SMPTE EG 2021-4:2015
Revision of
SMPTE EG 2021-4:2013

SMPTE ENGINEERING GUIDELINE

Broadcast Exchange Format (BXF) — Schema Documentation

Approved
October 9, 2015

Table of Contents

Page

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE EG 2021-4 was prepared by Technology Committee 34CS.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document

Broadcast Exchange Format (BXF) is a protocol for exchange of data among broadcast systems such as Traffic, Program Management, Automation, and Content Distribution. It is intended to facilitate the movement of content and its associated metadata for better management, coordination and reporting between these broadcast systems. The BXF Protocol serves as a replacement for the many proprietary interfaces in place today between vendors in these areas.

Unless otherwise specified the order of precedence of the types of normative information in this document shall be as follows. Normative prose shall be the authoritative definition. Tables shall be next, followed by formal languages, then figures, and then any other language forms. In the event of a conflict between the schema and other information in this document, the schema is authoritative.

SMPTE 2021 (BXF) is now broken into several parts. A brief outline of the parts can be found in SMPTE OV 2021-0, the Document Roadmap to this suite of documents.

As of BXF 3.0, many schema changes have taken place. Those changes are described under the appropriate schema sections in this document.

1 Scope

The Broadcast eXchange Format (BXF) defines the format and content of XML Messages for the interchange of data and metadata among professional systems, as follows:

1. Broadcast schedules, including playout and record schedules
2. As run information
3. Content metadata, such as Content ID, Title, Duration, etc.
4. Content management requests such as dub and purge requests
5. Requests for transfer of content some of which will result in the transfer of Content essence between professional systems.
6. Ports as used by TCP/IP for the exchange of messages

The primary systems envisioned as users of this standard are:

Program Management Systems
Broadcast Traffic Systems
Master Control Automation Systems
Content Distribution Systems

This particular document focuses on documentation of the BXF schemas.

2 Description of BXF Schema (Informative)

This documentation does not contain the actual schema, but instead provides a reference to the individual files that can be viewed with any number of tools, including easily accessible browsers such as Microsft's Windows® Internet Explorer®, Apple's Safari, and other third party tools. In addition to the XML Schema Files (.XSD) the user can also browse the schema using the .html files.

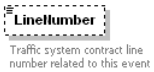
Depending on the tools you use, different parts of the schema may appear in different formatting. The section below describes the meaning of various parts of the schema as you would see them in the .html version.

3 Schema Documention

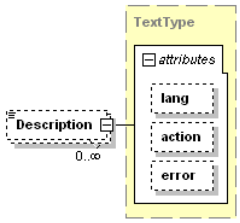
A number of graphics and symbols are used in the documentation to help describe the various elements of the schema and how the elements are related. These represent components and the relationships between schema components. The different components are represented by the following:

Single element – mandatory: Indicated by a rectangle with a solid border. The element name is inside the shape.

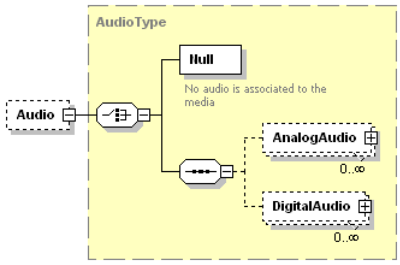
Single element – optional: Indicated by a rectangle with a dashed border. The element name is inside the shape.



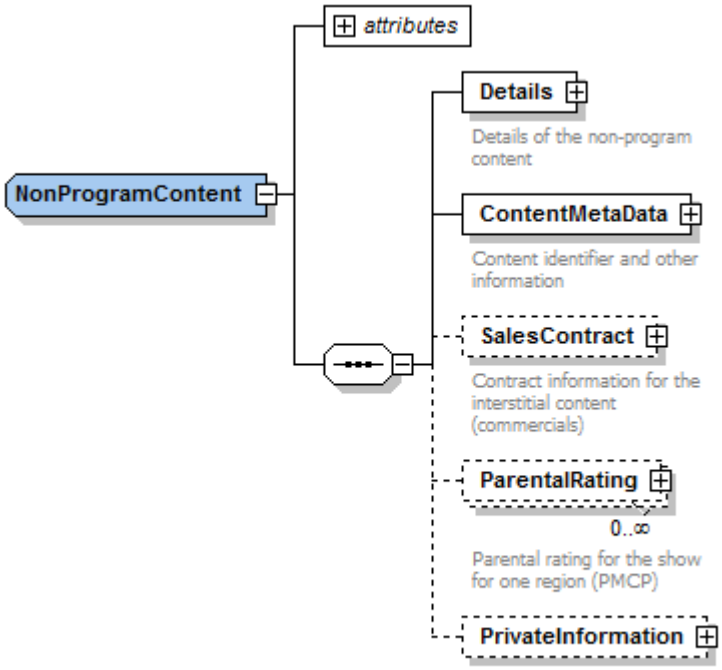
Multiple elements: Indicated by a rectangle with a solid border with a number range representing the minimum and maximum number of occurrences possible. In the example, 0 to infinity (0..∞) is shown.



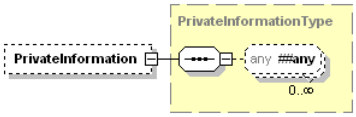
Elements containing child elements: Indicated by a [+] or [-] on the element, representing an element containing additional attributes or elements. The [+] indicates that additional elements are available for display. The [-] indicates that the child elements are displayed.



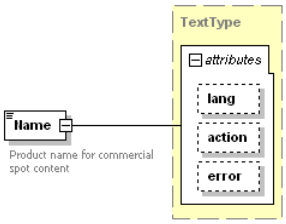
Complex type: Indicated by a partial hexagon and a child element symbol.



Wildcards: Indicated by an octagon with any at the left.

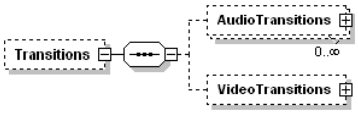


Attributes: Indicated by the word ‘attributes’

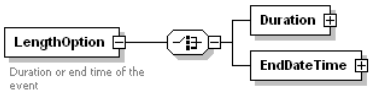


The relationships between components are represented by symbols for sequence and choice.

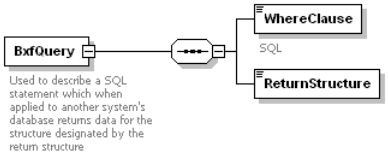
Sequence: The sequence compositor shows that all elements occur in sequence.



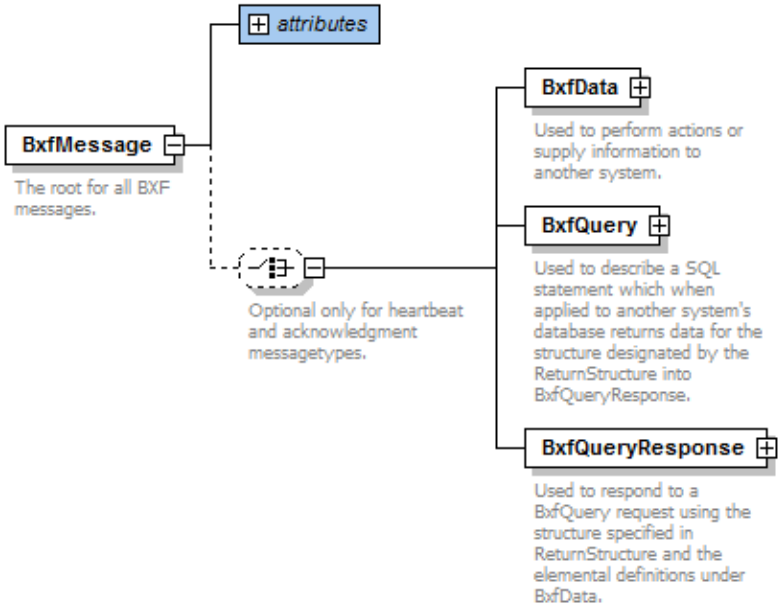
Choice: The choice compositor shows the ‘or’ relationship between associated components (only one choice may be made).



Solid lines: Solid lines connecting elements represent mandatory connections within the schema diagrams.

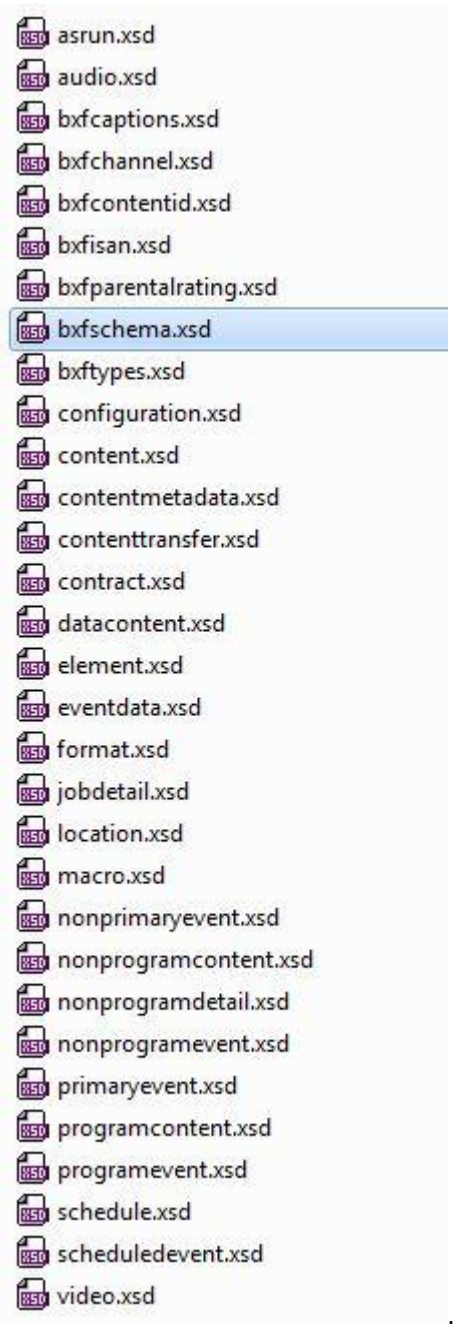


Dashed lines: Dashed lines between elements represent optional connections within the schema diagrams.



3.1 Schema File List

The Schema is actually composed of 31 files listed below. The root file (starting point) is bxfshema.xsd and all other files are referenced from this file. A brief description of each file is given below in alphabetical order. This same description may also be contained in the schema for major elements, but some elements do not contain a description and were segregated into separate files for the convenience of managing the schema.



3.2 Header Change in BXF 4.0

All XSD file headers have same change regarding copyright and targetNameSpace:

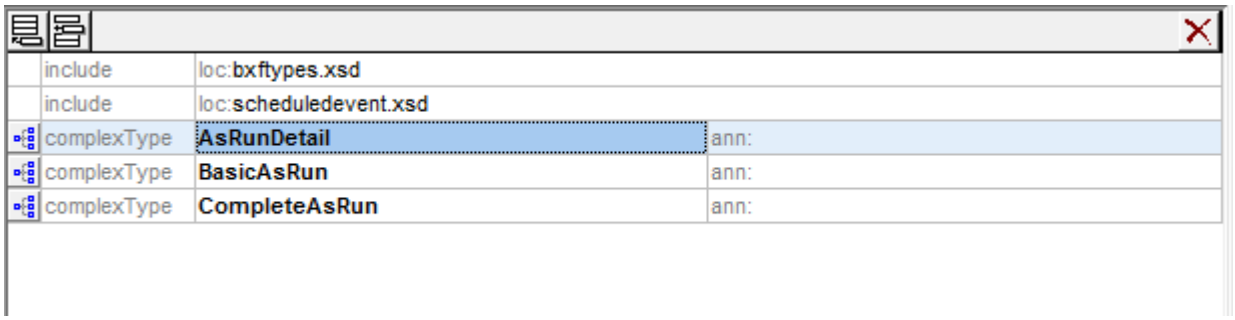
```
<!-- Copyright 2015 Society of Motion Picture and Television Engineers. All rights reserved. -->

<xs:schema xmlns="http://smpte-ra.org/schemas/2021/2015/BXF"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:pmcp="http://www.atsc.org/XMLSchemas/pmcp/2007/3.1" targetNamespace="http://smpte-
ra.org/schemas/2021/2015/BXF" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="4.000">
```

Most of the XSDs had only their header changed, and are listed below.

Asrun.XSD
Audio.XSD
BXFCaptions.XSD
BXFChannels.XSD
BXFContentID.XSD
BXFISAN.XSD
BXFParentalRating.XSD
BXFSchema.XSD
Configuration.XSD
Content.XSD
ContentTransfer.XSD
Contract.XSD
DataContent.XSD
Element.XSD
Location.XSD
Macro.XSD
NonPrimaryEvent.XSD
NonProgramContent.XSD
NonProgramDetail.XSD
NonProgramEvent.XSD
PrimaryEvent.XSD
Video.XSD

3.2.1 Asrun.XSD

Contains:

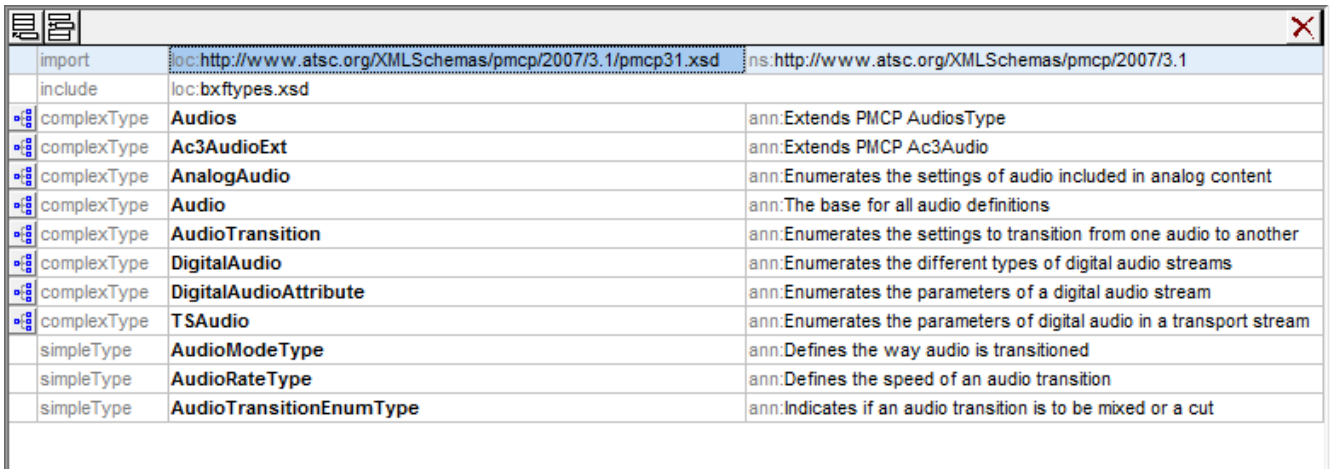


	include	loc:bxftypes.xsd	
	include	loc:scheduledevent.xsd	
	complexType	**AsRunDetail**	ann:
	complexType	**BasicAsRun**	ann:
	complexType	**CompleteAsRun**	ann:

Used to describe the exact timing of events on a schedule after the event has been aired, including any errors or other problems that occurred during the broadcast or transmission of the content.

3.2.2 Audio.XSD

Contains:

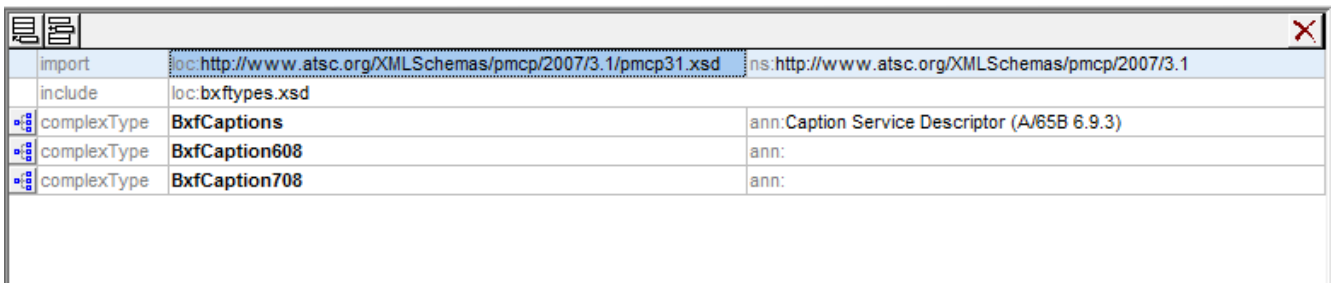


	import	loc:http://www.atsc.org/XMLSchemas/pmcp/2007/3.1/pmcp31.xsd	ns:http://www.atsc.org/XMLSchemas/pmcp/2007/3.1
	include	loc:bxftypes.xsd	
	complexType	**Audios**	ann:Extends PMCP AudiosType
	complexType	**Ac3AudioExt**	ann:Extends PMCP Ac3Audio
	complexType	**AnalogAudio**	ann:Enumerates the settings of audio included in analog content
	complexType	**Audio**	ann:The base for all audio definitions
	complexType	**AudioTransition**	ann:Enumerates the settings to transition from one audio to another
	complexType	**DigitalAudio**	ann:Enumerates the different types of digital audio streams
	complexType	**DigitalAudioAttribute**	ann:Enumerates the parameters of a digital audio stream
	complexType	**TSAudio**	ann:Enumerates the parameters of digital audio in a transport stream
	simpleType	**AudioModeType**	ann:Defines the way audio is transitioned
	simpleType	**AudioRateType**	ann:Defines the speed of an audio transition
	simpleType	**AudioTransitionEnumType**	ann:Indicates if an audio transition is to be mixed or a cut

The basis for all audio definitions.

3.2.3 BxfCaptions.XSD

Contains:

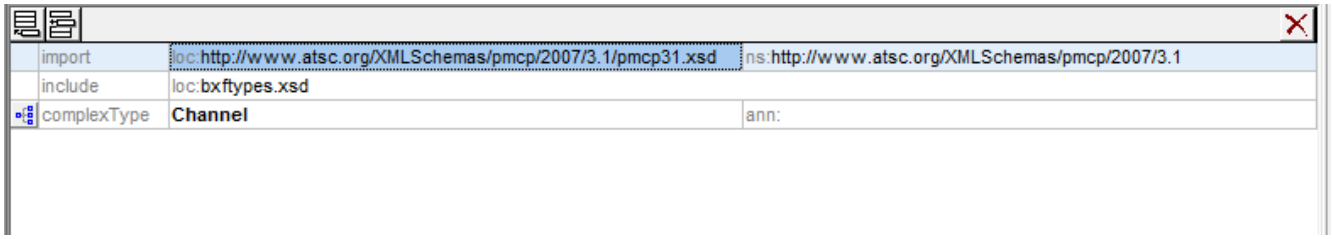


	import	loc:http://www.atsc.org/XMLSchemas/pmcp/2007/3.1/pmcp31.xsd	ns:http://www.atsc.org/XMLSchemas/pmcp/2007/3.1
	include	loc:bxftypes.xsd	
	complexType	**BxfCaptions**	ann:Caption Service Descriptor (A/65B 6.9.3)
	complexType	**BxfCaption608**	ann:
	complexType	**BxfCaption708**	ann:

Caption Service Descriptor (A/65B 6.9.3).

3.2.4 BxfChannel.XSD

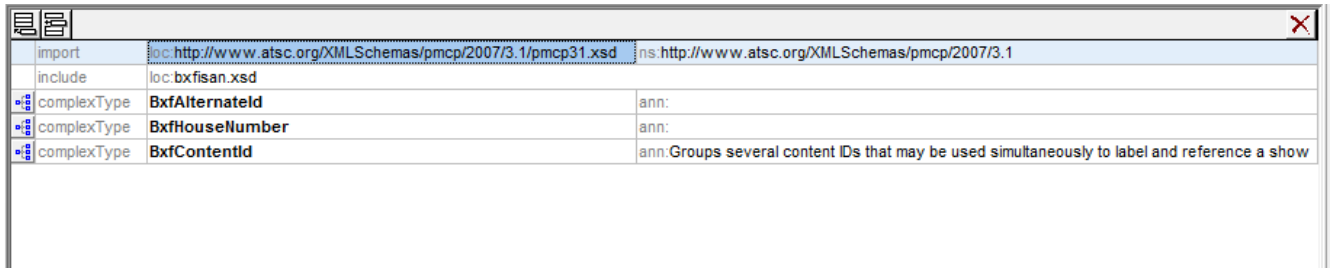
Contains:



Extends ATSC's PMCP 3.1 definition of a channel.

3.2.5 BxfContentId.XSD

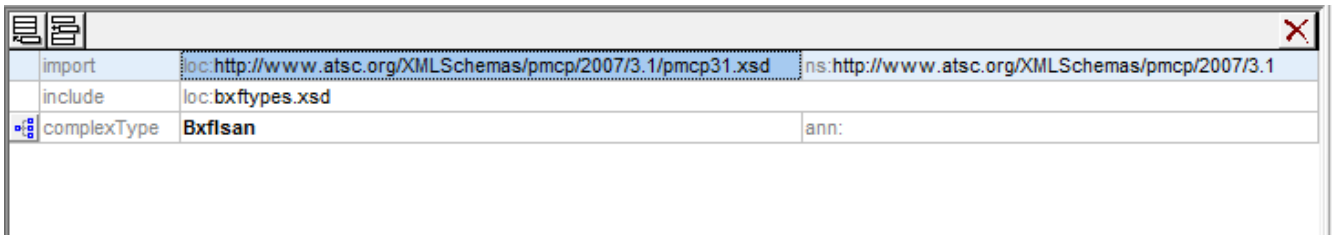
Contains:



Groups several Content IDs that may be used simultaneously to label and reference a show. In Version 2, support was added for the reuse of House Numbers by adding "duration of effectiveness" as supported under ATSC's A57/B standard.

3.2.6 BxfISAN.XSD

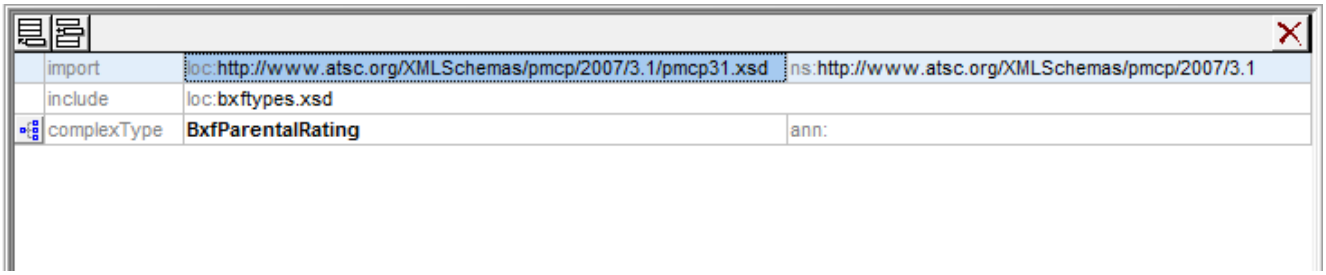
Contains:



Used to define entry of an ISAN value into ContentID. Go to www.isan.org for more information on the use of ISAN with content identification.

3.2.7 BxfParentalRating.XSD

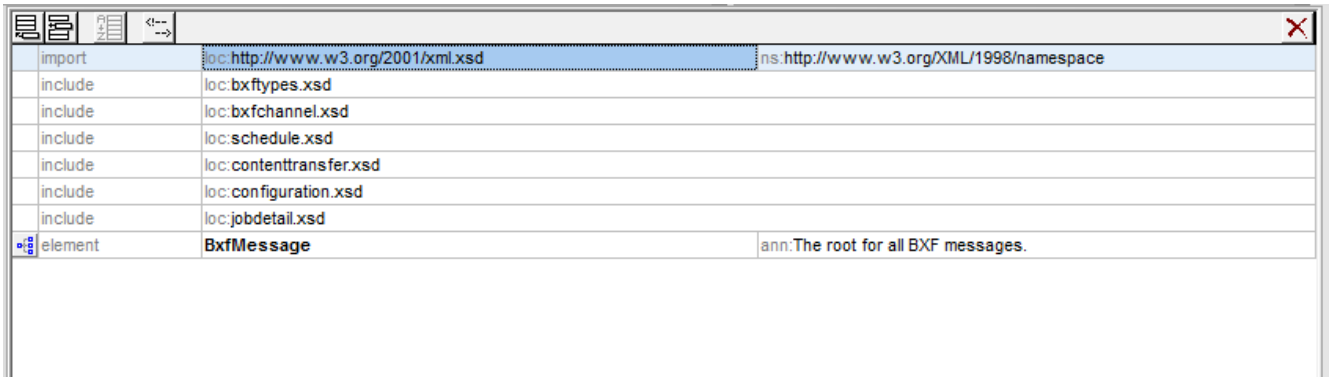
Contains:



Used to define entry of parental rating values as an extension of ATSC's PMCP 3.1.

3.2.8 BxfSchema.XSD

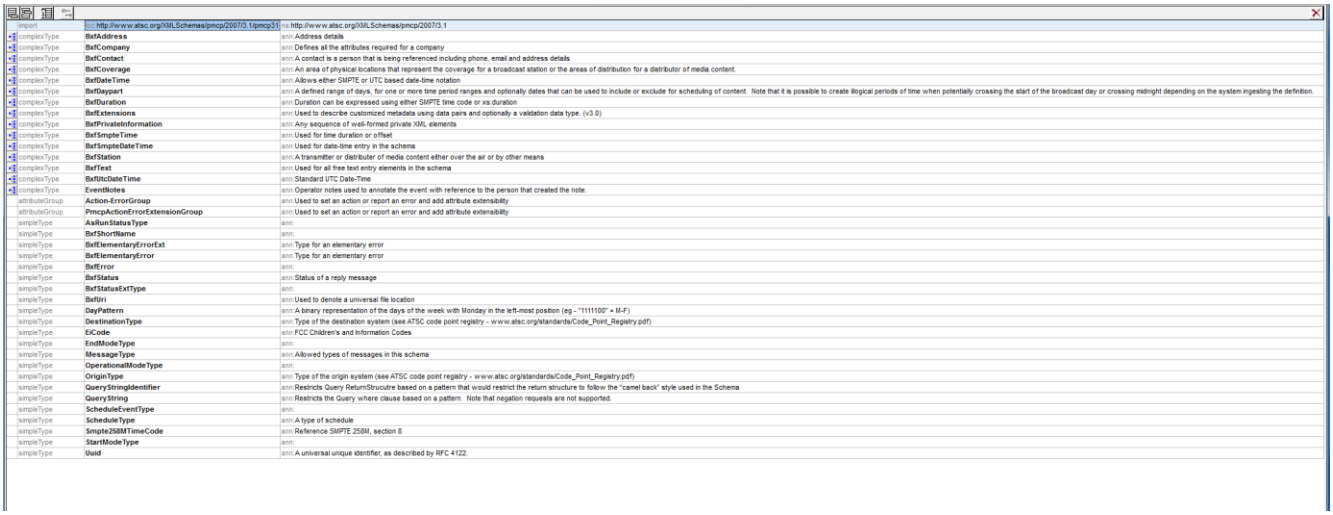
Contains:



The root for all BXF messages.

3.2.9 BxfTypes.XSD

Contains:



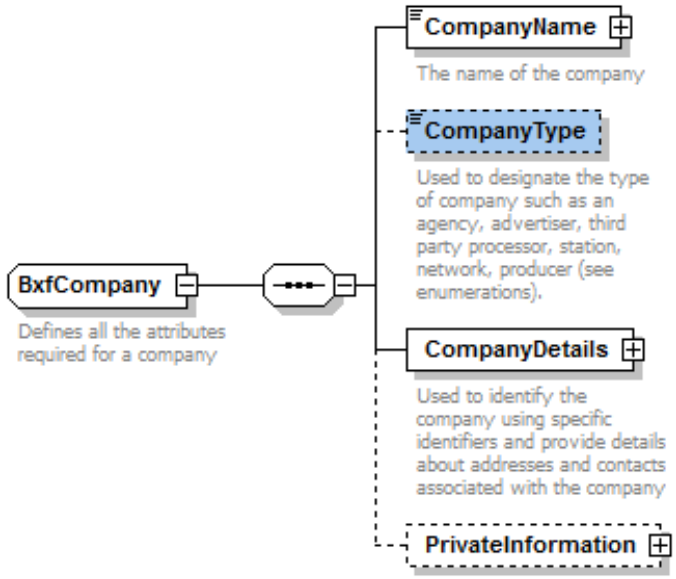
Used as to store all the utility elements that are used by most of the other schema files.

3.2.9.1 BXF 4.0 Changes

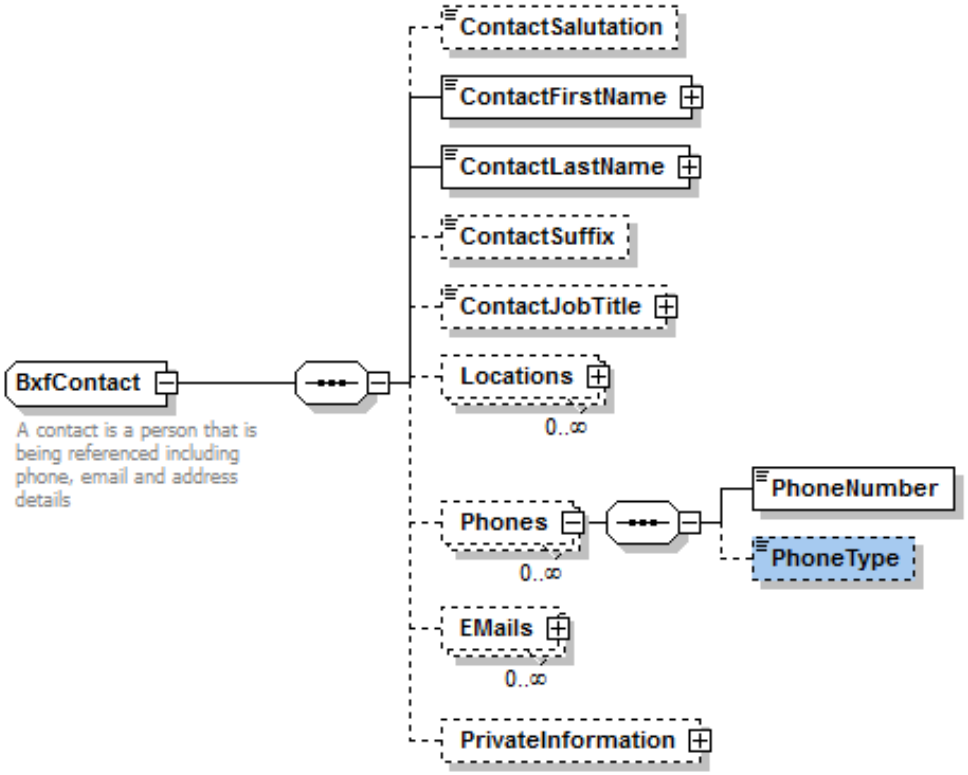
Description of change:
Under BxfCompany, “Producer” added as a new enumeration and the list resorted alphabetically for CompanyType along with adjustments to the annotation for the element. For PhoneType, under BxfContact, a new enumeration of “Fax” has been added.

Text representation:

```
<xs:element name="CompanyType" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Used to designate the type of company such as an agency, advertiser, third party processor, station, network, producer (see enumerations).</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="Advertiser"/>
            <xs:enumeration value="Agency"/>
            <xs:enumeration value="Cable Network"/>
            <xs:enumeration value="Interconnect Content Distributor"/>
            <xs:enumeration value="Local Cable System"/>
            <xs:enumeration value="Producer"/>
            <xs:enumeration value="Program Distributor"/>
            <xs:enumeration value="Radio Network"/>
            <xs:enumeration value="Radio Station"/>
            <xs:enumeration value="MSO"/>
            <xs:enumeration value="MVPD"/>
            <xs:enumeration value="Satellite Service"/>
            <xs:enumeration value="TV Network"/>
            <xs:enumeration value="TV Station"/>
            <xs:enumeration value="Other"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

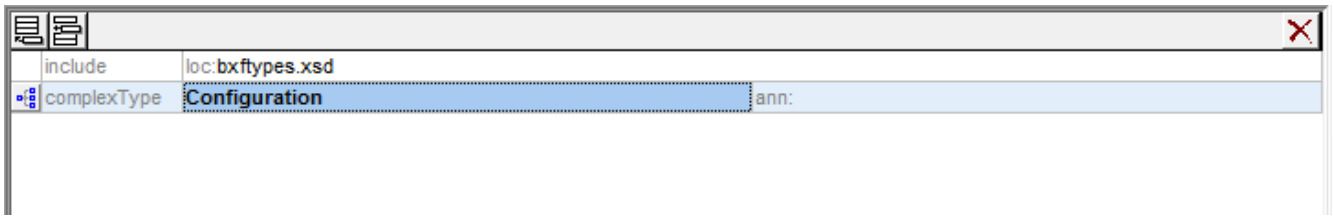


```
<xs:element name="PhoneType" minOccurs="0">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="Business"/>
            <xs:enumeration value="Home"/>
            <xs:enumeration value="Mobile"/>
            <xs:enumeration value="Alternate"/>
            <xs:enumeration value="Fax"/>
            <xs:enumeration value="Other"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```



3.2.10 Configuration.XSD

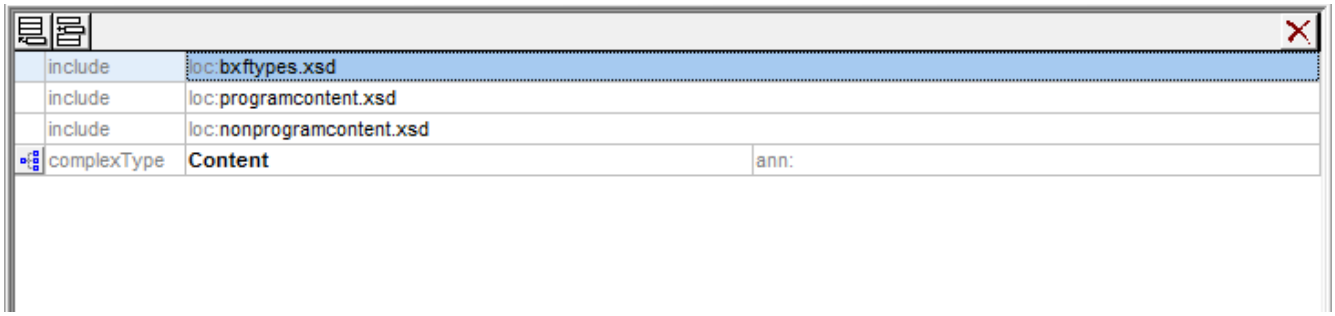
Contains:



Used to describe the configuration values of a system, to be shared with other BXF-compliant systems.

3.2.11 Content.XSD

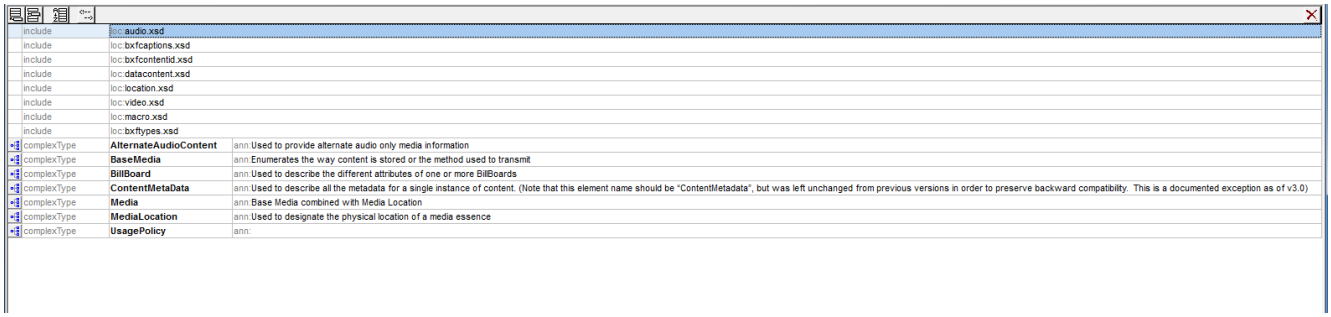
Contains:



Used to describe the content at a specific location.

3.2.12 ContentMetadata.XSD

Contains:



Content identification and other details.

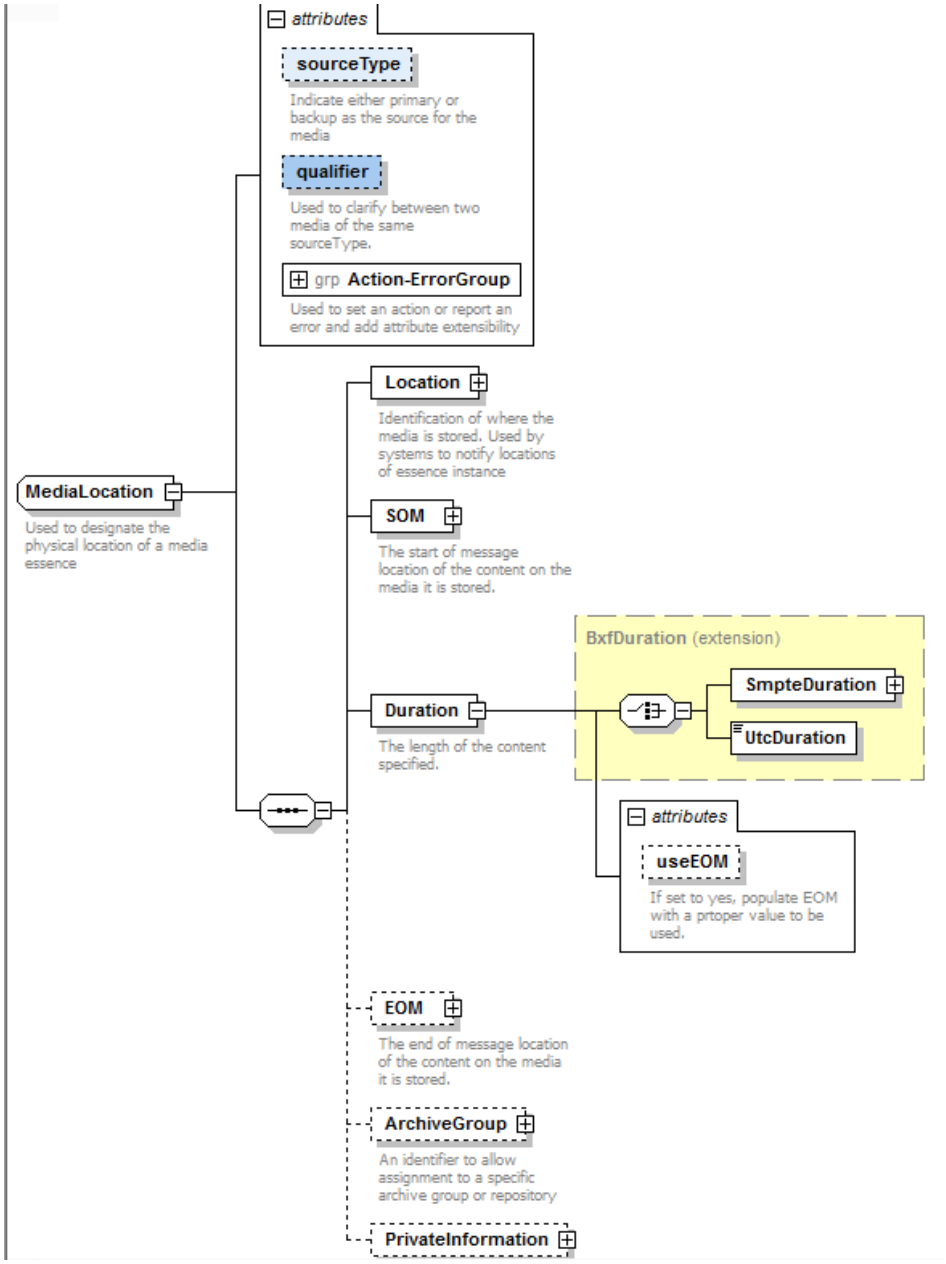
3.2.12.1 BXF 4.0 Changes

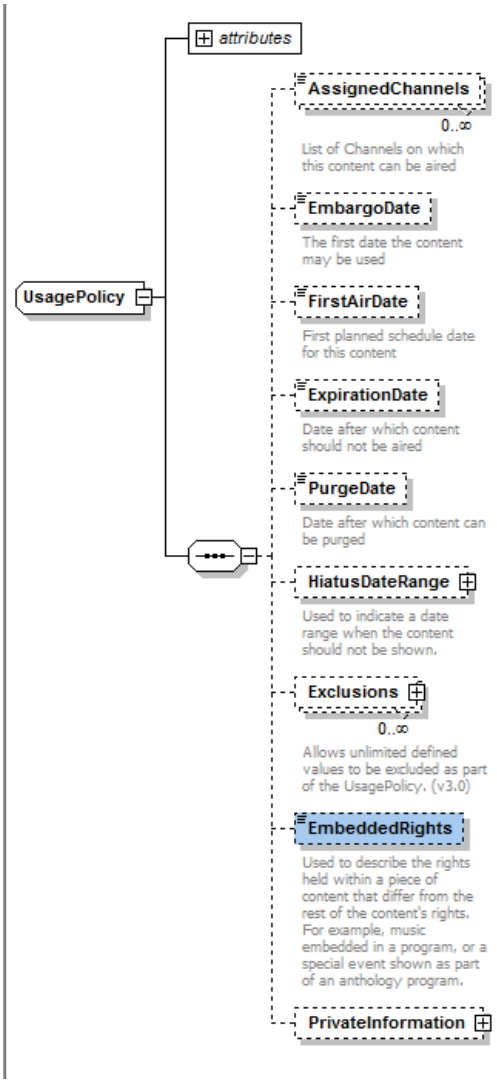
Description of change:
Duration used under MediaLocation has been extended to include a new boolean flag attribute called “useEOM” which signals than an EOM value is available and should be used rather than calculating it by using the SOM and Duration values. A new element “EOM” is also included to allow the entry of the EOM directly. Also, the attributes for MediaLocation have been enhanced. “sourceType” has three new enumeration options: Preview, Proxy and Other; and a new attribute “qualifier” is to be used to differentiate between two similar sources. UsagePolicy has also added a new element, “EmbeddedRights” used to describe the rights held within a piece of content that differ from the rest of the content's rights. For example, music embedded in a program, or a special event shown as part of an anthology program.

Text representation:

```
<xs:element name="Duration">
    <xs:annotation>
        <xs:documentation>The length of the content specified.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="BxfDuration">
                <xs:attribute name="useEOM" type="xs:boolean">
                    <xs:annotation>
                        <xs:documentation>If set to yes, populate EOM with a prtoper value to be
used.</xs:documentation>
```

```
                        </xs:annotation>
                    </xs:attribute>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="EOM" type="BxfSmpteTime" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The end of message location of the content on the media it is
stored.</xs:documentation>
        </xs:annotation>
    </xs:element>

    <xs:attribute name="sourceType" default="Primary">
        <xs:annotation>
            <xs:documentation>Indicate either primary or backup as the source for the media</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="Primary"/>
                <xs:enumeration value="Backup"/>
                <xs:enumeration value="Preview"/>
                <xs:enumeration value="Proxy"/>
                <xs:enumeration value="Other"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="qualifier" type="xs:string">
        <xs:annotation>
            <xs:documentation>Used to clarify between two media of the same sourceType.</xs:documentation>
        </xs:annotation>
    </xs:attribute>

    <xs:element name="EmbeddedRights" minOccurs="0">
        <xs:annotation>
            <xs:documentation>Used to describe the rights held within a piece of content that differ from the rest of
the content's rights.  For example, music embedded in a program, or a special event shown as part of an
anthology program.</xs:documentation>
        </xs:annotation>
    </xs:element>
```





3.2.13 ContentTransfer.XSD

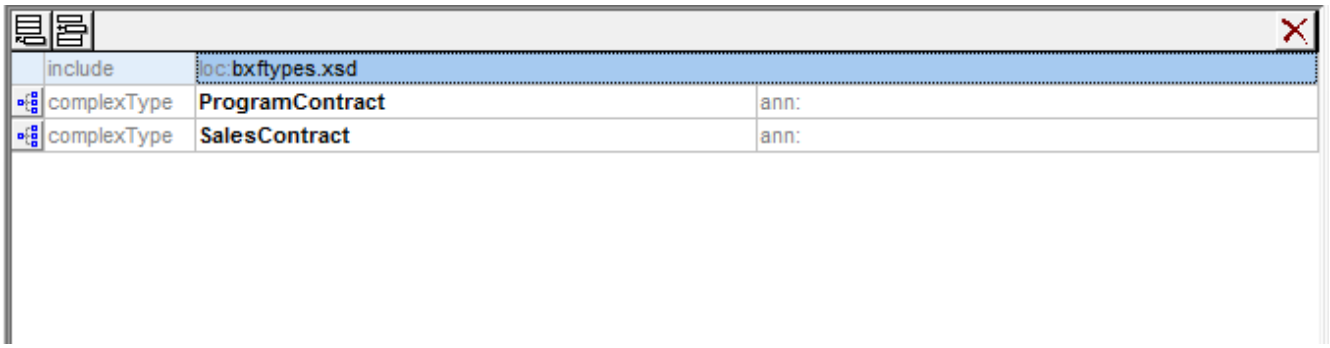
Contains:

	include	loc:content.xsd	
	include	loc:contentmetadata.xsd	
	complexType	**ContentTransfer**	ann:Transfer content from a source to a destination
	simpleType	**TransferType**	ann:
	simpleType	**PriorityType**	ann:
	simpleType	**TransferStatusType**	ann:

Used to transfer media from one location to another location, or to transcode it.

3.2.14 Contract.XSD

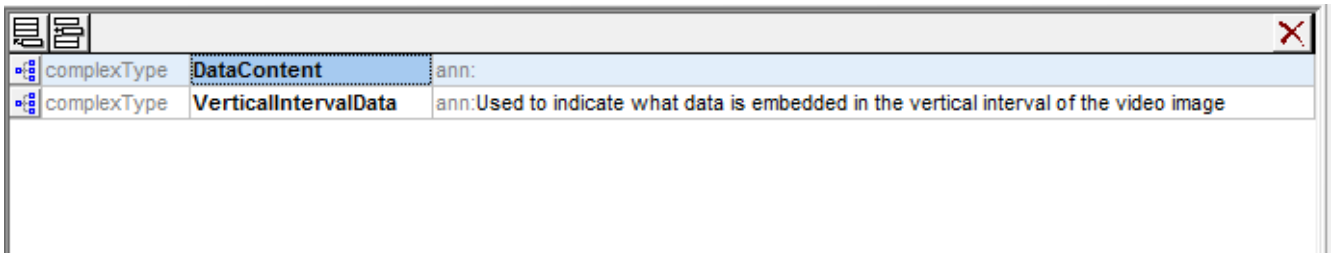
Contains:



ProgramContract - used if the program is purchased, it may be linked to a rights contract. SalesContract – used for contract information for the interstitial content (commercials).

3.2.15 DataContent.XSD

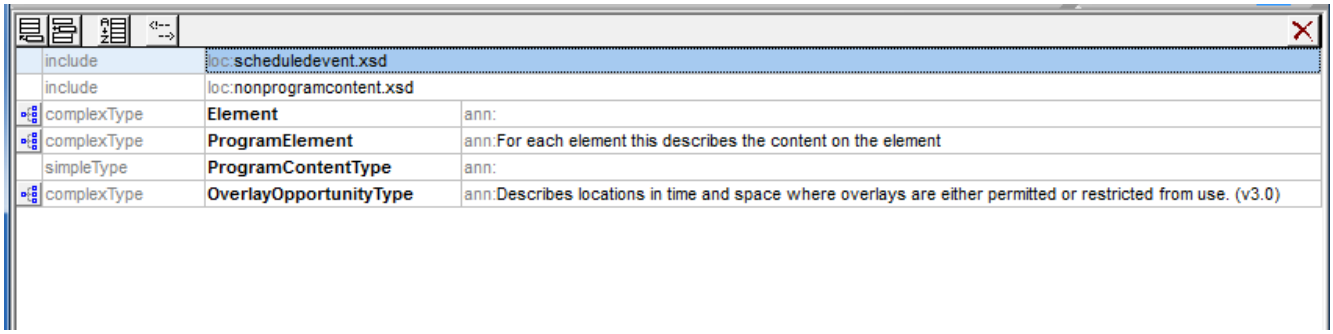
Contains:



Used to define data that can be added to the transport stream.

3.2.16 Element.XSD

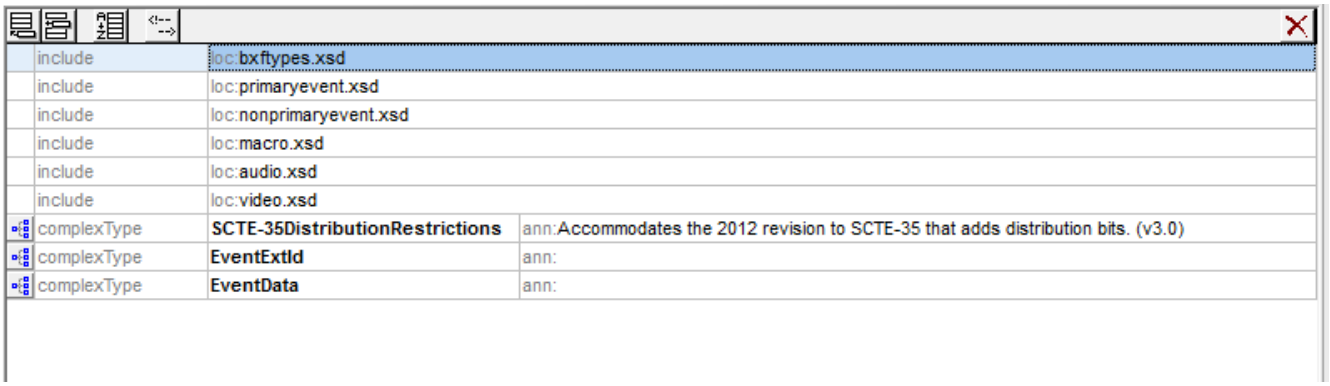
Contains:



If the program is being delivered in segmented form, this describes each segment's content, position and offset from the start of the program as well as other embedded content such as inserted barter content.

3.2.17 EventData.XSD

Contains:



	include	loc:bxftypes.xsd	
	include	loc:primaryevent.xsd	
	include	loc:nonprimaryevent.xsd	
	include	loc:macro.xsd	
	include	loc:audio.xsd	
	include	loc:video.xsd	
	complexType	**SCTE-35DistributionRestrictions**	ann:Accommodates the 2012 revision to SCTE-35 that adds distribution bits. (v3.0)
	complexType	**EventExtId**	ann:
	complexType	**EventData**	ann:

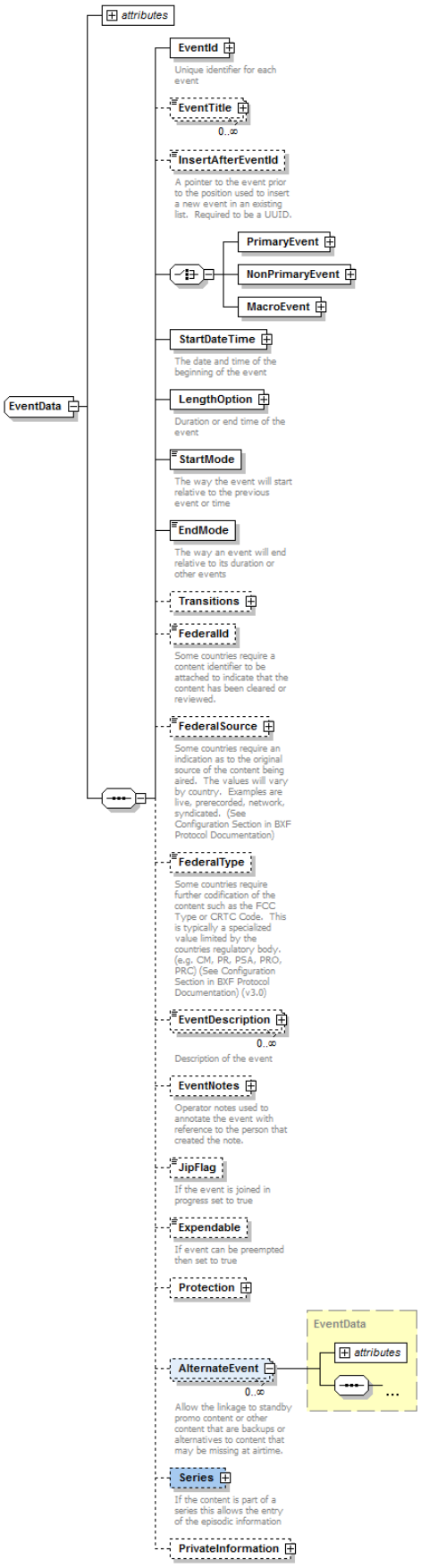
Specifies the data specific to a particular airing of a complete show or a single event. In Version 2, support for multiple titles and descriptions in a single message has been added in order to allow multi-language support.

3.2.17.1 BXF 4.0 Changes

Description of change:
"AlternateEvent" and "Series" added as new elements to EventData to provide linkage to standby promo content and to provide episodic detail entry at the event level.

Text representation:

```
<xs:element name="AlternateEvent" type="EventData" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>Allow the linkage to standby promo content or other content that are backups or alternatives to content that may be missing at airtime.</xs:documentation>
    </xs:annotation>
</xs:element>

<xs:element name="Series" type="Series" minOccurs="0">
    <xs:annotation>
        <xs:documentation>If the content is part of a series this allows the entry of the episodic information</xs:documentation>
    </xs:annotation>
</xs:element>
```



3.2.18 Format.XSD

Contains:

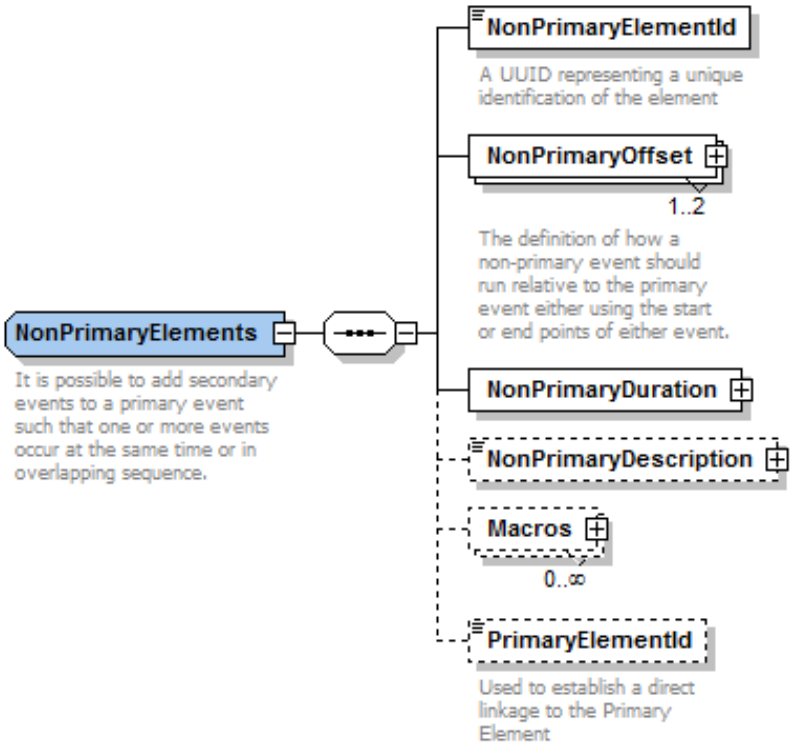
include	loc:bxftypes.xsd	
include	loc:macro.xsd	
complexType	**Format**	ann:
complexType	**FormatSubElements**	ann:**New** Used to subdivide a format element into smaller structures which when summed in duration would equal the total of the item.
complexType	**FormatUsage**	ann:Describes for a format which channels it can be used on
complexType	**NonPrimaryElements**	ann:**New** It is possible to add secondary events to a primary event such that one or more events occur at the same time or in overlapping sequence.

The definition of the skeletal structure of a program as used by the traffic system to construct a schedule grid.

3.2.18.1 BXF 4.0 Changes

Description of change:
Several enhancements were made to the Format.XSD. These include two new complex structures: “FormatSubElements” and “NonPrimaryElements”. NonPrimaryElements existed previously as an element, but not as a complex type. It is also an element of the new FormatSubElements and as such was encapsulated into its own new complex type. A new linkage element, “PrimaryElementId”, a UUID, is used to link the FormatSubElements to the primary element that can be created at the higher level. FormatSubElements are unbounded and are used to describe multiple “sub”-events that can be constructed to make a full event. The time associated to the sub-events should total to the length designated for the top-level event.

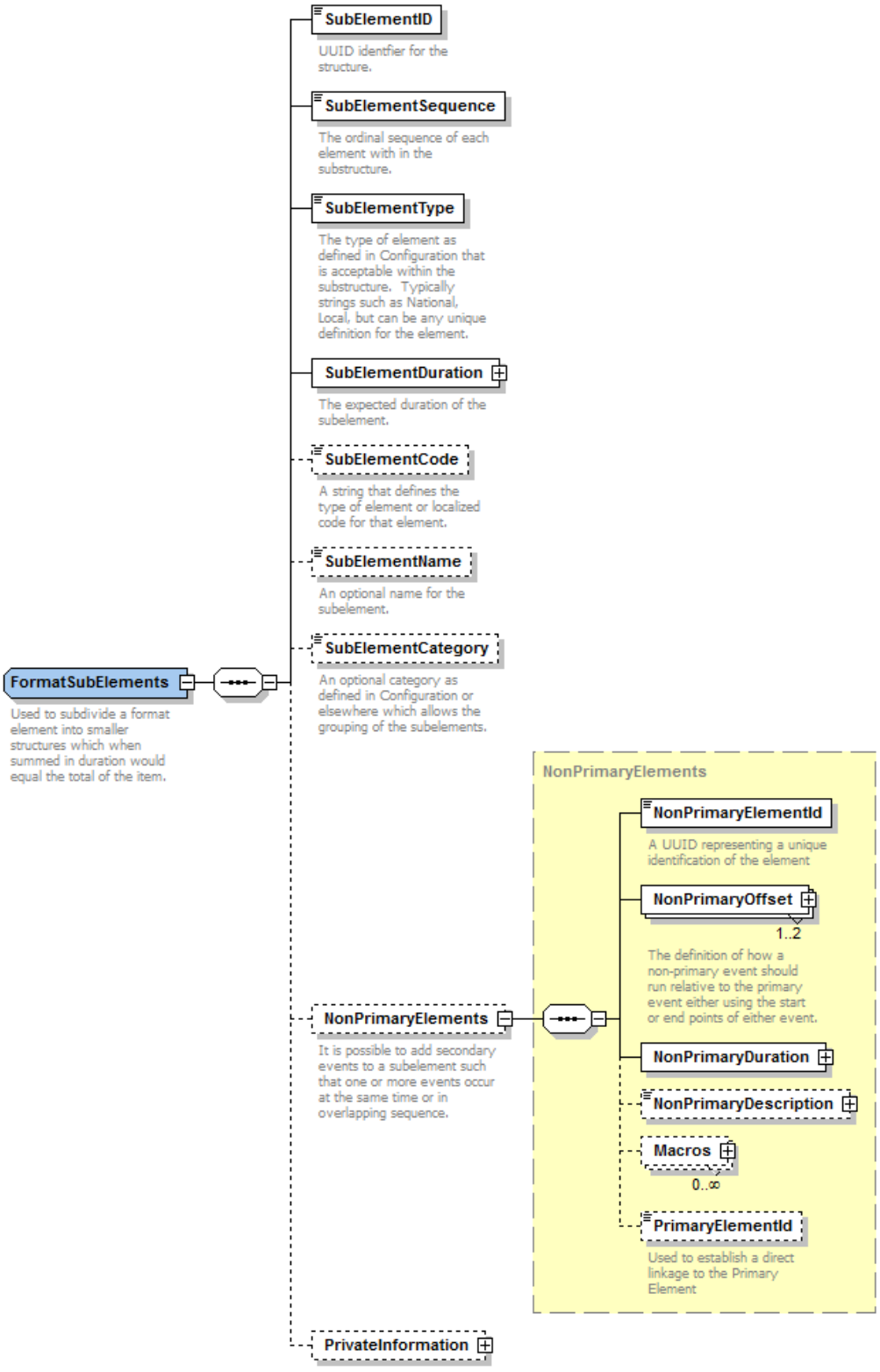
Text representation:
<xs:element name="NonPrimaryElements" type="NonPrimaryElements" minOccurs="0"/>



```
<xs:complexType name="NonPrimaryElements">
    <xs:annotation>
        <xs:documentation>It is possible to add secondary events to a primary event such that one or more events occur at the same time or in overlapping sequence.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="NonPrimaryElementId" type="Uuid">
            <xs:annotation>
                <xs:documentation>A UUID representing a unique identification of the element</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="NonPrimaryOffset" maxOccurs="2">
            <xs:annotation>
                <xs:documentation>The definition of how a non-primary event should run relative to the primary event either using the start or end points of either event.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="OffsetTime" type="BxfSmpteTime" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>The time to start the event relative to the start time of the primary event</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="OffsetFrom">
                    <xs:annotation>
                        <xs:documentation>Defines the point of the primary event from which the offset is applied either start or end</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="BeginningofEvent"/>
                            <xs:enumeration value="EndofEvent"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="OffsetType">
                    <xs:annotation>
                        <xs:documentation>This describes whether the start or end point of the non-primary event should be used when the offset is applied</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="Start"/>
                            <xs:enumeration value="End"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="Direction">
                    <xs:annotation>
                        <xs:documentation>Defines whether the offset value is applied as a positive or negative value</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="Positive"/>
```

```
                                <xs:enumeration value="Negative"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:complexType>
            </xs:element>
            <xs:element name="NonPrimaryDuration" type="BxfDuration"/>
            <xs:element name="NonPrimaryDescription" type="BxfText" minOccurs="0"/>
            <xs:element name="Macros" type="Macro" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="PrimaryElementId" type="Uuid" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Used to establish a direct linkage to the Primary
Element</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
</xs:complexType>



<xs:element name="FormatSubElements" type="FormatSubElements" minOccurs="0"
maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>Used to subdivide a format element into smaller structures which when summed in
duration would equal the total of the item.</xs:documentation>
        </xs:annotation>
</xs:element>

<xs:complexType name="FormatSubElements">
        <xs:annotation>
            <xs:documentation>Used to subdivide a format element into smaller structures which when summed in
duration would equal the total of the item.</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="SubElementID" type="Uuid">
                <xs:annotation>
                    <xs:documentation>UUID identfier for the structure.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="SubElementSequence" type="xs:integer">
                <xs:annotation>
                    <xs:documentation>The ordinal sequence of each element with in the
substructure.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="SubElementType" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The type of element as defined in Configuration that is acceptable within
the substructure.  Typically strings such as National, Local, but can be any unique definition for the
element.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="SubElementDuration">
                <xs:annotation>
                    <xs:documentation>The expected duration of the subelement.</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:complexContent>
```

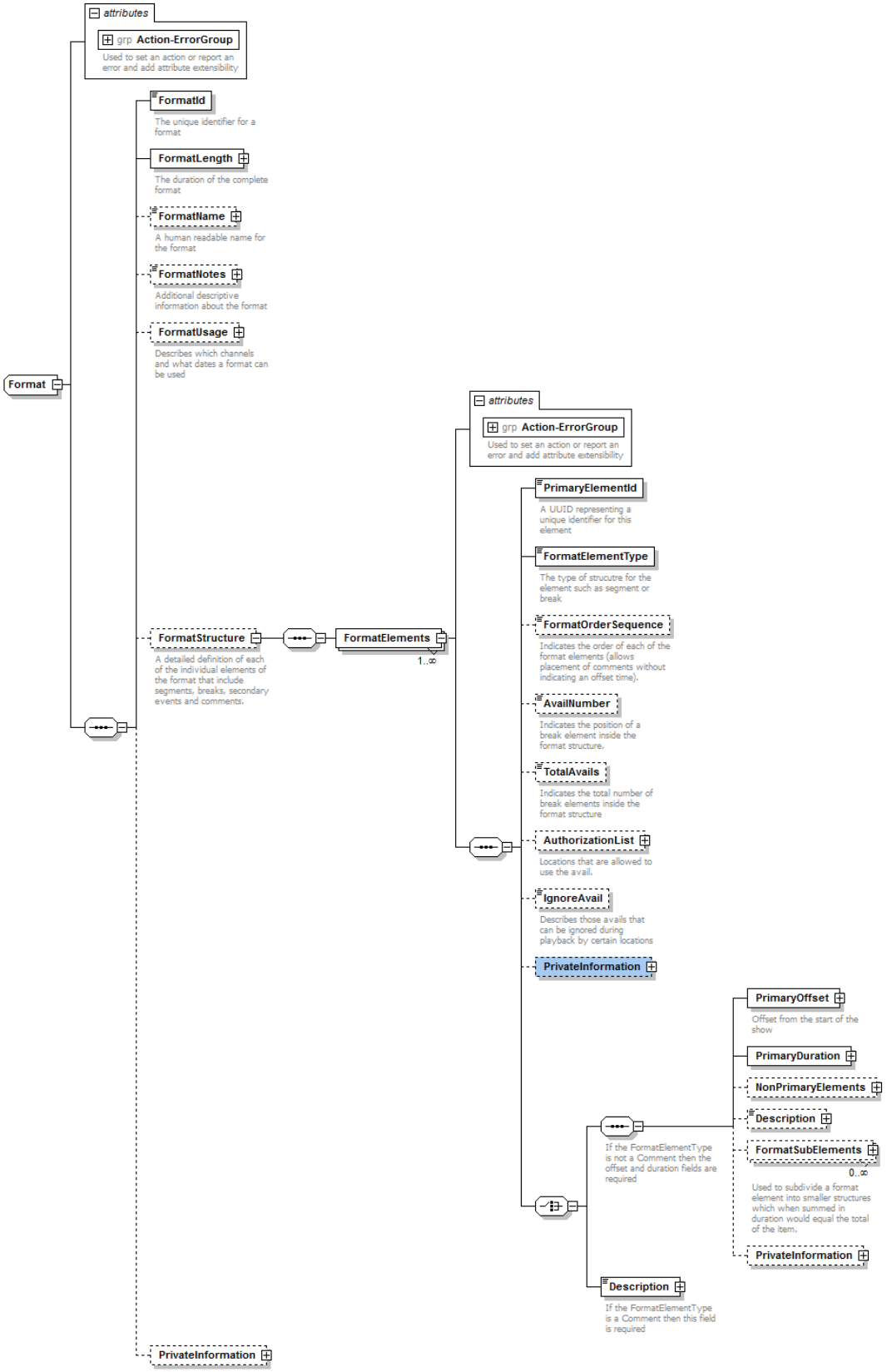
```
                        <xs:extension base="BxfDuration">
                            <xs:attribute name="variable">
                                <xs:annotation>
                                    <xs:documentation>Used to indicate if the duration can be changed based on the contents added to the format element.</xs:documentation>
                                </xs:annotation>
                            </xs:attribute>
                        </xs:extension>
                    </xs:complexContent>
                </xs:complexType>
            </xs:element>
            <xs:element name="SubElementCode" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>A string that defines the type of element or localized code for that element.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="SubElementName" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>An optional name for the subelement.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="SubElementCategory" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>An optional category as defined in Configuration or elsewhere which allows the grouping of the subelements.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="NonPrimaryElements" type="NonPrimaryElements" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>It is possible to add secondary events to a subelement such that one or more events occur at the same time or in overlapping sequence.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="PrivateInformation" type="BxfPrivateInformation" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
```



PrivateInformation was left off of the FormatElements structure in previous versions and has been added for consistency.

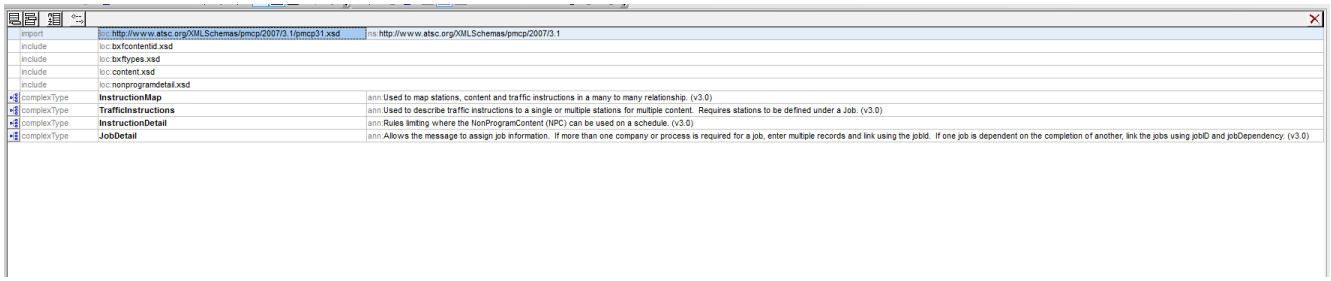
Text representation:

<xs:element name="PrivateInformation" type="BxfPrivateInformation" minOccurs="0"></xs:element>



3.2.19 JobDetail.XSD

Contains:



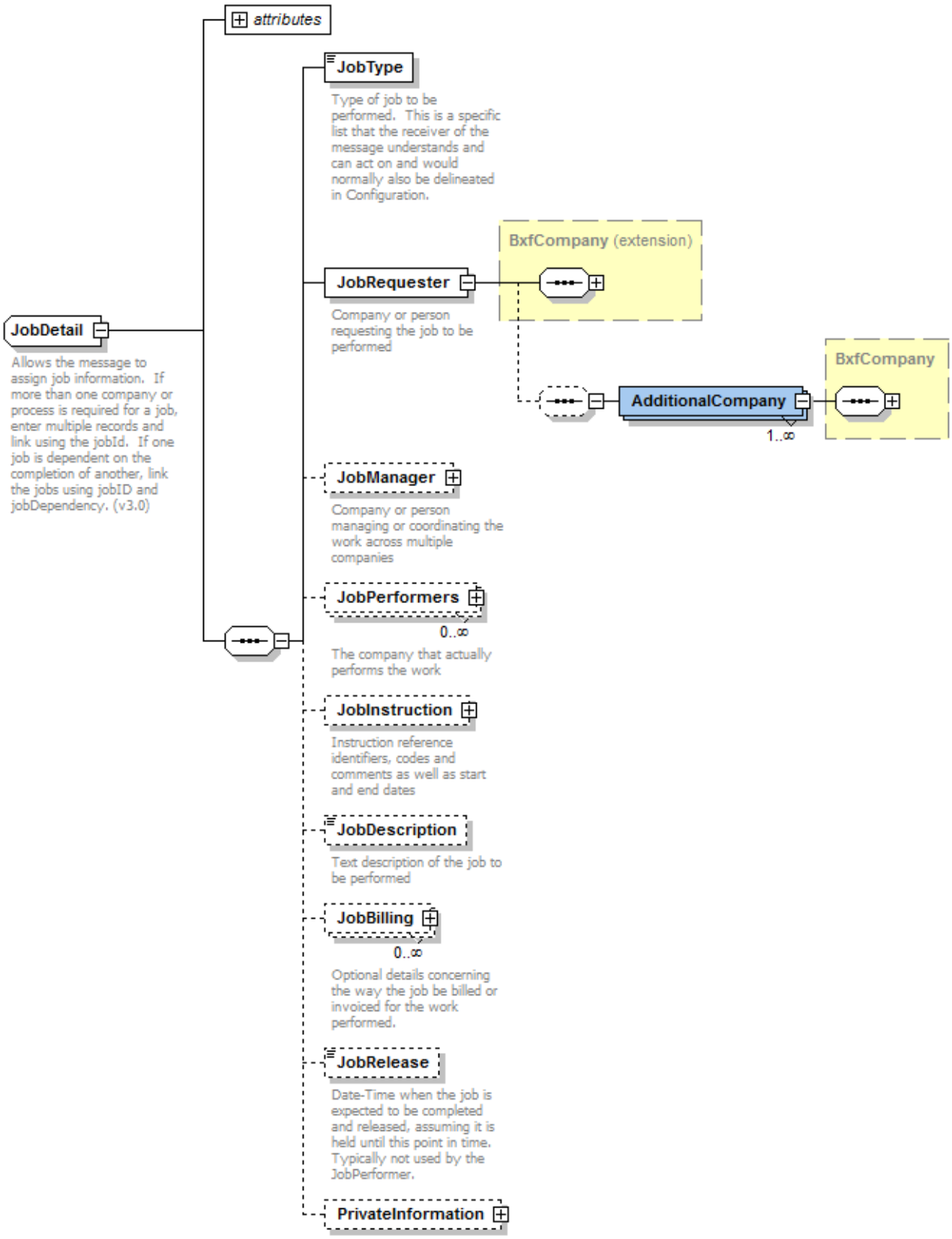
Allows the message to assign job information. If more than one company or process is required for a job, enter multiple records and link using the jobId. If one job is dependent on the completion of another, link the jobs using jobID and jobDependency.

3.2.19.1 BXF 4.0 Changes

Description of change:
The JobRequester required extension to include multiple companies. As such, BxfCompany was extended to allow the entry of multiple company records below the primary company requesting the job.

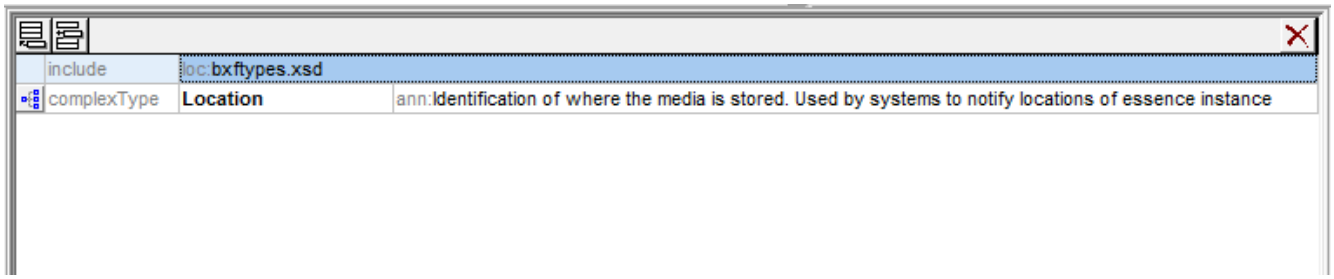
Text representation:

```
<xs:element name="JobRequester">
    <xs:annotation>
        <xs:documentation>Company or person requesting the job to be performed</xs:documentation>
    </xs:annotation>
        <xs:complexType>
            <xs:complexContent>
                <xs:extension base="BxfCompany">
                    <xs:sequence minOccurs="0">
                        <xs:element name="AdditionalCompany" type="BxfCompany"
maxOccurs="unbounded"></xs:element>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
</xs:element>
```



3.2.20 Location.XSD

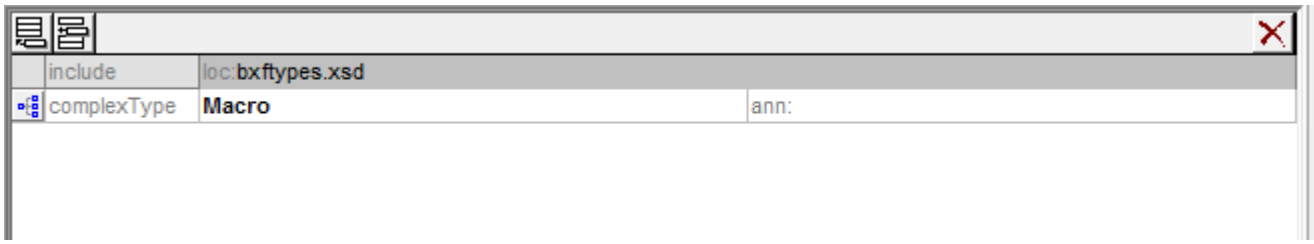
Contains:



Identification of where the media is stored. Used by systems to notify locations of content.

3.2.21 Macro.XSD

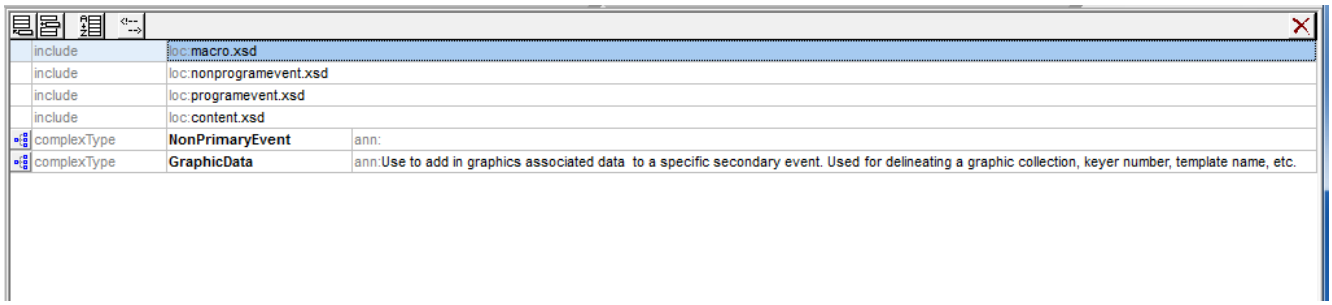
Contains:



Used to describe a MacroEvent under EventData.

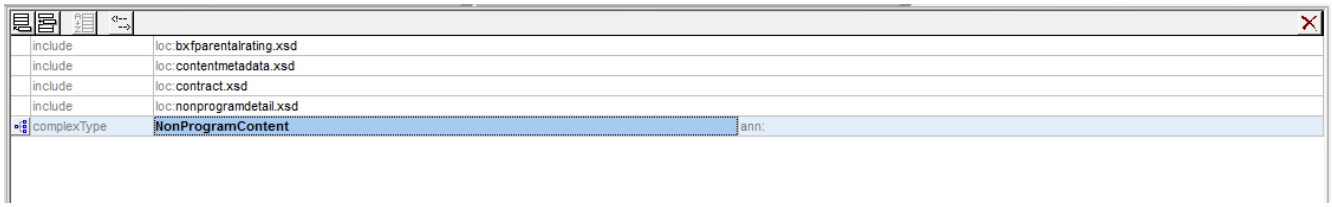
3.2.22 NonPrimaryEvent.XSD

Contains:



Used to describe those events that are not the primary content of a transport stream or playlist. Secondary events happen at the same time or in conjunction with the broadcast of the primary video and audio streams and may be started and completed using offsets from the start of the primary events. Note that in Version 2, BXF supports the inclusion of program events referenced by the secondary event.

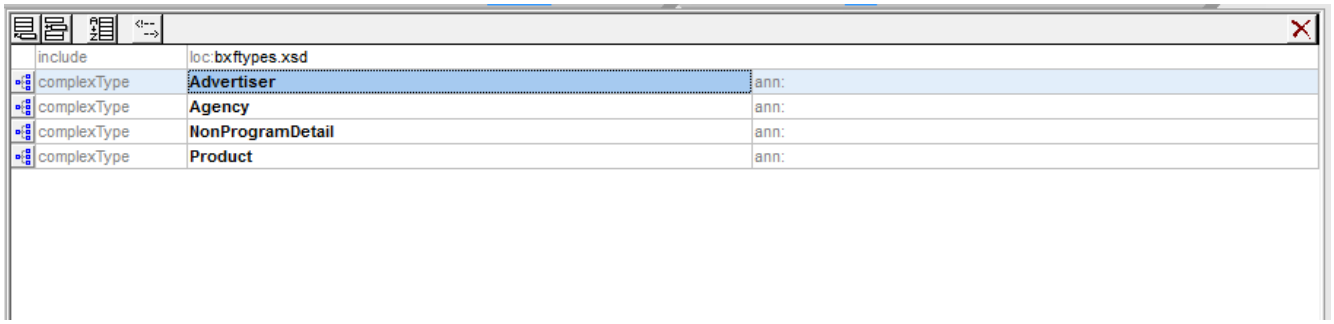
3.2.23 NonProgramContent.XSD



Non-program content typically describes short form content such as commercials, psa's, id's and other events that are not considered full length program content.

3.2.24 NonProgramDetail.XSD

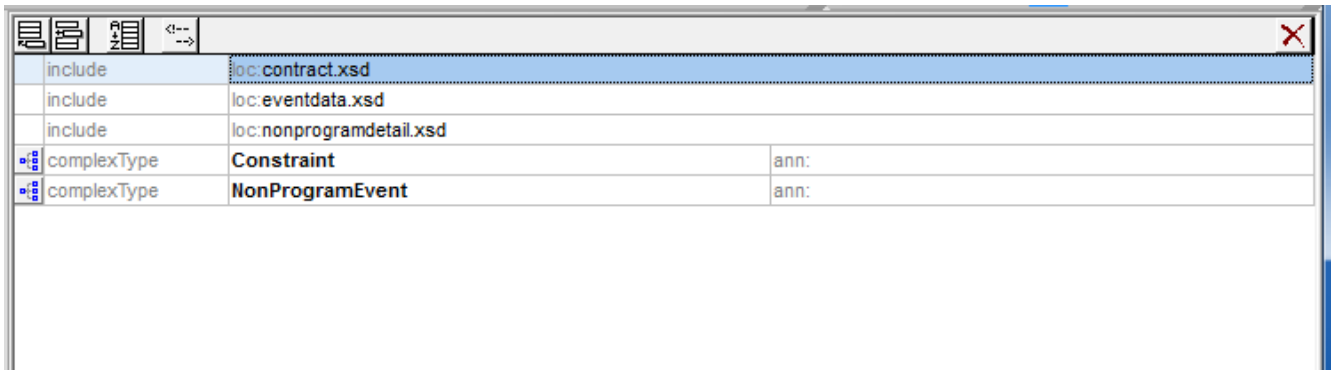
Contains:



Details of the non-program content.

3.2.25 NonProgramEvent.XSD

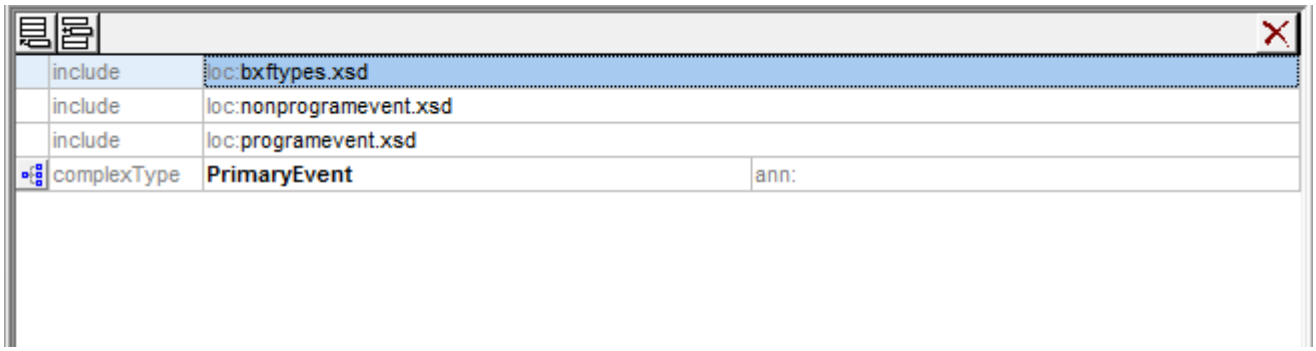
Contains:



Describes an event that contains NonProgramContent such as commercials and other interstitials, typically of short duration.

3.2.26 PrimaryEvent.XSD

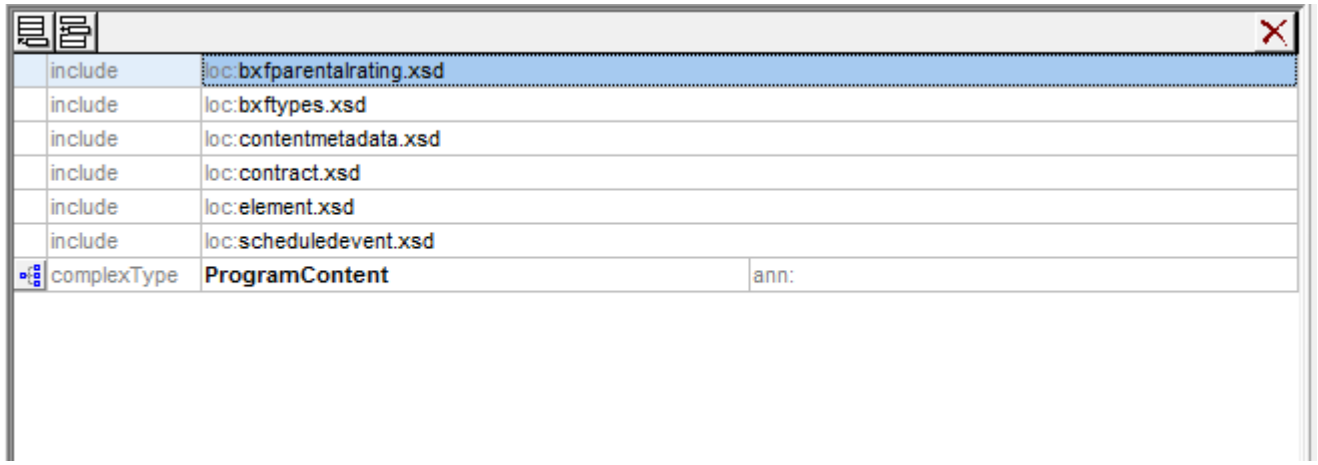
Contains:



Describes an event that contains the primary audio and video content for the transport stream or playlist.

3.2.27 ProgramContent.XSD

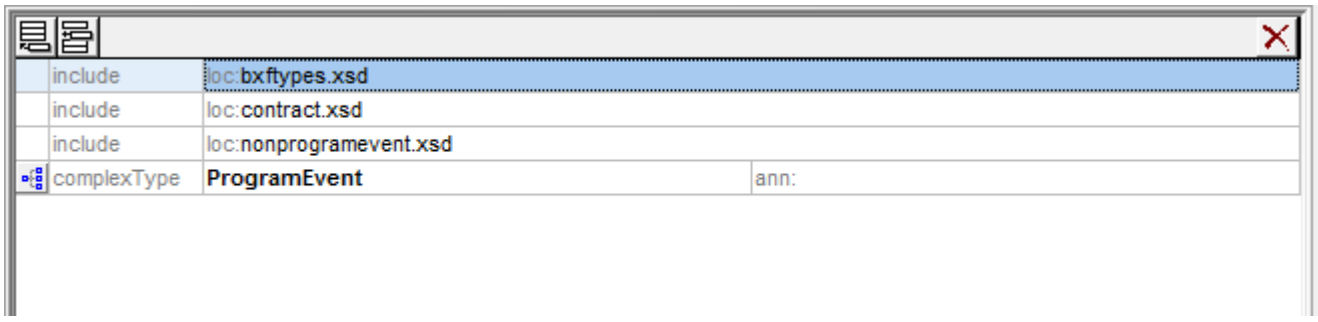
Contains:



Used to describe the full length content of a show.

3.2.28 ProgramEvent.XSD

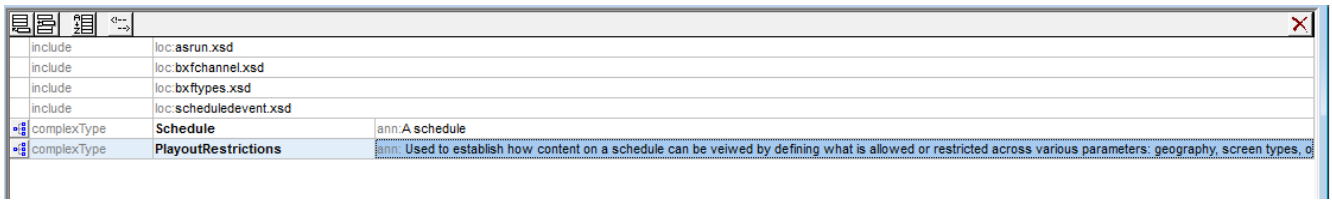
Contains:



Describes an event that contains ProgramContent.

3.2.29 Schedule.XSD

Contains:



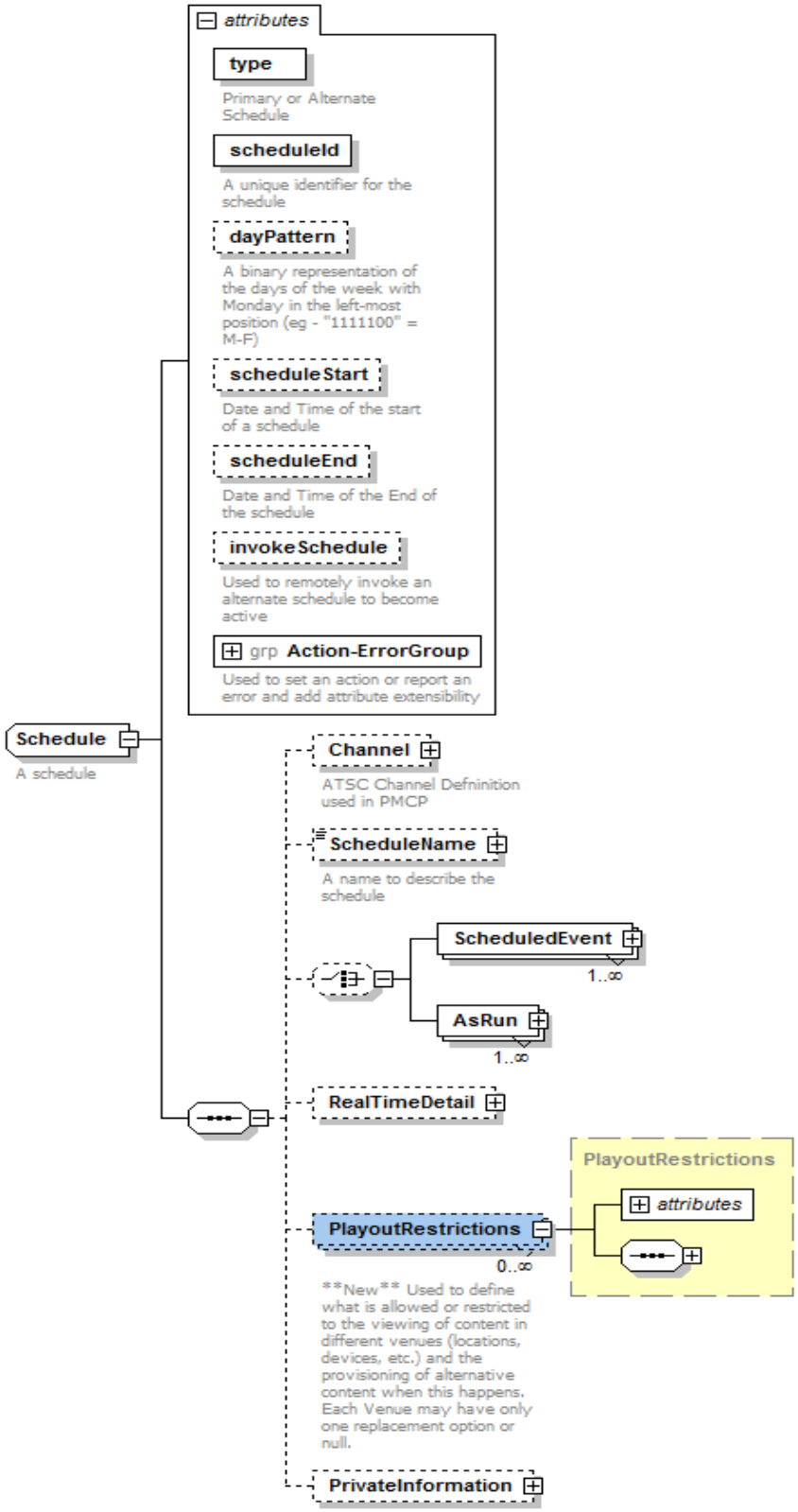
Used to describe a specific list of content in a linear playout sequence. In Version 2, support has been added to communicate schedule changes in real time by a new node, RealTimeDetail. This allows the indication of which event is the current event playing as well as the events that had played out previously and those that will playout next.

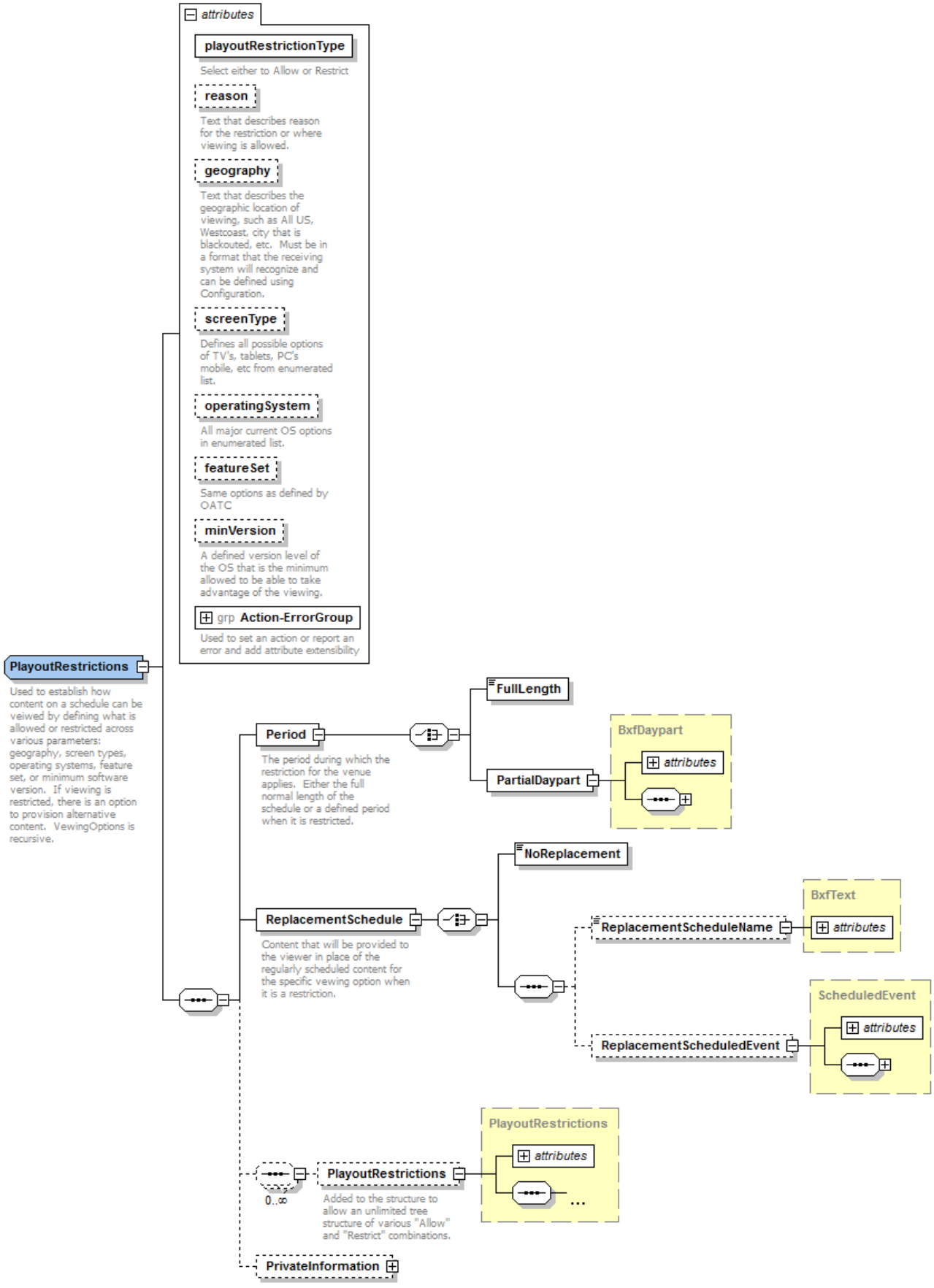
3.2.29.1 BXF 4.0 Changes

Description of change:
In Version 4, we have added PlayoutRestrictions as an additional element under Schedule which can be used to establish how content on a schedule can be veiwed by defining what is allowed or restricted across various parameters: geography, screen types, operating systems, feature set, or minimum software version. If viewing is restricted, there is an option to provision alternative content. VewingOptions is recursive.

Text representation:

```
<xs:element name="PlayoutRestrictions" type="PlayoutRestrictions" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation> Used to define what is allowed or restricted to the viewing of content in different venues (locations, devices, etc.) and the provisioning of alternative content when this happens. Each Venue may have only one replacement option or null</xs:documentation>
    </xs:annotation>
</xs:element>
```





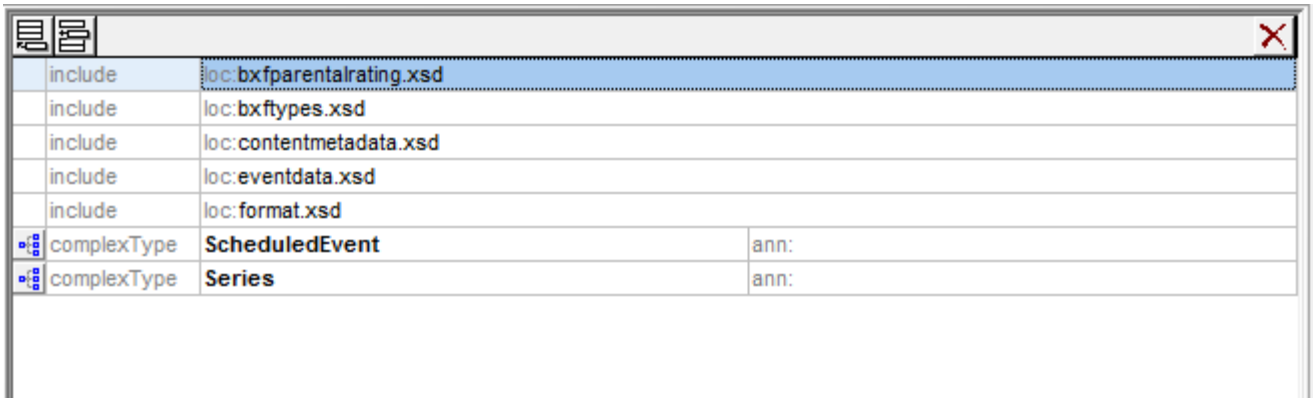
```
<xs:complexType name="PlayoutRestrictions">
    <xs:annotation>
        <xs:documentation>Used to establish how content on a schedule can be veiwed by defining what is allowed or restricted across various parameters: geography, screen types, operating systems, feature set. or minmum software version.  If viewing is restricted, there is an option to provision alternative content. VewingOptions is recursive.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Period">
            <xs:annotation>
                <xs:documentation>The period during which the restriction for the venue applies.  Either the full normal length of the schedule or a defined period when it is restricted.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:choice>
                    <xs:element name="FullLength" type="xs:boolean" default="1"/>
                    <xs:element name="PartialDaypart" type="BxfDaypart"/>
                </xs:choice>
            </xs:complexType>
        </xs:element>
        <xs:element name="ReplacementSchedule">
            <xs:annotation>
                <xs:documentation>Content that will be provided to the viewer in place of the regularly scheduled content for the specific vewing option when it is a restriction.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:choice>
                    <xs:element name="NoReplacement" default="1">
                        <xs:complexType>
                            <xs:simpleContent>
                                <xs:extension base="xs:boolean"/>
                            </xs:simpleContent>
                        </xs:complexType>
                    </xs:element>
                    <xs:sequence>
                        <xs:element name="ReplacementScheduleName" type="BxfText" minOccurs="0"/>
                        <xs:element name="ReplacementScheduledEvent" type="ScheduledEvent" minOccurs="0"/>
                    </xs:sequence>
                </xs:choice>
            </xs:complexType>
        </xs:element>
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="PlayoutRestrictions" type="PlayoutRestrictions" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Added to the structure to allow an unlimited tree structure of various "Allow" and "Restrict" combinations.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:element name="PrivateInformation" type="BxfPrivateInformation" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="playoutRestrictionType" use="required">
        <xs:annotation>
            <xs:documentation>Select either to Allow or Restrict</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
```

```
                    <xs:enumeration value="Allow"/>
                    <xs:enumeration value="Restrict"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="reason" type="xs:string">
            <xs:annotation>
                <xs:documentation>Text that describes reason for the restriction or where viewing is
allowed.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="geography" type="xs:string">
            <xs:annotation>
                <xs:documentation>Text that describes the geographic location of viewing, such as All US,
Westcoast, city that is blackouted, etc.  Must be in a format that the receiving system will recognize and can be
defined using Configuration.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="screenType">
            <xs:annotation>
                <xs:documentation>Defines all possible options of TV's, tablets, PC's mobile, etc from enumerated
list.</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="All"/>
                    <xs:enumeration value="TV Only"/>
                    <xs:enumeration value="All Non-TV Devices"/>
                    <xs:enumeration value="Desktop Computers"/>
                    <xs:enumeration value="Tablets"/>
                    <xs:enumeration value="Mobile Phones"/>
                    <xs:enumeration value="Tablets and Phones"/>
                    <xs:enumeration value="Computers and Phones"/>
                    <xs:enumeration value="Computers and Tablets"/>
                    <xs:enumeration value="Other"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="operatingSystem">
            <xs:annotation>
                <xs:documentation>All major current OS options in enumerated list.</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="Android"/>
                    <xs:enumeration value="Linux"/>
                    <xs:enumeration value="MAC-OSX"/>
                    <xs:enumeration value="OSI"/>
                    <xs:enumeration value="PC-Windows"/>
                    <xs:enumeration value="Unknown"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="featureSet">
            <xs:annotation>
                <xs:documentation>Same options as defined by OATC</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
```

```
                <xs:restriction base="xs:string">
                    <xs:enumeration value="DigitalOutput"/>
                    <xs:enumeration value="LocationAware"/>
                    <xs:enumeration value="LocalStorage"/>
                    <xs:enumeration value="AnalogOutput"/>
                    <xs:enumeration value="HDCP"/>
                    <xs:enumeration value="FastForward"/>
                    <xs:enumeration value="ProgrammerBrandingRequired"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="minVersion" type="xs:decimal">
            <xs:annotation>
                <xs:documentation>A defined version level of the OS that is the minimum allowed to be able to
take advantage of the viewing.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attributeGroup ref="Action-ErrorGroup"/>
</xs:complexType>
```

3.2.30 ScheduledEvent.XSD

Contains:



Used to describe a single event within a specific list of content arranged in a linear playout sequence.

3.2.30.1 BXF 4.0 Changes

Description of change:
Several enhancements have been added to extend ScheduleEvent to be more eflexible. First, a new attribute had been added, “excludeFromEPG” used to indicate to downstream systems that the event name and metadata should not be displayed in the user EPG. Next, the Content element, which previously only allowed ContentMetadata elements, has been expanded to include the full unbounded Content option that includes nonprogramcontent structures. The older structure is still in place for backward compatibility, but the new structure should be used for future development. ScheduledElements also has a new unbounded “choice” element between Content and MultiContent elements that does the same extension at this level. Lastly, the AlternateScheduleEventFlag is added to create linkage to standby promo and other backup content and PlayoutRestrictions is also added to this level to provide a way to restriction playout is specific situations. The Series Complex Type has also been enhanced to include some new elements. This includes a SeasonCode, SeriesCode, EpisodeNumer and an unbounded Producers element to indicate the companies that produced the series.

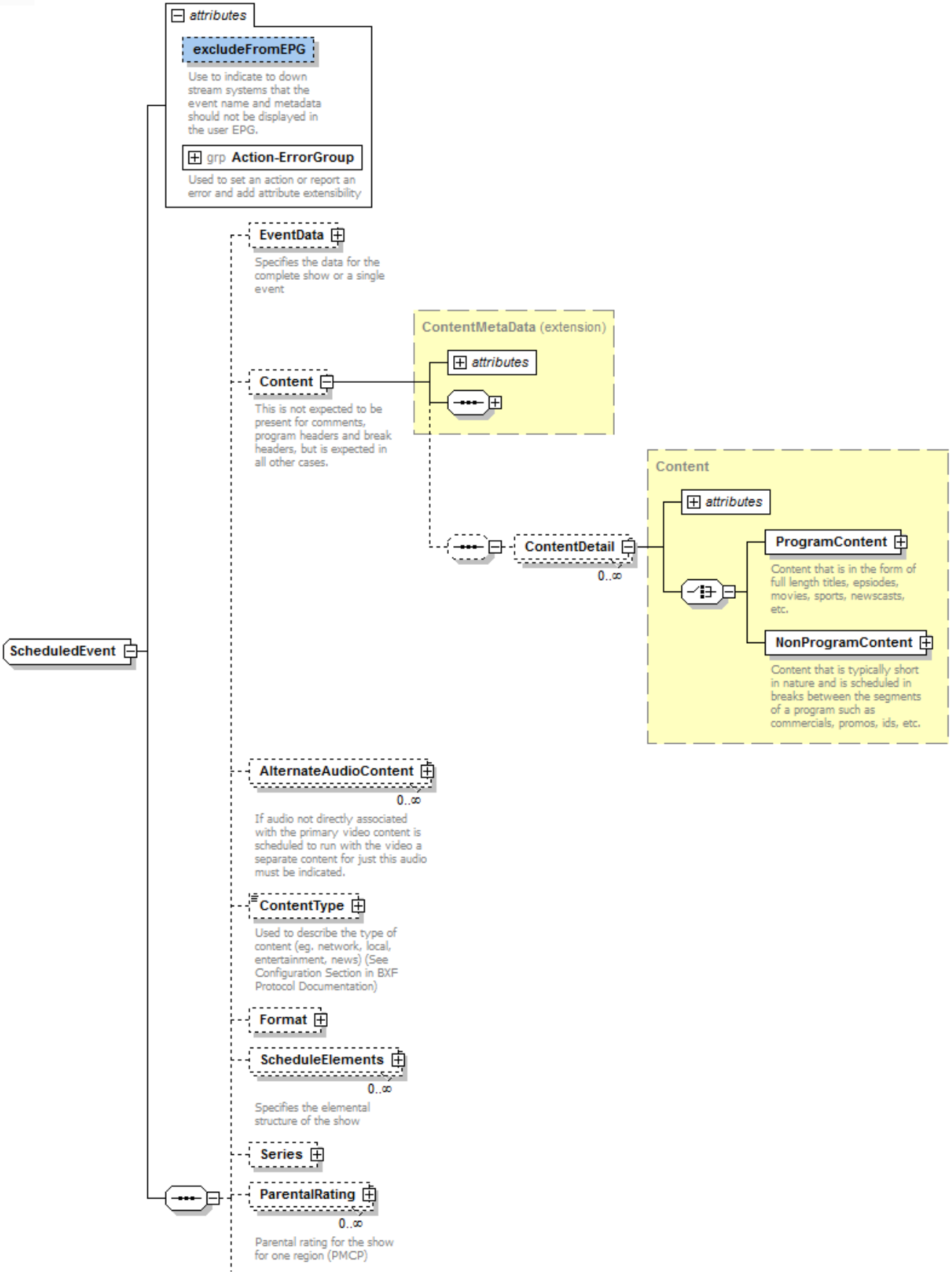
Text representation:

```
<xs:complexType name="ScheduledEvent">
```

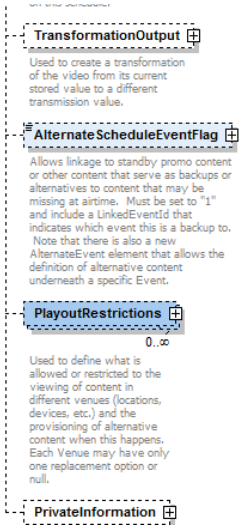
```
    <xs:sequence>
        <xs:element name="EventData" type="EventData" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Specifies the data for the complete show or a single
event</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Content" minOccurs="0">
            <xs:annotation>
                <xs:documentation>This is not expected to be present for comments, program headers and
break headers, but is expected in all other cases.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:complexContent>
                    <xs:extension base="ContentMetaData">
                        <xs:sequence minOccurs="0">
                            <xs:element name="ContentDetail" type="Content" maxOccurs="unbounded"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="AlternateAudioContent" type="AlternateAudioContent" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>If audio not directly associated with the primary video content is
scheduled to run with the video a separate content for just this audio must be indicated.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ContentType" type="BxfText" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Used to describe the type of content (eg. network, local, entertainment,
news) (See Configuration Section in BXF Protocol Documentation)</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Format" minOccurs="0">
            <xs:complexType>
                <xs:choice>
                    <xs:element name="Formats" type="Format">
                        <xs:annotation>
                            <xs:documentation>The definition of the empty structure of a program as used
by the traffic system to construct a schedule grid.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="FormatId" type="Uuid">
                        <xs:annotation>
                            <xs:documentation>The unique identifier for a format</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                </xs:choice>
            </xs:complexType>
        </xs:element>
        <xs:element name="ScheduleElements" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>Specifies the elemental structure of the show</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence>
```

```
                        <xs:element name="EventData" type="EventData" minOccurs="0">
                            <xs:annotation>
                                <xs:documentation>Specifies the data for the complete show or a single event</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:choice minOccurs="0" maxOccurs="unbounded">
                            <xs:annotation>
                                <xs:documentation>Added a choice option to create the ability to supply multiple content elements to a single schedule element and maintain historical consistency.</xs:documentation>
                            </xs:annotation>
                            <xs:element name="Content">
                                <xs:annotation>
                                    <xs:documentation>This is not expected to be present for comments, program headers and break headers, but is expected in all other cases.</xs:documentation>
                                </xs:annotation>
                                <xs:complexType>
                                    <xs:complexContent>
                                        <xs:extension base="ContentMetaData"/>
                                    </xs:complexContent>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="MultiContent" type="Content">
                                <xs:annotation>
                                    <xs:documentation>**New** Intended in cases where you would have more than one program content embedded within a single program such as compilation programs, cartoons, etc.</xs:documentation>
                                </xs:annotation>
                            </xs:element>
                        </xs:choice>
                        <xs:element name="AlternateAudioContent" type="AlternateAudioContent" minOccurs="0" maxOccurs="unbounded">
                            <xs:annotation>
                                <xs:documentation>If audio not directly associated with the primary video content is scheduled to run with the video a separate content for just this audio must be indicated.</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="PrivateInformation" type="BxfPrivateInformation" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Series" type="Series" minOccurs="0"/>
            <xs:element name="ParentalRating" type="BxfParentalRating" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>Parental rating for the show for one region (PMCP)</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="EiCode" type="EiCode" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Used for FCC Education/Information Children's code</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ContentPlayNumber" type="xs:positiveInteger" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>The play number value relative to the contract used to authorize the content or the total times the content has been used on the schedule.</xs:documentation>
```

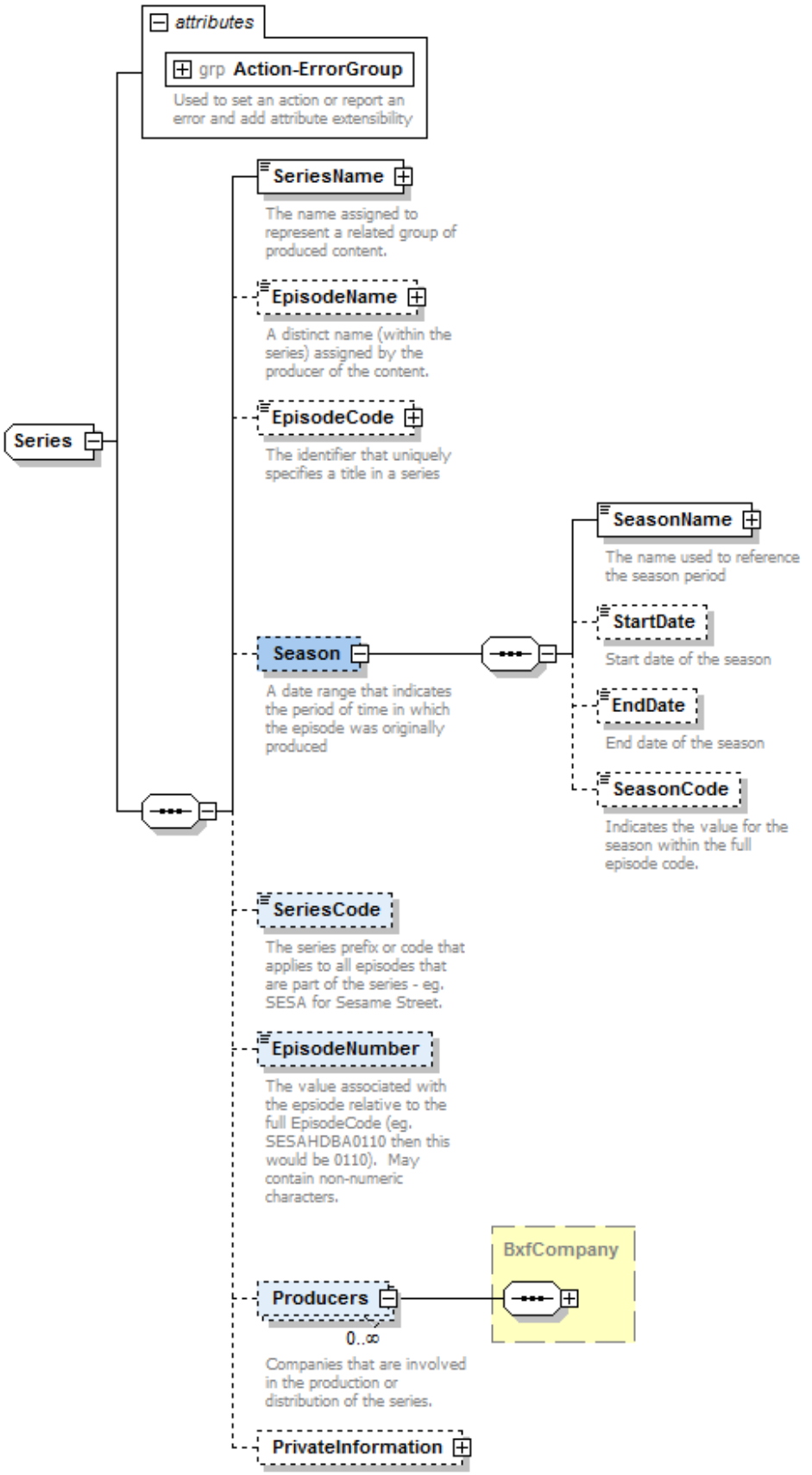
```
            </xs:annotation>
        </xs:element>
        <xs:element name="PremiereFlag" type="xs:boolean" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Used to indicate that this is the first airing of the content on this
schedule.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="LastPlayFlag" type="xs:boolean" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Used to indicate that this is the last airing of the content on this
schedule.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="TransformationOutput" type="BaseMedia" minOccurs="0">
            <xs:annotation>
                <xs:documentation>Used to create a transformation of the video from its current stored value
to a different transmission value.</xs:documentation>
            </xs:annotation>
        </xs:element>
```



```
            <xs:element name="AlternateScheduleEventFlag" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Allows linkage to standby promo content or other content that serve as backups or alternatives to content that may be missing at airtime.  Must be set to "1" and include a LinkedEventId that indicates which event this is a backup to.  Note that there is also a new AlternateEvent element that allows the definition of alternative content underneath a specific Event.</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:boolean">
                            <xs:attribute name="LinkedEventId" type="Uuid" use="required">
                                <xs:annotation>
                                    <xs:documentation>Must match to an existing EventId defined earlier</xs:documentation>
                                </xs:annotation>
                            </xs:attribute>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
            <xs:element name="PlayoutRestrictions" type="PlayoutRestrictions" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>Used to define what is allowed or restricted to the viewing of content in different venues (locations, devices, etc.) and the provisioning of alternative content when this happens.  Each Venue may have only one replacement option or null</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="PrivateInformation" type="BxfPrivateInformation" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="ExcludeFromEPG" type="xs:boolean">
            <xs:annotation>
                <xs:documentation>Use to indicate to down stream systems that the event name and metadata should not be displayed in the user EPG</xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attributeGroup ref="Action-ErrorGroup"/>
    </xs:complexType>
```

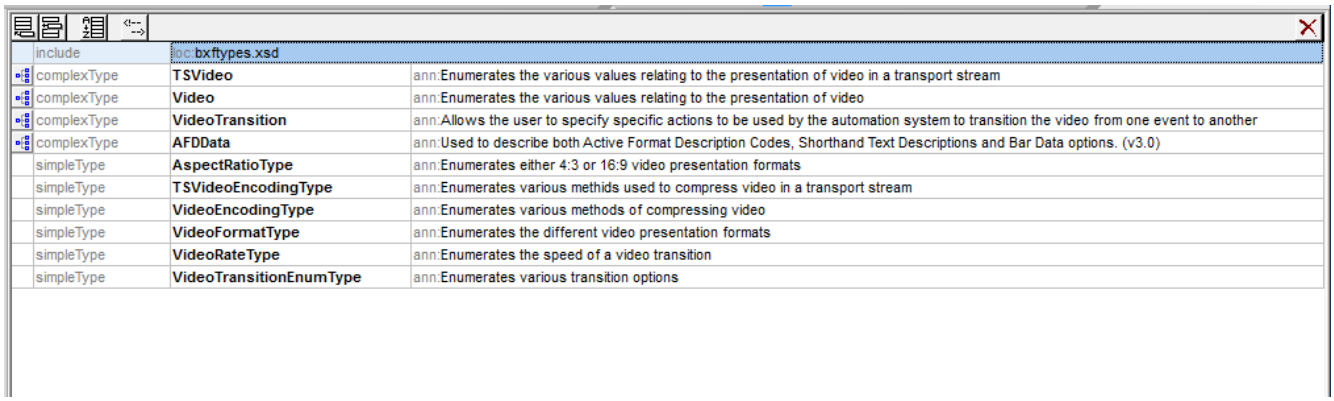


```
<xs:element name="SeasonCode" type="xs:string" minOccurs="0">
	<xs:annotation>
		<xs:documentation>Indicates the value for the season within the full episode code.</xs:documentation>
	</xs:annotation>
</xs:element>

<xs:element name="SeriesCode" type="xs:string" minOccurs="0">
	<xs:annotation>
		<xs:documentation>The series prefix or code that applies to all episodes that are part of the series - eg. SESA for Sesame Street.</xs:documentation>
	</xs:annotation>
</xs:element>
<xs:element name="EpisodeNumber" type="xs:string" minOccurs="0">
	<xs:annotation>
		<xs:documentation>The value associated with the epsiode relative to the full EpisodeCode (eg. SESAHDBA0110 then this would be 0110).  May contain non-numeric characters.</xs:documentation>
	</xs:annotation>
</xs:element>
<xs:element name="Producers" type="BxfCompany" minOccurs="0" maxOccurs="unbounded">
	<xs:annotation>
		<xs:documentation>Companies that are involved in the production or distribution of the series.</xs:documentation>
	</xs:annotation>
</xs:element>
```



3.2.31 Video.XSD

Contains:



include	bxftypes.xsd	
complexType	TSVideo	ann:Enumerates the various values relating to the presentation of video in a transport stream
complexType	Video	ann:Enumerates the various values relating to the presentation of video
complexType	VideoTransition	ann:Allows the user to specify specific actions to be used by the automation system to transition the video from one event to another
complexType	AFDData	ann:Used to describe both Active Format Description Codes, Shorthand Text Descriptions and Bar Data options. (v3.0)
simpleType	AspectRatioType	ann:Enumerates either 4:3 or 16:9 video presentation formats
simpleType	TSVideoEncodingType	ann:Enumerates various methids used to compress video in a transport stream
simpleType	VideoEncodingType	ann:Enumerates various methods of compressing video
simpleType	VideoFormatType	ann:Enumerates the different video presentation formats
simpleType	VideoRateType	ann:Enumerates the speed of a video transition
simpleType	VideoTransitionEnumType	ann:Enumerates various transition options

Enumerates the various values relating to the presentation of video.

3.2.31.1 BXF 4.0 Changes

Description of change:
Under AFDData there is an element for AFDCode which previously used a restriction for entry based on “xs:boolean” which was incorrect as it did not allow for the proper entry of code “0001” through “1111” as is required by SMPTE ST2016-1:2009. To fix this, the restriction was changed to a xs:positiveInteger and a new pattern was implemented to restrict the entry to “0’s and 1’s” and requires four charatcers.

Text representation:

```
<xs:element name="AFDCode">
                <xs:annotation>
                    <xs:documentation>Four bits that indicate the code for the Active Format Description as referenced by SMPTE ST2016-1:2009.</xs:documentation>
                </xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:positiveInteger">
                        <xs:pattern value="[0|1][0|1][0|1][0|1]"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
```