SMPTE 2021-0:2012
Revision of
SMPTE 2021-0:2009

SMPTE ROADMAP

Broadcast Exchange Format — Roadmap for the 2021 Document Suite

Document Roadmap

The SMPTE 2021 suite of documents specifies the Broadcast eXchange Format (BXF). This particular document (SMPTE 2021-0:2012) serves as a roadmap to the entire suite of 2021 documents, as well as other related files.

Documents in the SMPTE 2021 Suite

SMPTE ST 2021-1 – Requirements and Informative Notes (Normative and Informative Sections)

This document serves as the base standard document for the 2021 suite. It includes a comprehensive set of definitions, discussion of system data flow, security, configuration, and a large set of informative notes. This document is critical to anyone implementing BXF, as it is referenced by all other documents in this suite.

The distribution of SMPTE ST 2021-1 also includes:

Schema Definition File Collection (2021-1a-2012 Schema.zip)

This is the collection of XSD files which are required to implement BXF.

HTML Schema Documentation (2021-1b-2012 Schema HTML.zip)

This is a visual representation of the entire BXF schema, including all XSDs. It provides a graphical, non technical view into the structure of the schema from top to bottom, including relationships between portions of the schema, as well as complete documentation of all schema attributes and elements.

SMPTE ST 2021-2 – Protocol (Entirely Normative)

This document specifically and exclusively deals with issues surrounding protocol as it relates to BXF. It addresses the nature of BXF TCP/IP connections, sending of messages, dealing with inactive connections, initiation of servers, timeout of services, encryption of BXF messages, message formats, as well as file-based transport.

SMPTE EG 2021-3 – Engineering Guideline – BXF Use Cases (Entirely Informative)

This document provides a rich set of use cases, illustrating the use of BXF in real-world scenarios, complete with XML samples. The use cases cover such examples as: schedule, dub order, purge order, record order, transfer order, content notification, queries, invoking schedules, heartbeat messages, as runs, playlist updates, and acquisition failures. This document is intended to assist those wishing to implement BXF in specific ways.

Copyright © 2012 by THE SOCIETY OF
MOTION PICTURE AND TELEVISION ENGINEERS
3 Barker Avenue, White Plains, NY 10601
(914) 761-1100

Approved
March 14, 2012

SMPTE EG 2021-4 – Engineering Guideline – BXF Schema Documentation (Entirely Informative)

This document provides documentation of each of the schema definition files included with 2021 (the XSDs). It covers all of the XSD files, and explains all of the attributes and elements contained in each.

SMPTE RP 2021-9 – Recommended Practice (Normative and Informative Sections)

This document was created to assist BXF implementers who have read the other documents in the 2021 suite to create interoperable implementations. It covers such topics as: wellformedness and validation, processing model, general conventions related to various messages, as well as transport protocol (both connection-based and file-based). Perhaps the most useful portion of this Recommended Practice is its enumeration of message types that are typically used by different types of applications. If you are implementing BXF for an automation system, for instance, you can go directly to that portion of the document to see what parts of BXF are specifically applicable to your system. The document then goes into each major subset of BXF (configuration, content, content transfer, format, and schedule) in detail. The document also includes annexes covering: supported extensions, message type usages, error handling, and protocol transport.

Changes Contained In BXF 2.0

BXF 2.0 consists of a collection of four significant schema updates, adding new capabilities required by implementers. It is important to note that these changes are all backward-compatible. Existing BXF implementations will continue to operate as they always have. A new set of XSD (schema definition files) as well as XML samples is included in the 2021 package.

Below is an outline of the changes included within BXF 2.0.

1. Change to all XSD headers for this new release:

In order to distinguish BXF 2.0 schemas from those of BXF 1.0, the headers in each of the BXF XSD files have been updated to reflect the new version, as well as a new location for the BXF 2.0 schemas.

Old Version of all headers:

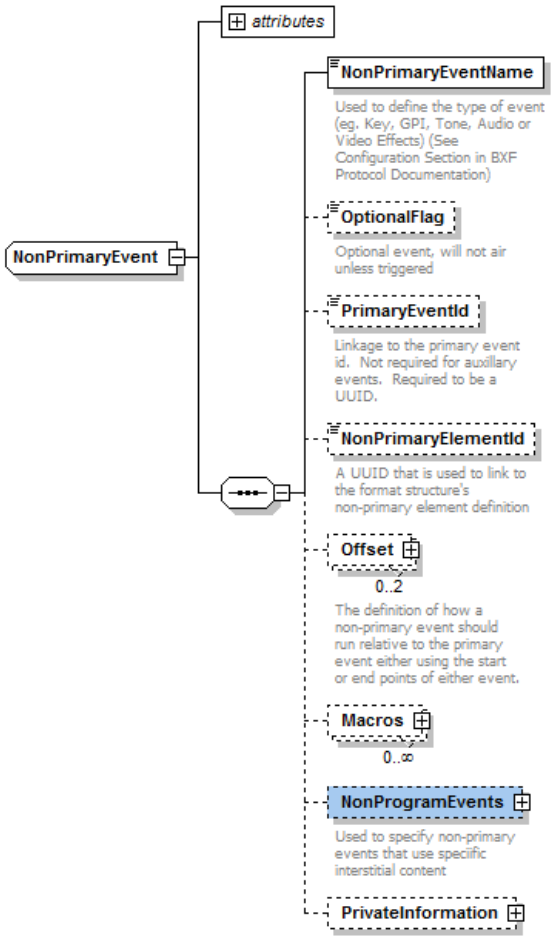
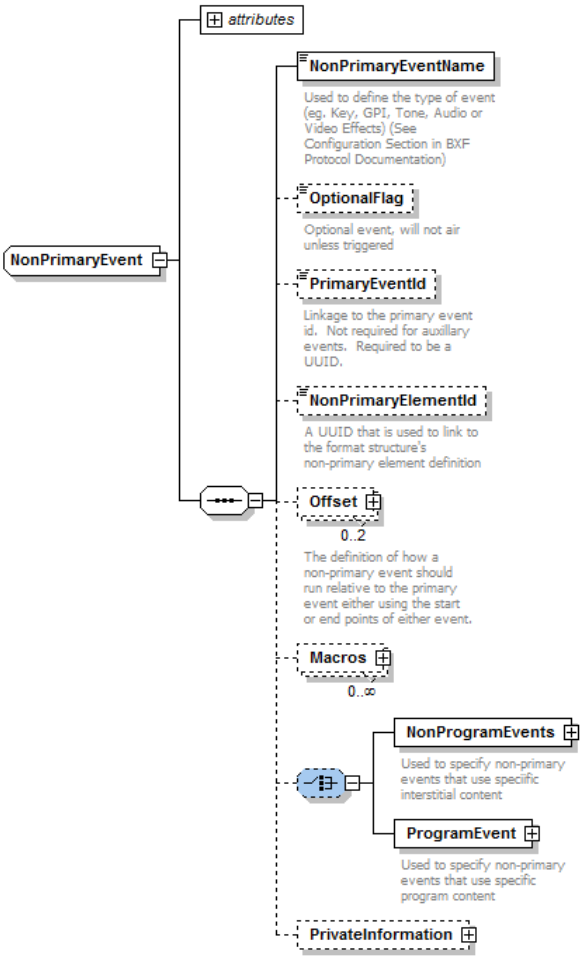
```
<!-- Copyright 2008 Society of Motion Picture and Television Engineers. All rights reserved. -->
<xs:schema xmlns="http://smpte-ra.org/schemas/2021/2008/BXF"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:pmcp="http://www.atsc.org/XMLSchemas/pmcp/2007/3.1" targetNamespace="http://smpte-
ra.org/schemas/2021/2008/BXF" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.000">
```

New Version of all headers:

```
<!-- Copyright 2012 Society of Motion Picture and Television Engineers. All rights reserved. -->
<xs:schema xmlns="http://smpte-ra.org/schemas/2021/2012/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/2021/2012/BXF" elementFormDefault="qualified" attributeFormDefault="unqualified"
version="2.000">
```

2. Addition of support for sponsored secondary events

A new structure has been added within the BXF nonprimaryevent.xsd schema definition. This allows a non-primary event to be either a standard non-advertising related event (as was allowed in BXF 1.0), or an advertising-related event (indicated in the schema as a NonProgramEvent). This allows the scheduling of a non-primary program event that plays concurrently with the primary event being played. An example would be the squeezed-back end credits of a program (non-primary event) that airs concurrently with the primary event start of the following program.

<table>
<tr><th colspan="2">nonprimaryevent.xsd</th></tr>
<tr><th>Version 1.0</th><th>Version 2.0</th></tr>
<tr><td colspan="2">New Version supports a choice between NonProgramEvents and ProgramEvent</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>

```
<xs:element name="NonProgramEvents"
type="NonProgramEvent" minOccurs="0">
        <xs:annotation>
                <xs:documentation>Used to
specify non-primary events that use speciific
interstitial content</xs:documentation>
        </xs:annotation>
</xs:element>
```

</td><td>

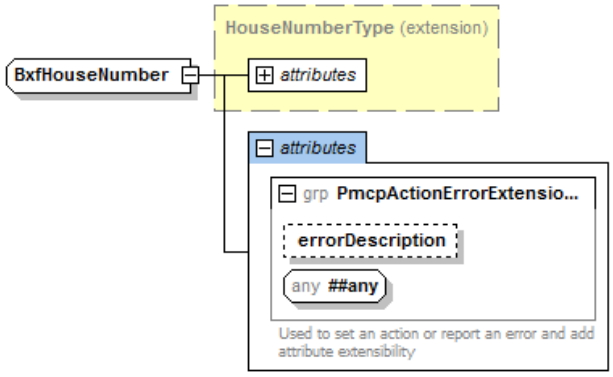
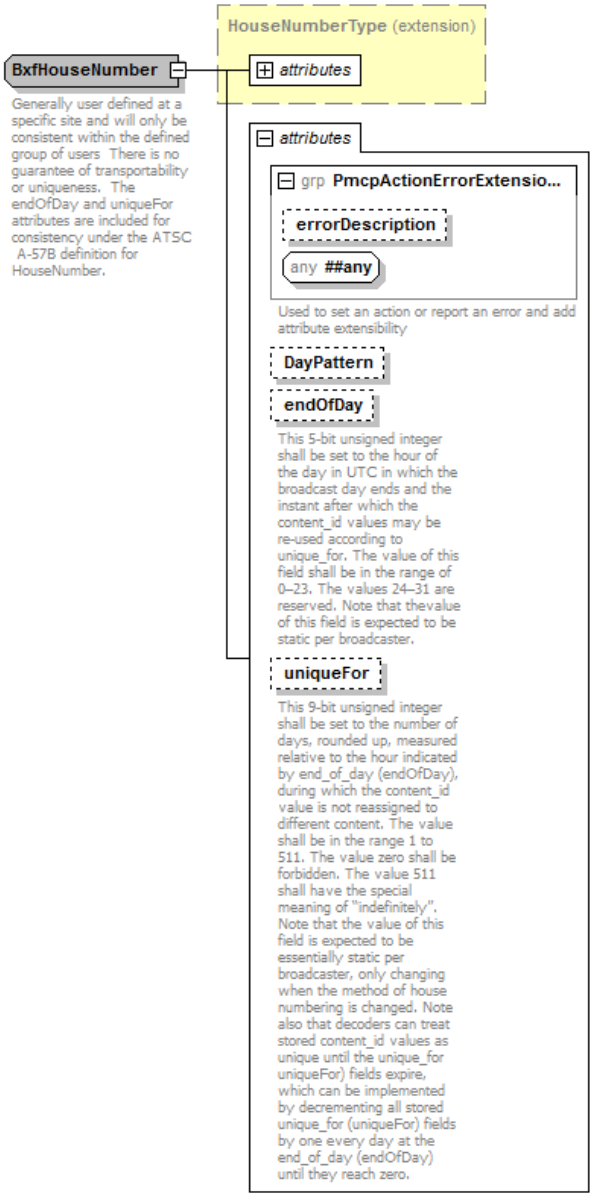
```
<xs:choice minOccurs="0">
        <xs:element name="NonProgramEvents"
type="NonProgramEvent">
                <xs:annotation>
                        <xs:documentation>Used to
                specify non-primary events that use
                speciific interstitial
                content</xs:documentation>
                </xs:annotation>
```

</td></tr>
</table>

	</xs:element> <xs:element name="ProgramEvent" type="ProgramEvent"> <xs:annotation> <xs:documentation>Used to specify non-primary events that use specific program content</xs:documentation> </xs:annotation> </xs:element> </xs:choice>

3. Support for Reuse of House Numbers

It is a reality that house numbers are re-used in broadcast operations worldwide. The ATSC's A/57B standard accommodates this fact with support for a "duration of effectiveness" for house numbers. In BXF 2.0, support has been added to the bxfcontentid schema definition file to include four new attributes to allow for support of this.

<table>
<tr><th colspan="2">bxfcontentid.xsd</th></tr>
<tr><th>Version 1.0</th><th>Version 2.0</th></tr>
<tr><td colspan="2">New Version supports the ability to set the duration of effectiveness for a house number and to make it compatible with A/57's definitions.</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td><xs:complexType name="BxfHouseNumber">
<xs:complexContent>
<xs:extension</td><td><xs:complexType name="BxfHouseNumber">
<xs:annotation>
<xs:documentation>Generally user defined</td></tr>
</table>

```
base="pmcp:HouseNumberType">
        <xs:attributeGroup
ref="PmcpActionErrorExtensionGroup"/>
        </xs:extension>
</xs:complexContent>
</xs:complexType>
```

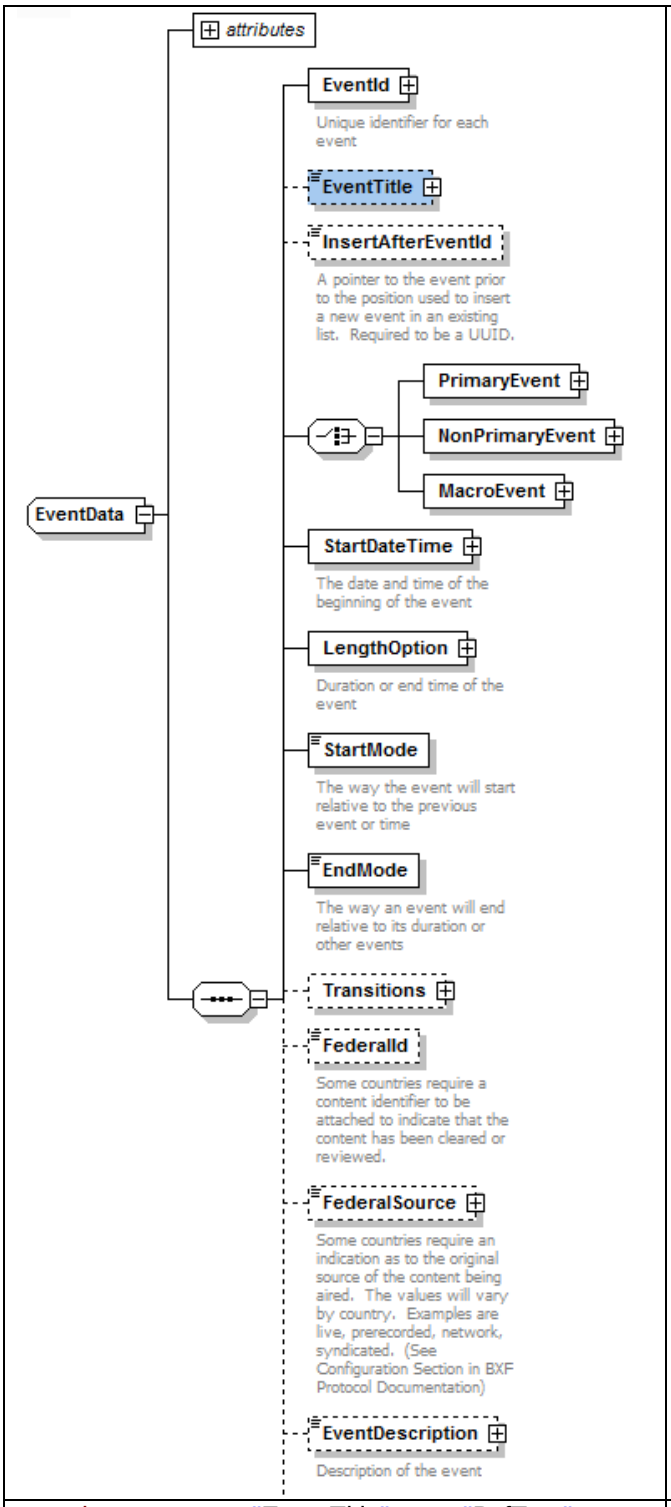
```
at a specific site and will only be consistent within
the defined group of users  There is no guarantee of
transportability or uniqueness.  The endOfDay and
uniqueFor attributes are included for consistency
under the ATSC  A-57B definition for
HouseNumber.</xs:documentation>
</xs:annotation>
<xs:complexContent>
        <xs:extension
base="pmcp:HouseNumberType">
        <xs:attributeGroup
ref="PmcpActionErrorExtensionGroup"/>
        <xs:attribute name="DayPattern"
type="DayPattern"/>
        <xs:attribute name="endOfDay" fixed="5">
        <xs:annotation>
        <xs:documentation>This 5-bit unsigned
integer shall be set to the hour of the day in UTC in
which the broadcast day ends and the instant after
which the content_id values may be re-used
according to unique_for. The value of this field shall
be in the range of 0–23. The values 24–31 are
reserved. Note that thevalue of this field is expected
to be static per broadcaster.</xs:documentation>
        </xs:annotation>
<xs:simpleType>
        <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="23"/>
</xs:restriction>
</xs:simpleType>
        </xs:attribute>
        <xs:attribute name="uniqueFor" fixed="9">
        <xs:annotation>
        <xs:documentation>This 9-bit unsigned
integer shall be set to the number of days, rounded
up, measured relative to the hour indicated by
end_of_day (endOfDay), during which the
content_id value is not reassigned to different
content. The value shall be in the range 1 to 511.
The value zero shall be forbidden. The value 511
shall have the special meaning of “indefinitely”. Note
that the value of this field is expected to be
essentially static per broadcaster, only changing
when the method of house numbering is changed.
Note also that decoders can treat stored content_id
values as unique until the unique_for uniqueFor)
fields expire, which can be implemented by
decrementing all stored unique_for (uniqueFor)
fields by one every day at the end_of_day
(endOfDay) until they reach
zero.</xs:documentation>
        </xs:annotation>
<xs:simpleType>
        <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="1"/>
```

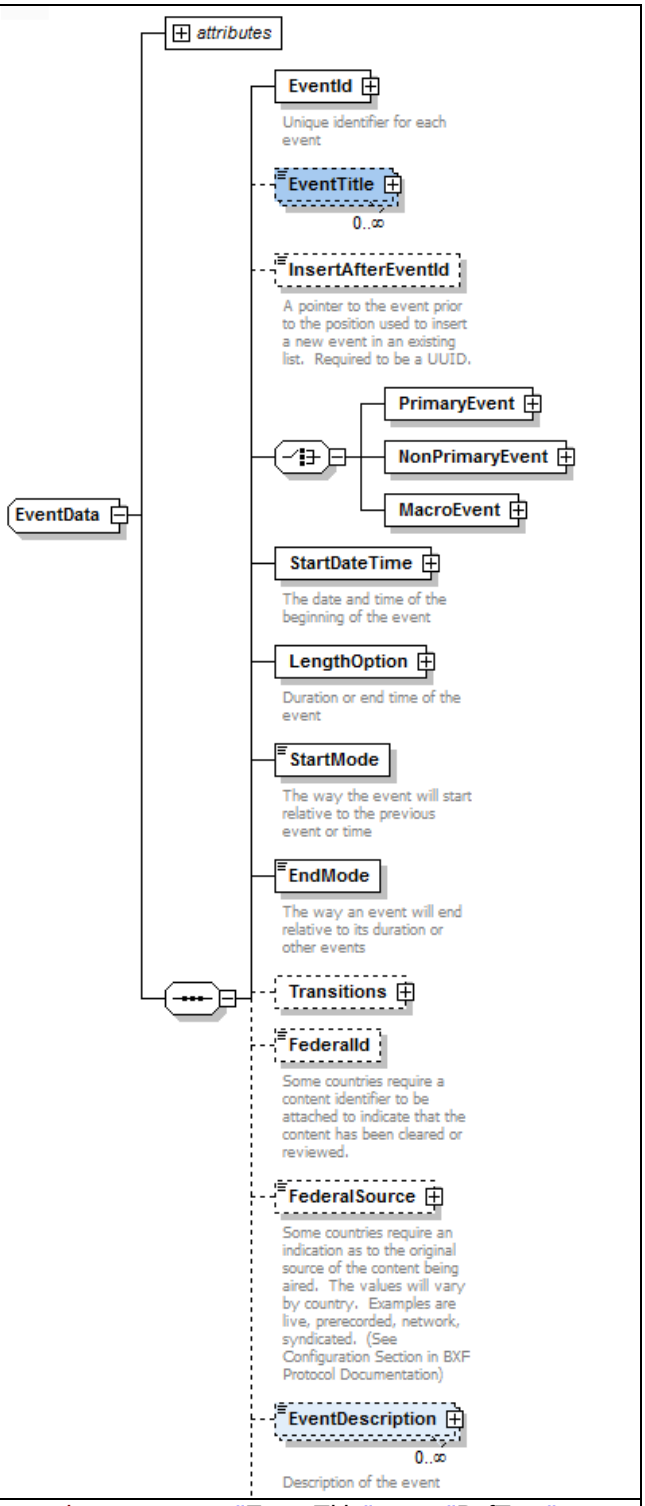
	<xs:maxInclusive value="511"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:extension> </xs:complexContent> </xs:complexType>

4. Multi Language Support

Many users of BXF must deal with on-air material that exists in multiple languages. BXF 1.0 only allowed for a single Event Title and Event Description on each event. This meant that a single language had to be chosen. BXF 2.0 adds support for multiple Event Titles and Event Descriptions, allowing the title and description to be included in multiple languages.

eventdata.xsd	
Version 1.0	**Version 2.0**
New Version supports the ability to enter multiple EventTitles and EventDescriptions so that more than one language can be included in a single message.	






```
<xs:element name="EventTitle" type="BxfText"
minOccurs="0"/>

<xs:element name="EventDescription"
type="BxfText" minOccurs="0">
        <xs:annotation>
        <xs:documentation>Description of the
```

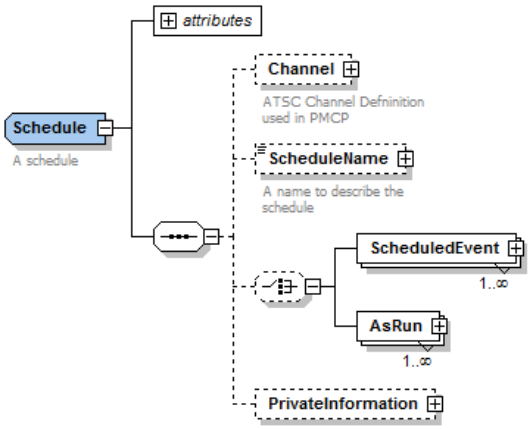
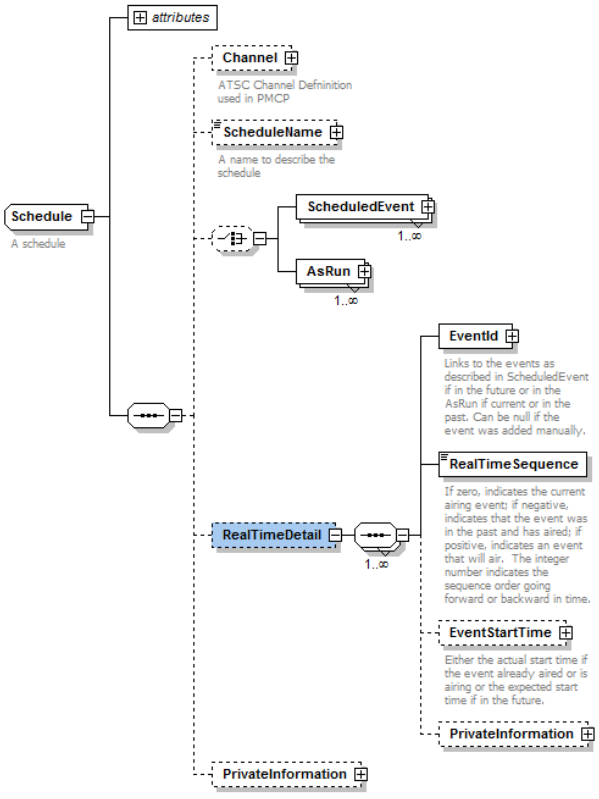
```
<xs:element name="EventTitle" type="BxfText"
minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="EventDescription"
type="BxfText" minOccurs="0"
maxOccurs="unbounded">
        <xs:annotation>
```

event</xs:documentation> </xs:annotation> </xs:element>	<xs:documentation>Description of the event</xs:documentation> </xs:annotation> </xs:element>

5. Ability to communicate events surrounding currently airing event

As BXF has introduced the ability for systems to exchange data relating to schedules and as runs in near real time, the need has arisen for a simple structure to communicate a quick update, showing events just aired, the event currently airing, and events about to air. BXF 2.0 adds this ability to the schedule schema definition via a new node (RealTimeDetail). Using this new structure the current event, "x" events immediately prior to the one currently airing (a negative value in the RealTimeSequence node), as well as "y" events immediately following the one currently airing (a positive value in the RealTimeSequence node) can be communicated.

schedule.xsd	
Version 1.0	**Version 2.0**
New Version supports the ability to handle real time actions on the automation event process. This would allow the communication of the prior event completed, the event that is currently running and the event that is next up.	
	
	<xs:element name="RealTimeDetail" minOccurs="0"> <xs:complexType> <xs:sequence maxOccurs="unbounded"> <xs:element name="EventId" type="EventExtId"> <xs:annotation>

	<xs:documentation>Links to the events as described in ScheduledEvent if in the future or in the AsRun if current or in the past. Can be null if the event was added manually. </xs:documentation> </xs:annotation> </xs:element> <xs:element name="RealTimeSequence" type="xs:integer"> <xs:annotation> <xs:documentation>If zero, indicates the current airing event; if negative, indicates that the event was in the past and has aired; if positive, indicates an event that will air. The integer number indicates the sequence order going forward or backward in time. </xs:documentation> </xs:annotation> </xs:element> <xs:element name="EventStartTime" type="BxfDateTime" minOccurs="0"> <xs:annotation> <xs:documentation>Either the actual start time if the event already aired or is airing or the expected start time if in the future. </xs:documentation> </xs:annotation> </xs:element> <xs:element name="PrivateInformation" type="BxfPrivateInformation" minOccurs="0"/> </xs:sequence> </xs:complexType> </xs:element>