly Rock
enumeration	Rockabilly
enumeration	Soft Rock
enumeration	Southern Rock
enumeration	Surf Rock
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	Funpunk
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	Britpop
enumeration	Dance Pop
enumeration	Deutschpop
enumeration	Disco
enumeration	Easy Listening
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	New Age
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	German Folk / Volksmusik
enumeration	Jazz
enumeration	Acid Jazz
enumeration	Avantgarde
enumeration	Bebop
enumeration	Big Band
enumeration	Classic Jazz
enumeration	Cool Jazz
enumeration	Dixieland music
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	Hip Hop
enumeration	Alternative Hip Hop
enumeration	Crunk
enumeration	Dirty South
enumeration	G-Funk
enumeration	Gangsta Rap
enumeration	Golden Era
enumeration	Grime
enumeration	Hyphy
enumeration	Instrumental Hip Hop
enumeration	Miami Bass

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	Doo-wop
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	Glitch hop
enumeration	House
enumeration	Acid House
enumeration	Deep House
enumeration	Disco House
enumeration	Electro House
enumeration	Fidget House
enumeration	Hard House
enumeration	Progressive House
enumeration	Soulful House
enumeration	Tech House
enumeration	Tribal
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	Ancient music
enumeration	Medieval music
enumeration	Renaissance
enumeration	Baroque
enumeration	Classical period
enumeration	Romantic
enumeration	Neoromanticism
enumeration	Neoclassicism
enumeration	New Music / Contemporary Music
enumeration	Modern, 20th / 21st Century
enumeration	Postmodern Music
enumeration	Music and other Media / Arts
enumeration	Music and Word
enumeration	12-Tone Composition
enumeration	Anthem
enumeration	Ballet
enumeration	Cantata
enumeration	Chamber Music
enumeration	Choral
enumeration	Crossover / Popular Classicism
enumeration	Electronic Music / Computer Music
enumeration	Madrigal
enumeration	March
enumeration	Minimal Music
enumeration	Motet
enumeration	Musical
enumeration	Opera Arias
enumeration	Opera Baroque
enumeration	Opera Classical
enumeration	Opera Renaissance
enumeration	Opera Romantic
enumeration	Operetta
enumeration	Oratorio
enumeration	Passion
enumeration	Requiem
enumeration	Serialism

enumeration	Sonata
enumeration	Suite
enumeration	Symphonic Music / Orchestral Music
enumeration	Symphony
enumeration	Waltz
enumeration	Brass Ensemble
enumeration	Concerto / Solo Instrument with Orchestra
enumeration	Mixed Ensemble (Strings / Wind)
enumeration	Mixed Wind Ensemble (Woodwind / Brass)
enumeration	Several Solo Instruments
enumeration	Solo Instrument
enumeration	String Ensemble
enumeration	String Orchestra
enumeration	String Quartet
enumeration	String Trio
enumeration	Woodwind Ensemble
enumeration	A cappella
enumeration	Vocal Ensemble
enumeration	Vocal Music
enumeration	Choir
enumeration	Boy's Choir
enumeration	Children's Choir
enumeration	Choir with Orchestra
enumeration	Women's Choir
enumeration	Men's Choir
enumeration	Mixed Choir
enumeration	Soprano
enumeration	Mezzosoprano
enumeration	Alto
enumeration	Tenor
enumeration	Baritone
enumeration	Bass
enumeration	Accordion
enumeration	Ancient Instruments
enumeration	Bassoon
enumeration	Cembalo
enumeration	Clarinet
enumeration	Double Bass
enumeration	Flute
enumeration	Guitar
enumeration	Harp
enumeration	Harpsichord
enumeration	Horn
enumeration	Lute
enumeration	Mandolin

enumeration	Oboe
enumeration	Organ
enumeration	Percussion (Vibraphone etc.)
enumeration	Piano
enumeration	Recorder / English Flute
enumeration	Saxophone
enumeration	Trombone
enumeration	Trumpet
enumeration	Tuba
enumeration	Viola
enumeration	Violin
enumeration	Violoncello
enumeration	Miscellaneous Lead Instrument
enumeration	Reggae
enumeration	Contemporary Reggae
enumeration	Dancehall
enumeration	Dub
enumeration	Lover's Rock
enumeration	Reggaeton
enumeration	Roots
enumeration	Ska
enumeration	World
enumeration	African Music
enumeration	Afro Beat
enumeration	Afro Pop
enumeration	Asian Music
enumeration	Austropop
enumeration	Calypso
enumeration	Caribbean Music
enumeration	Celtic Music
enumeration	Chanson
enumeration	Coupé Decalé
enumeration	Enka
enumeration	European Music
enumeration	Ghazal
enumeration	Griot
enumeration	Gypsy
enumeration	Highlife
enumeration	Judaica Music / Yiddish / Klezmer
enumeration	Kuduro
enumeration	Kwaito
enumeration	Makossa
enumeration	Marching Band
enumeration	Mento
enumeration	Middle Eastern Music
enumeration	Nordic / Scandinavia

enumeration	North American Music
enumeration	South American Music
enumeration	Parang
enumeration	Polka
enumeration	Rai
enumeration	Soca
enumeration	Soukous
enumeration	Zouk
enumeration	Zulu
enumeration	Latin
enumeration	Bachata
enumeration	Banda
enumeration	Bhangra
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	Gregorian Music
enumeration	Hymn
enumeration	Mass
enumeration	Spiritual
enumeration	Worship
enumeration	Miscellaneous
enumeration	Anime / Video Game Soundtracks
enumeration	Bollywood
enumeration	Instrumental
enumeration	Vocal

enumeration	Acoustic
enumeration	Unplugged
enumeration	Live
enumeration	Traditional
enumeration	Karaoke
enumeration	Movie Scores
enumeration	Movie Soundtracks
enumeration	Sound Effects
enumeration	Soundtrack
enumeration	TV Soundtrack
enumeration	Wedding Music
enumeration	Holiday
enumeration	Mashup
enumeration	unclassifiable
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	Popular Characters
enumeration	Animal Stories
enumeration	Children's Book Classics
enumeration	Children's 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	Esoteric
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	Literature
enumeration	Novels
enumeration	Erotica
enumeration	Romance
enumeration	Contemporary Literature
enumeration	Contemporary German Literature
enumeration	Entertainment
enumeration	Youth
enumeration	Youth Detective Stories
enumeration	Fantasy
enumeration	For Girls
enumeration	Knowledge for Teenagers
enumeration	Mystery
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	Afrikaans
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	Time
enumeration	Middle Ages
enumeration	20"s
enumeration	30"s
enumeration	40"s
enumeration	50"s

enumeration	60 "s
enumeration	70 "
enumeration	80 "s
enumeration	90 "s
enumeration	2000 "s
enumeration	2010 "s
enumeration	2020 "s
enumeration	Adult
enumeration	Children
enumeration	Age: up to 6 years
enumeration	Age: 6 years +
enumeration	Age: 8 years +
enumeration	Kids & Family
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	South Sudan (SS)
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 openSDX-genres.</ <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="Deutschrock"/> <xsd:enumeration value="Emo"/> <xsd:enumeration value="Experimental Rock"/> <xsd:enumeration value="Glam Rock"/> <xsd:enumeration value="Hard Rock"/> <xsd:enumeration value="Krautrock"/> <xsd:enumeration value="Progressive Rock"/> <xsd:enumeration value="Psychedelic Rock"/> <xsd:enumeration value="Psychobilly Rock"/> <xsd:enumeration value="Rockabilly"/> <xsd:enumeration value="Soft Rock"/> <xsd:enumeration value="Southern Rock"/> <xsd:enumeration value="Surf Rock"/> <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="Funpunk"/> <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="Britpop"/> <xsd:enumeration value="Dance Pop"/> <xsd:enumeration value="Deutschpop"/> <xsd:enumeration value="Disco"/> <xsd:enumeration value="Easy Listening"/> <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="New Age"/> <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"/> <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"/></pre>

```
<xsd:enumeration value="Irish Folk"/>
<xsd:enumeration value="German Folk / Volksmusik"/>
<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="Classic Jazz"/>
<xsd:enumeration value="Cool Jazz"/>
<xsd:enumeration value="Dixieland music"/>
<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="Hip Hop"/>
<xsd:enumeration value="Alternative Hip Hop"/>
<xsd:enumeration value="Crunk"/>
<xsd:enumeration value="Dirty South"/>
<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 Hip Hop"/>
<xsd:enumeration value="Miami Bass"/>
<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="Doo-wop"/>
<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="Glitch hop"/>
<xsd:enumeration value="House"/>
<xsd:enumeration value="Acid House"/>
<xsd:enumeration value="Deep House"/>
<xsd:enumeration value="Disco House"/>
<xsd:enumeration value="Electro House"/>
<xsd:enumeration value="Fidget House"/>
<xsd:enumeration value="Hard House"/>
<xsd:enumeration value="Progressive House"/>
<xsd:enumeration value="Soulful House"/>
<xsd:enumeration value="Tech House"/>
<xsd:enumeration value="Tribal"/>
<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="Ancient music"/>
<xsd:enumeration value="Medieval music"/>
<xsd:enumeration value="Renaissance"/>
<xsd:enumeration value="Baroque"/>
<xsd:enumeration value="Classical period"/>
<xsd:enumeration value="Romantic"/>
<xsd:enumeration value="Neoromanticism"/>
<xsd:enumeration value="Neoclassicism"/>
<xsd:enumeration value="New Music / Contemporary Music"/>
<xsd:enumeration value="Modern, 20th / 21st Century"/>
<xsd:enumeration value="Postmodern Music"/>
<xsd:enumeration value="Music and other Media / Arts"/>
<xsd:enumeration value="Music and Word"/>
<xsd:enumeration value="12-Tone Composition"/>
<xsd:enumeration value="Anthem"/>
<xsd:enumeration value="Ballet"/>
<xsd:enumeration value="Cantata"/>
<xsd:enumeration value="Chamber Music"/>
<xsd:enumeration value="Choral"/>
<xsd:enumeration value="Crossover / Popular Classicism"/>
<xsd:enumeration value="Electronic Music / Computer Music"/>
<xsd:enumeration value="Madrigal"/>
<xsd:enumeration value="March"/>
<xsd:enumeration value="Minimal Music"/>
<xsd:enumeration value="Motet"/>
<xsd:enumeration value="Musical"/>
<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="Operetta"/>
<xsd:enumeration value="Oratorio"/>
<xsd:enumeration value="Passion"/>
<xsd:enumeration value="Requiem"/>
<xsd:enumeration value="Serialism"/>
<xsd:enumeration value="Sonata"/>
<xsd:enumeration value="Suite"/>
<xsd:enumeration value="Symphonic Music / Orchestral Music"/>
<xsd:enumeration value="Symphony"/>
<xsd:enumeration value="Waltz"/>
<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="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="Woodwind Ensemble"/>
<xsd:enumeration value="A cappella"/>
<xsd:enumeration value="Vocal Ensemble"/>
<xsd:enumeration value="Vocal Music"/>
<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="Women's Choir"/>
<xsd:enumeration value="Men's Choir"/>
<xsd:enumeration value="Mixed Choir"/>
<xsd:enumeration value="Soprano"/>
<xsd:enumeration value="Mezzosoprano"/>
<xsd:enumeration value="Alto"/>
<xsd:enumeration value="Tenor"/>
<xsd:enumeration value="Baritone"/>
<xsd:enumeration value="Bass"/>
<xsd:enumeration value="Accordion"/>
<xsd:enumeration value="Ancient Instruments"/>
<xsd:enumeration value="Bassoon"/>
<xsd:enumeration value="Cembalo"/>
<xsd:enumeration value="Clarinet"/>
<xsd:enumeration value="Double Bass"/>
<xsd:enumeration value="Flute"/>
<xsd:enumeration value="Guitar"/>
<xsd:enumeration value="Harp"/>
<xsd:enumeration value="Harpsichord"/>
```

```
<xsd:enumeration value="Horn"/>
<xsd:enumeration value="Lute"/>
<xsd:enumeration value="Mandolin"/>
<xsd:enumeration value="Oboe"/>
<xsd:enumeration value="Organ"/>
<xsd:enumeration value="Percussion (Vibraphone etc.)"/>
<xsd:enumeration value="Piano"/>
<xsd:enumeration value="Recorder / English Flute"/>
<xsd:enumeration value="Saxophone"/>
<xsd:enumeration value="Trombone"/>
<xsd:enumeration value="Trumpet"/>
<xsd:enumeration value="Tuba"/>
<xsd:enumeration value="Viola"/>
<xsd:enumeration value="Violin"/>
<xsd:enumeration value="Violoncello"/>
<xsd:enumeration value="Miscellaneous Lead Instrument"/>
<xsd:enumeration value="Reggae"/>
<xsd:enumeration value="Contemporary Reggae"/>
<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="Afro Beat"/>
<xsd:enumeration value="Afro Pop"/>
<xsd:enumeration value="Asian Music"/>
<xsd:enumeration value="Austropop"/>
<xsd:enumeration value="Calypso"/>
<xsd:enumeration value="Caribbean Music"/>
<xsd:enumeration value="Celtic Music"/>
<xsd:enumeration value="Chanson"/>
<xsd:enumeration value="Coupé Decalé"/>
<xsd:enumeration value="Enka"/>
<xsd:enumeration value="European Music"/>
<xsd:enumeration value="Ghazal"/>
<xsd:enumeration value="Griot"/>
<xsd:enumeration value="Gypsy"/>
<xsd:enumeration value="Highlife"/>
<xsd:enumeration value="Judaica Music / Yiddish / Klezmer"/>
<xsd:enumeration value="Kuduro"/>
<xsd:enumeration value="Kwaito"/>
<xsd:enumeration value="Makossa"/>
<xsd:enumeration value="Marching Band"/>
<xsd:enumeration value="Mento"/>
<xsd:enumeration value="Middle Eastern Music"/>
<xsd:enumeration value="Nordic / Scandinavia"/>
<xsd:enumeration value="North American Music"/>
<xsd:enumeration value="South 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="Zulu"/>
<xsd:enumeration value="Latin"/>
<xsd:enumeration value="Bachata"/>
<xsd:enumeration value="Banda"/>
<xsd:enumeration value="Bhangra"/>
<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="Gregorian Music"/>
<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="Vocal"/>
<xsd:enumeration value="Acoustic"/>
<xsd:enumeration value="Unplugged"/>
<xsd:enumeration value="Live"/>
<xsd:enumeration value="Traditional"/>
<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="Wedding Music"/>
<xsd:enumeration value="Holiday"/>
<xsd:enumeration value="Mashup"/>
<xsd:enumeration value="unclassifiable"/>
<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="Popular Characters"/>
<xsd:enumeration value="Animal Stories"/>
<xsd:enumeration value="Children's Book Classics"/>
<xsd:enumeration value="Children's 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="Esoteric"/>
<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="Literature" />
<xsd:enumeration value="Novels" />
<xsd:enumeration value="Erotica" />
<xsd:enumeration value="Romance" />
<xsd:enumeration value="Contemporary Literature" />
<xsd:enumeration value="Contemporary German Literature" />
<xsd:enumeration value="Entertainment" />
<xsd:enumeration value="Youth" />
<xsd:enumeration value="Youth Detective Stories" />
<xsd:enumeration value="Fantasy" />
<xsd:enumeration value="For Girls" />
<xsd:enumeration value="Knowledge for Teenagers" />
<xsd:enumeration value="Mystery" />
<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="Afrikaans" />
<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="Time"/>
<xsd:enumeration value="Middle Ages"/>
<xsd:enumeration value="20"s"/>
<xsd:enumeration value="30"s"/>
<xsd:enumeration value="40"s"/>
<xsd:enumeration value="50"s"/>
<xsd:enumeration value="60"s"/>
<xsd:enumeration value="70"s"/>
<xsd:enumeration value="80"s"/>
<xsd:enumeration value="90"s"/>
<xsd:enumeration value="2000"s"/>
<xsd:enumeration value="2010"s"/>
<xsd:enumeration value="2020"s"/>
<xsd:enumeration value="Adult"/>
<xsd:enumeration value="Children"/>
<xsd:enumeration value="Age: up to 6 years"/>
<xsd:enumeration value="Age: 6 years +"/>
<xsd:enumeration value="Age: 8 years +"/>
<xsd:enumeration value="Kids & Family"/>
<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="South Sudan (SS)" />
<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	<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> </table>	enumeration	aa	Afar	enumeration	ab	Abkhazian	enumeration	af	Afrikaans	enumeration	am	Amharic
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	Sinhalese
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></pre>		

```
<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: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>Faroeese</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>
```