SMPTE ST 2021-1:2012
Revision of
SMPTE ST 2021-1:2009

SMPTE STANDARD

Broadcast Exchange Format (BXF) — Requirements and Informative Notes

Approved
March 14, 2012

Table of Contents

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 Part XIII of its Administrative practices.

SMPTE ST 2021-1 was prepared by Technology Committee 34CS.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

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.

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

Background

To understand the scope of BXF, a little background is helpful. The genesis of BXF can be traced to the need for a consistent yet flexible means for exchanging schedule, as run, and content metadata between Traffic and Automation systems. Literally hundreds of proprietary, fixed interfaces and protocols have been created over the past 20 years or so between these two types of systems. Vendors who create and develop these systems were invited to share ideas in search of a better way to facilitate the exchange of data between systems. The group quickly embraced Content Distribution and Program Management vendors as well, as they too were seeking a form of standardization and improvement in their interfacing efforts.

BXF supports the exchange of single or multiple records at one time, over a variety of transport mechanisms.

While endeavoring to be a comprehensive protocol, it is acknowledged that it is possible that additional data elements, or data elements that apply to a few specific systems, may need to be exchanged. For this reason,

Private Information structures have been placed at various points in the schema to allow vendors the flexibility to add data elements.

It is expected that BXF could be extended in the future, as warranted, and it has been designed with this in mind.

BXF is intended to complement, not replace, the following:

- Advanced Authoring Format
- Material eXchange Format
- MDP – Media Dispatch Protocol

It is acknowledged that some may see an area of overlap in BXF's ability to request the movement of content with other standards, such as MDP. However, MDP itself does not include support for these specific requests. Instead,

BXF incorporates aspects of several pre-existing standards into its schema, such as:

- Programming Metadata Communication Protocol (ATSC)
- International Standard Audiovisual Number (ISO)
- Ad-ID (AAAA/ANA)

The following figure illustrates the scope of BXF within a broadcast facility. Those areas outside of the BXF box are considered outside the intended scope of the BXF protocol and this document.

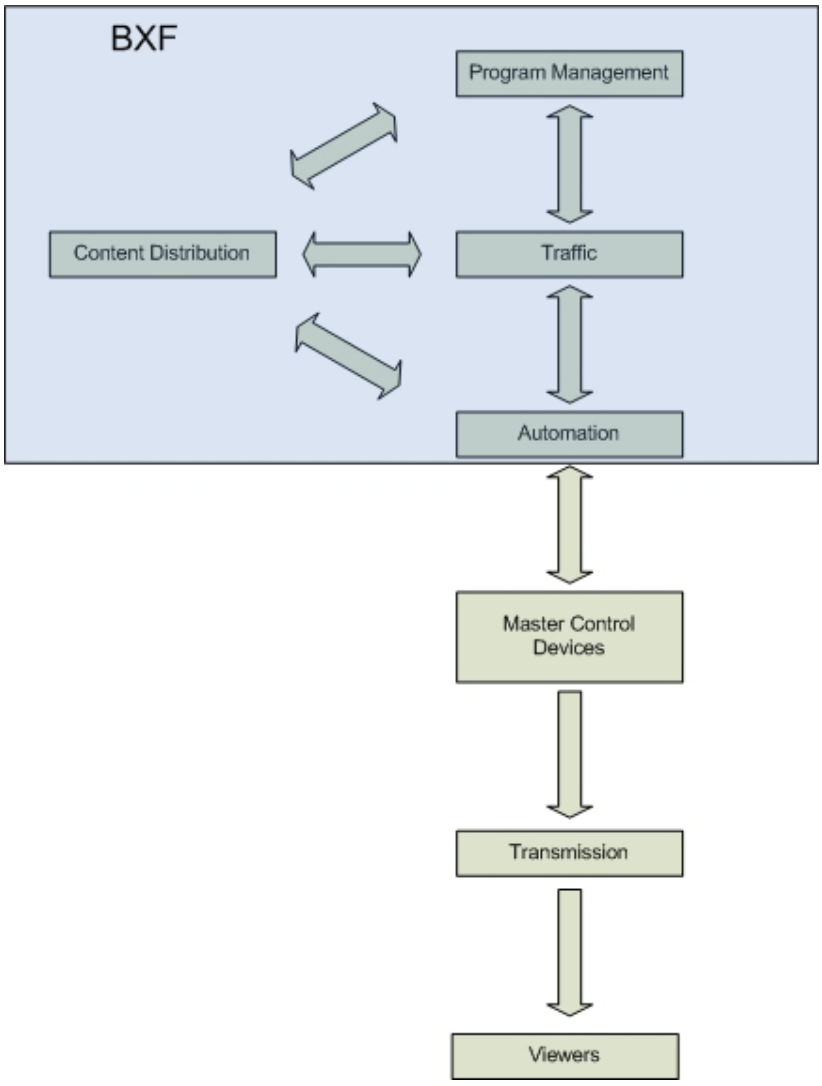


Figure 1 – Scope of BXF

Broadcast Exchange Format (BXF)

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 document serves as the master document for the 2021 suite (thus its Part 1 designation). It includes general BXF details as well as informative notes. SMPTE 2021-0 provides a Roadmap for the 2021 Document Suite

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.

2 Conformance Notation

Documents consist of normative text and, optionally, informative text. Normative text is that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in a Standard, Recommended Practice, Amendment, Addendum, or Corrigendum, is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

Normative references are external documents referenced in normative text that are indispensable to the user. Bibliographic references are references made in informative text or are those otherwise not indispensable to the user. Normative references shall conform to the types and procedures specified in the Engineering Administrative Practices.

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted. The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding

others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to an Engineering Document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

3 Document Elements

This document is comprised of the following elements, which form an integral piece of this Standard. Additionally, the schema files may be found at http://smpte-ra.org/schemas/2021/2012/BXF. (Accessible only by appropriately-designed software applications, for schema validation. Not intended to be human-accessible.)

a) Prose document st2021-1-2012.pdf (this file) [Normative]

b) XML schema s2021-2012.xml [Normative]

c) HTML schema guide s2021-2012.html [Informative]

4 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

- XML Schema Part 1: Structures Second Edition, W3C Recommendation 28 October 2004, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/
- XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/ Extensible Markup Language (XML) 1.0 (Second Edition)
- W3C Recommendation, 6 October 2000, http://www.w3.org/TR/2000/REC-xml-20001006
- XML Schema, W3C Recommendation, 2 May 2001, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502
- A/76B “Programming Metadata Communication Protocol, Revision B”, Advanced Television Systems Committee, Washington, DC, January, 2008
- ISO 15706-2:2007, Information and documentation – International Standard Audiovisual Number (ISAN) – Part 2: Version identifier
- “XML Path Language”, W3C Recommendation 16 November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116
- SMPTE ST 258:2004, Television – Transfer of Edit Decision Lists
- A Universally Unique IDentifier (UUID) URN Namespace P. Leach, M. Mealling, R. Salz, 2005 IETF RFC4122 http://www.ietf.org/rfc/rfc4122.txt

- IETF RFC3986, Uniform Resource Identifiers (URI): Generic Syntax T. Berners-Lee, et al. The Internet Society, 2005. http://www.ietf.org/rfc/rfc3986.txt
- IETF RFC3066, Tags for the Identification of Languages, H. Alvestrand, The Internet Society, 2001. http://www.ietf.org/rfc/rfc3066.txt
- MPEG-2 ISO/IEC 13818-2:2000, Information technology — Generic Coding of Moving Pictures and Associated Audio Information: Video
- A/65C, “Program and System Information Protocol for Terrestrial Broadcast and Cable, Revision C, with Amendment No. 1”, Advanced Television Systems Committee, Washington, DC, 9 May 2006
- CEA-708-D, Digital Television (DTV) Closed Captioning, Consumer Electronics Association

5 Definitions

The following definitions apply within the BXF protocol standard.

Word	Element/Attribute Name	Definition
Advertiser Name	AdvertiserName	Time in a broadcast schedule may be sold to a third party for the purpose of advertising products, goods, or services. These are commonly referred to as commercials, and the third party that purchased the time for the commercial is an Advertiser.
Agency	Agency	The entity that manages placement of commercials, and in some cases the production of commercials
Alternate Audio Content	AlternateAudioContent	If audio that is not directly associated with the primary video content is scheduled to run with the primary video, a separate piece of content for just this audio must be indicated. This includes audio that may be provided for the purpose of accessibility issues.
As Run (Schedule)	AsRun CompleteAsRun BasicAsRun	A term typically applying to broadcast playout schedules referring to the exact events that aired during a specific period of time, usually a broadcast day and also includes errors concerning scheduled content that did not air or was aired improperly.
Asset Server	AssetServer	Asset server is a term that describes a server where commercial and program content is stored.
Authorization List	AuthorizationList	For cable operations a list of cable headend locations that are authorized to present specific commercial content.
Avail Number	AvailNumber	A count of the number of format positions that are allowed to contain commercial content.
Billing Reference Code	BillingReferenceCode	A unique reference code for each billable event that typically links back to the system that created the event, and is used in the reconciliation of the as run against the schedule.
Channel	Channel	A definition used to describe the means by

		which a schedule is broadcast.
Configuration	Configuration	There are several schema elements that are not specifically enumerated in the schema and must be configured between the two systems before the elements can be used.
Constraints	Constraints	A set of rules that restricts the placement of commercial content on a schedule.
Content	Content	Content may contain multiple essence types, (e.g. audio, video, etc.)
Content Metadata	ContentMetaData	Information concerning a specific piece of content
Content Play Number	ContentPlayNumber	Each time a program is scheduled and aired by an entity it is given a play number that allows the entity to track how many times it has been aired.
Content Transfer	ContentTransfer	The movement or copying of content from one location to another.
Content Type	ContentType	Another way to categorize content into specific groupings (e.g. network, local, news, entertainment).
Day Pattern	dayPattern	When creating a schedule description, it is possible to define only part of a schedule that covers specific days and periods of time. The day pattern is used to describe this using Monday as the arbitrary first day of the week and having seven elements where each one represents one day of the week. For example, the number string 1111100 would represent M-F, 0000010 represents Sat, and 0000001 is Sun.
Dub Request (May sometimes be called “Copy”)	transferType:Duplication	A set of instructions that cause the specified content to be copied from its original location to a new destination.
Duration	Duration	Duration of content

E-I Code	E-ICode	Certain programs may be designated by the entity to conform to a governmental standard of being education and/or informational in nature for a specific age group, usually children and teens (FCC 06-143).
Elements	Elements	A stream of content can be divided into its individual elements (i.e. Commercials, program segments, graphics, etc).
Embargo Date	EmbargoDate	The first date content may be used
Embedded Non-program Content	EmbeddedNonProgramContent	This is another name for commercial content supplied by the distributor of the content and is considered part of a program segment rather than being broken out into its own event. When this happens, the commercial content is referred to as being embedded.
End Mode	EndMode	Each event can end using one of several possible options: Duration - when the value set for the event's duration is reached. Manual - when the operator ends the event via manual intervention. External - continue to run until an external event has taken place that triggers the end of the event.
EOM		End of Material is a point beyond which the content should not be played.
Episode	EpisodeName EpisodeCode	A single program of a series.

Event	EventData	An event is a period of time during which one or more types of content is displayed on the channel. Examples include: Single event: playback of a program segment that is uninterrupted. Single event: playback of a non-program type content (commercials). Single event with no duration: turning on the tower lights at dusk. Single event with no duration: the generation of a commercial cue tone. Multiple events: playback of a promotional message at the end of a movie while the credits are rolling.
Expendable	Expendable	A flag that is set on an event by the traffic management system to indicate that the event may be preempted.
Expiration Date	ExpirationDate	The last date and time when the content can be used on a schedule as determined by its contract or some other restriction.
Federal Source	FederalSource	A value that is set for each event as required by the local or federal government in order to categorize the event in some way.
First Air Date	FirstAirDate	The first date that the content is aired on a schedule.
Format Element Type	FormatElementType	Formats consist of either segments or breaks. Segments are placeholders for program content and breaks are placeholders for commercial content. Comments can also be added to formats but can not hold any type of content.
Format	Format	The definition of the empty structure of a program as used by the traffic system to construct a schedule grid.
Frame		A frame contains all of the lines of spatial information of a video signal required to make up one complete picture.
Frame Rate	frameRate	The number of frames per second

Genre	Genre	Programs may be grouped into separate categories for the convenience of organizing them. Genre usually represents the type of content based on the storyline or style of the content. (E.g. police drama, western, game show, children's show, etc.)
Heartbeat Message	messageType:Heartbeat	An XML message conforming to the BXF Schema definitions that is used by one system to query if another system is operational.
Hiatus Date Range	HiatusDateRange	A date range period after the embargo date, but before the expiration date during which the content can not be aired.
IgnoreAvail	IgnoreAvail	A flag to indicate that commercial content may not be aired in some locations while still being authorized to air in others based on external agreements.
JIP Flag	JipFlag	Set when a program is joined in progress rather than starting at its beginning.
Last Play Flag	LastPlayFlag	Set when a program is being aired for the last time.
Macro	MacroEvent	In order to facilitate the processing of a complex sequence of instructions to an automation system, a single macro name is substituted which may also have a set of parameters that define how the macro is to be processed by the device. The execution of the macro by the automation system triggers a predefined series of discreet events rather than requiring that each event be specifically called.
Media	Media	The physical means on which a piece of content is stored (i.e. Tape, dvd, and server).
Non-primary Elements	NonPrimaryElements	The individual components of a format that make up a non-primary event.
Non-primary Event	NonPrimaryEvent	An event that depends on a primary event and is scheduled relative to that event.
Non-program Content	NonProgramContent	Short form content that is interspersed between the program content.

Parental Rating	ParentalRating	A rating applied to most program content that allows parental control.
Physical Asset	PhysicalAsset	The storage of content on a physical medium such as tapes, CDs, DVDs, etc.
Playout Schedule		A list of events to be played in sequential order based on the instructions included as part of the schedule.
Preemption Warning	PreemptionWarning	A flag set by the traffic system to warn the operator of the automation system not to preempt the airing of the flagged content.
Premiere Flag	PremiereFlag	Set when a program is first aired by the entity.
Primary Duration	PrimaryDuration	The length of time assigned as the duration of a format element.
Primary Element ID	PrimaryElementId	A value assigned to each element of a format that can then be used to link it to a specific event.
Primary Event	PrimaryEvent	An event that does not depend upon any other event to happen.
Primary Offset	PrimaryOffset	Each format element is given an offset value in time to indicate when it starts relative to the start of the program.
Product Code	ProductCode	The category of the product.
Product Name	ProductName	The name of the product being advertised.
Program Contract	ProgramContract	Many programs are purchased or licensed from a third party and the program contract specifies the limitations on how that program may be aired by the entity.
Program Management		The process of managing a broadcast entity's library of content and coordinating the delivery of other media content from third party sources.
Program Content	ProgramContent	Content that is typically longer in duration and excludes non-program types.
Protection	Protection	In order to protect against a possible equipment failure, a protection source may be requested so

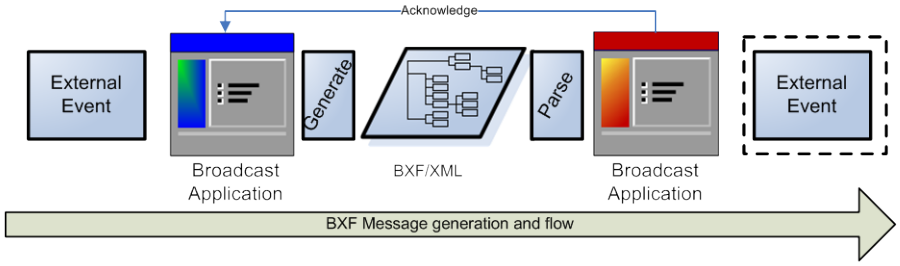
		that one or more identical copies of the primary content be running at the same time.
Purge Date	PurgeDate	The date when the content may be deleted from its storage location.
Purge Request	PurgeRequest	A set of instructions that determines when a system should delete content from its current location.
Record Schedule		A set of instructions that determines when a system expects to record content being sent to it, typically via satellite.
Routers	RouterSource	Routers are used to direct content from one location to another
Sales Contract	SalesContract	Commercial content is added to a schedule based on a sales contract which is the legal agreement between the agency or advertiser and the entity that airs the content. It includes a contract number and a number of contract lines.
Satellite	Satellite	Used to uniquely identify a specific orbiting communication satellite used to transfer content from one location to another over great distances.
Schedule Elements	ScheduleElements	Each schedule can be further divided into schedule elements which contain the details of each event that airs and the content information for each event.
Schedule ID	scheduleId	Each unique sequence of program and non-program events is assigned a unique number within the entity's broadcast domain to distinguish one from the other.
Schedule Name	scheduleName	Each sequence of events is usually assigned a unique name as a reference to the airing of the content.

Schedule Types	type	Schedules are either primary or alternate. A primary schedule is the normal planned schedule for a specific date and time. An alternate schedule is a backup to the primary schedule in cases when events are not predictable (e.g. used to manage rain outs or extra innings during a baseball game).
Season	Season	Some series extend over multiple years and each year, a new group of episodes is produced. These are grouped into seasons.
Segment Number	SegmentNumber	A format is used to describe the organization of program content and segments are the place holders for that content. Each segment is given a number in the structure to indicate its order.
Series	Series	Some program content is produced as a continuing storyline. Each program is then referred to as an episode of a series.
SOM	SOM	Start of Material is a point where content playback is to begin.
Spot Sales Classification	SpotSalesClassification	A way used to organize commercial content into sales groups.
Spot Type	SpotType	Used to delineate the various types of non-program type content. Examples include paid commercials, public service announcements (PSAs), promotional announcements (Promos), station identifications (IDs), barter commercials, and other customized values that can be configured.
Start Mode	StartMode	Each event can be started using one of several possible options: Follow - when the previous event finishes, start this event Fixed - start this event at the designated time, whether or not the previous event is finished Manual - wait until an operator triggers the event External - wait until an external event has taken place that triggers this event

Start of Broadcast Day		A time established by the broadcast entity as to when a new broadcast day starts.
Time Code	SmpteTimeCode	A label for the duration for content.
Total Avails	TotalAvails	The sum of the time created by format positions that are allowed to contain commercial content.
Traffic		The process of inserting commercial content into a schedule.
Traffic Caution Flag	TrafficCautionFlag	A flag set by the traffic system to warn the operator of the automation system not to change any of the conditions on the airing of the flagged content.
Transformation Output	TransformationOutput	Content may be encoded in various formats and it may be required to transcode the content from one format into another before being able to air the content.
Transitions	Transitions	The transition from one event to another. Multiple parameters may be set to describe these transitions.
UsagePolicy	UsagePolicy	These are rules that typically limit when or where a piece of content can be used by the entity.
XML Time	XmlTime	A specific time value using hours, minutes, seconds and milliseconds.

6 System Data Flow (Informative)

The following diagrams illustrate examples of information flow using BXF messages. Figure 2 – BXF Data Flow, Illustration A shows an example of BXF message processing integrated in end-to-end messages between two applications. Figure 3 – BXF Dataflow, Illustration B shows the peer-to-peer design of BXF messages, where any of multiple parties may initiate transactions.



BXF Data Flow

1. An external event will cause a Broadcast Application (Blue) to generate a pertinent message for another application
2. Blue application will manipulate internal data and extract it, generating an XML message (conforming to the BXF schema)
3. The message will transverse the BXF network to the target Broadcast Application (Red)
4. Red application will parse the BXF formatted message
5. The receiving application will signal the sending application an acknowledgement of the message receipt
6. The new data may cause an external event on the Red system

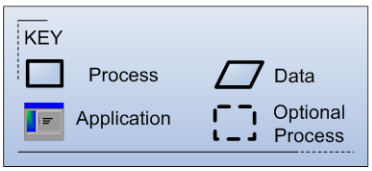
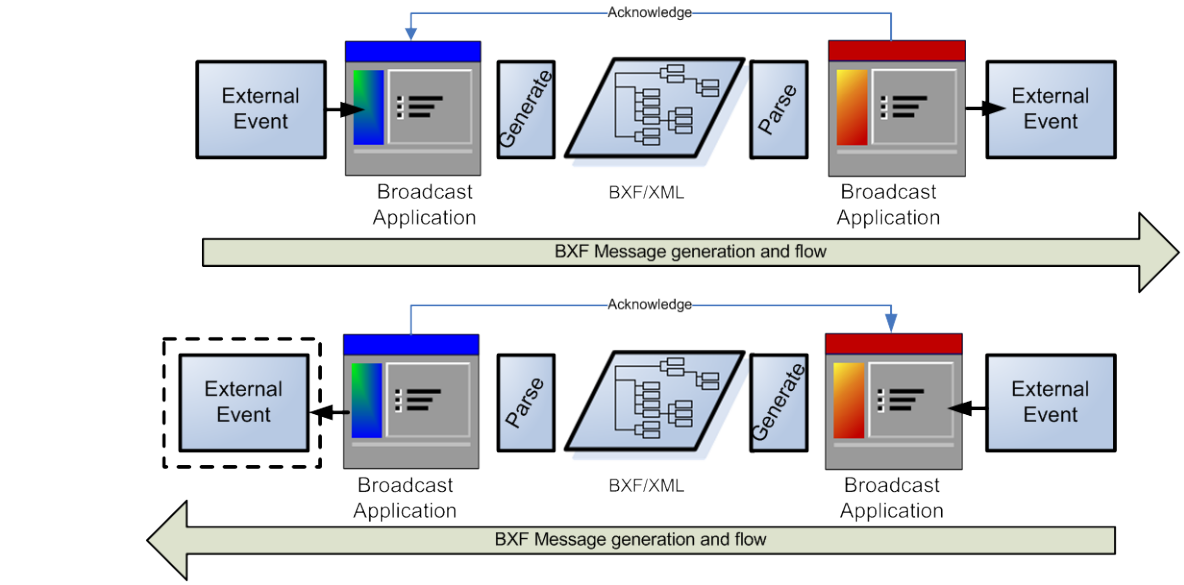


Figure 2 – BXF Data Flow, Illustration A



BXF Data Flow

1. An external event will cause a Broadcast Application (Blue) to generate a pertinent message for another application
2. Blue application will manipulate internal data and extract it, generating an XML message (conforming to the BXF schema)
3. The message will transverse the BXF network to the target Broadcast Application (Red)
4. Red application will parse the BXF formatted message
5. The receiving application will signal the sending application an acknowledgement of the message receipt
6. The new data causes an external event on the Red system
7. The Red application will generate a BXF message to the original Blue application in response

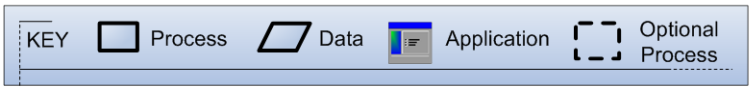


Figure 3 – BXF Dataflow, Illustration B

7 System Security (Informative)

The following statements apply to management of security and authentication in BXF messages.

- BXF is a messaging protocol. It does not define an encryption algorithm or prescribe the use of a specific security technology. BXF messages do not preclude the use of external encryption agents, port blocking, access control lists or other security schemes.
- BXF messages are XML documents that can be exchanged among BXF-capable systems independent of the transport medium. Examples of transport mediums include (wired or wireless) real-time exchange using direct connection like IP, Web Services, or SOAP, or off-line exchange using FTP or other file-based methods.
- BXF messages specify the use of a GUID whenever appropriate (as defined in the schema) to refer to systems, users, processes and events in order guard against message spoofing, although this mechanism is not intended to be the only one employed.
- BXF messages support journaling; journaling in this context means that BXF messages carry information that enables applications to record each transaction exchange between systems using BXF.
- Each BXF message can carry the information identified in the table below.

8 Configuration (Informative)

This section lists those elements in the BXF Schema which do not have specific enumerations. It is expected that values for these elements would be established between the two entities that plan to use the BXF Schema to communicate metadata changes and information.

Table 1 – Configuration Options

Configuration Options in BXF Schema based on version 2.00				
Configuration Name	**Filename**	**Elements or Attributes**	**Description**	**Examples**
AgencyCode	NonProgramDetail.xsd	AgencyCode	List of Agency Codes	
AlternateIdType	BxfContentId.xsd	idType	Type of alternate IDs	ISCI
AlternateIdSource	BxfContentId.xsd	authoritativeSource	The source of the content that is using the alternative ID	Traffic; Automation; CDS; Program Management
AssetName	Location.xsd	assetName	List of tape storage formats	BetaSp; DigiBeta; MiniDV; etc.
Classification	NonProgramEvent.xsd	Classification	List of program or spot classifications that may restrict the movement of a program or spot from one event to another by the operator.	
ContentType	ScheduledEvent.xsd	ContentType	List of Content Types	Network, local, entertainment, news
FederalSource	EventData.xsd	FederalSource	List of defined federal sources for broadcast content that will vary	Live, recorded, network

Configuration Options in BXF Schema based on version 2.00				
Configuration Name	**Filename**	**Elements or Attributes**	**Description**	**Examples**
			by country	
MacroName	Macro.xsd	MacroName	List of Macro Names	MACRO1; MACRO2
MessageDestination	BXFSchema.xsd	destination	Device or Application Names of the intended destination of a BXF message	Traffic; Automation; CDS ;Program Management
MessageOrigin	BXFSchema.xsd	origin	Device or Application Names of the system originating the BXF message	Traffic; Automation; CDS; Program Management
NonPrimaryEventName	NonPrimaryEvent.xsd	NonPrimaryEventName	List of NonPrimary Event Types	Key, GPI, Tone, Audio or Video Effects
ProductCode	NonProgramDetail.xsd	ProductCode	List of Product Codes that categorize the product	
ProtectionSourceName	EventData.xsd	ProtectionSourceName	Name of the type of protection source to be used	
RouterSourceName	Location.xsd	RouterSource\Name	List of Router Source Names	ROUTER1; ROUTER2
SpotSalesClassification	NonProgramEvent.xsd	SpotSalesClassification	List of Rate section or priority code for the spot	
SpotType	NonProgramDetail.xsd	SpotType	List of NonProgram Types	BarterSpot; LocalAvail; Promo; PSA; ID; comments
TransitionPattern	Video.xsd	TransitionPattern	Available patterns for transitions from one video source to another	

8.1 Configuration – Procedure

When one system needs to provide for another a list of valid configuration items, it is possible to query that system for enumerations for those items that are not specifically enumerated in the schema. This can be done through the “messageType=Request” where the BxfQuery specifies the specific element and the ReturnStructure is set to “Configuration”. The resulting message should provide all of the valid enumerations that the system recognizes. This can be done in both directions depending on the flexibility of each system to recognize and set appropriate values.

8.2 Configuration – Non-schema Settings

Each system is responsible for handling the conversion of time received in a BXF message to its own appropriate start of broadcast day.

Start of Broadcast Day = “BxfTimeType Value”

Interruptions in communication during the receipt of any BXF message shall be considered in error if this configurable timeout is exceeded.

Message Timeout = integer value set to 20 seconds by default

9 Bibliography (Informative)

AD-ID LLC, Ad-ID, Advertising Digital Identification, LLC, Charlotte, NC

ATSC A/65C Program and System Information Protocol for Terrestrial Broadcast and Cable, Revision C with Amendment No. 1, Advanced Television Systems Committee, 9 May 2006

ATSC A/76 Programming Metadata Communication Protocol Standard, Advanced Television Systems Committee, Washington, DC, 10 November 2004

ATSC Code Point Registry, Advanced Television Systems, Committee, Washington, DC

Berners-Lee, T et al., IETF RFC3986 – Uniform Resource Identifiers (URI): General Syntax, The Internet Society, The Internet Engineering Task Force, 2005

CEA-708-C, Digital Television (DTV) Closed Captioning, Consumer Electronics Association

ISO 15706-2:2007, Information and documentation – International Standard Audiovisual Number (ISAN) – Part 2: Version identifier

ISO/IEC 13818-2:2000, MPEG-2 (for conditional access descriptor) Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video

Leach, P. and M. Mealling and E. Salz, A Universally Unique Identifier (UUID), 2005 IETF RCF4122

W3, XML Schema, W3C Recommendation, 2 May 2001

W3, XML Schema – Part 1: Structures Second Edition, W3C Recommendation, 28 October 2004

W3, XML Schema – Part 2: Datatypes Second Edition, W3C Recommendation, 28 October 2004

10 Notes (Informative)

This section describes issues concerning the implementation of the schema.

The Programming Metadata Communication Protocol specification A/76A is significant as a predecessor document. The BXF standard adopts a similar philosophy and approach to communication of data, and for practical purposes, extends the scope of PMCP, which was simply schedules and program attributes, to encompass the general case of data exchange between systems participating in television broadcast operations. Additional references from the field of television broadcast are listed.

10.1 Design Considerations

All text entered as strings and all XML messages themselves conform to UTF-8 as the standard for encoding all characters.

All date and time values should be in UTC, with no timezone offset.

10.2 Schema

The BXF schema is included as Annex C of this document.

Note: The BXF schema is included as Annex C of this document in a format that was exported from Altova's XMLspy, Professional Edition version 2008 The schema and all of its elements, annotations, restrictions, and enumerations shall be considered the normative representation of the standard and must be used to validate any messages that are generated by systems that use this standard as a means of communicating or transferring data. Where there are inconsistencies between this document and the schema, the schema shall take precedence. Certain attributes and elements of the schema refer to other schemas. These are not included in Annex C and must be referenced from their appropriate sources.

Table 2 – BXF Message Structure

Name	Type	Description
Message ID	GUID	Globally unique identifier
Message Date	Xs:dateTime	Identifies when the message was sent by the originating system
Message Type	BXF message type	See documentation for detailed listing of types
Originating System	String	A string name uniquely identifying the system
Destination System	String	A string name uniquely identifying the system
Authorized User	String	User authorizing the message, including cron job or process – include automated user name – i.e. logupdate@traffic1
Process or Application Name	String	A string name uniquely identifying the process or application.

The diagram that follows shows the structure and primary fields of the BXF message. The table above describes the primary message fields used to convey origin, destination and security/ authorization-related values.

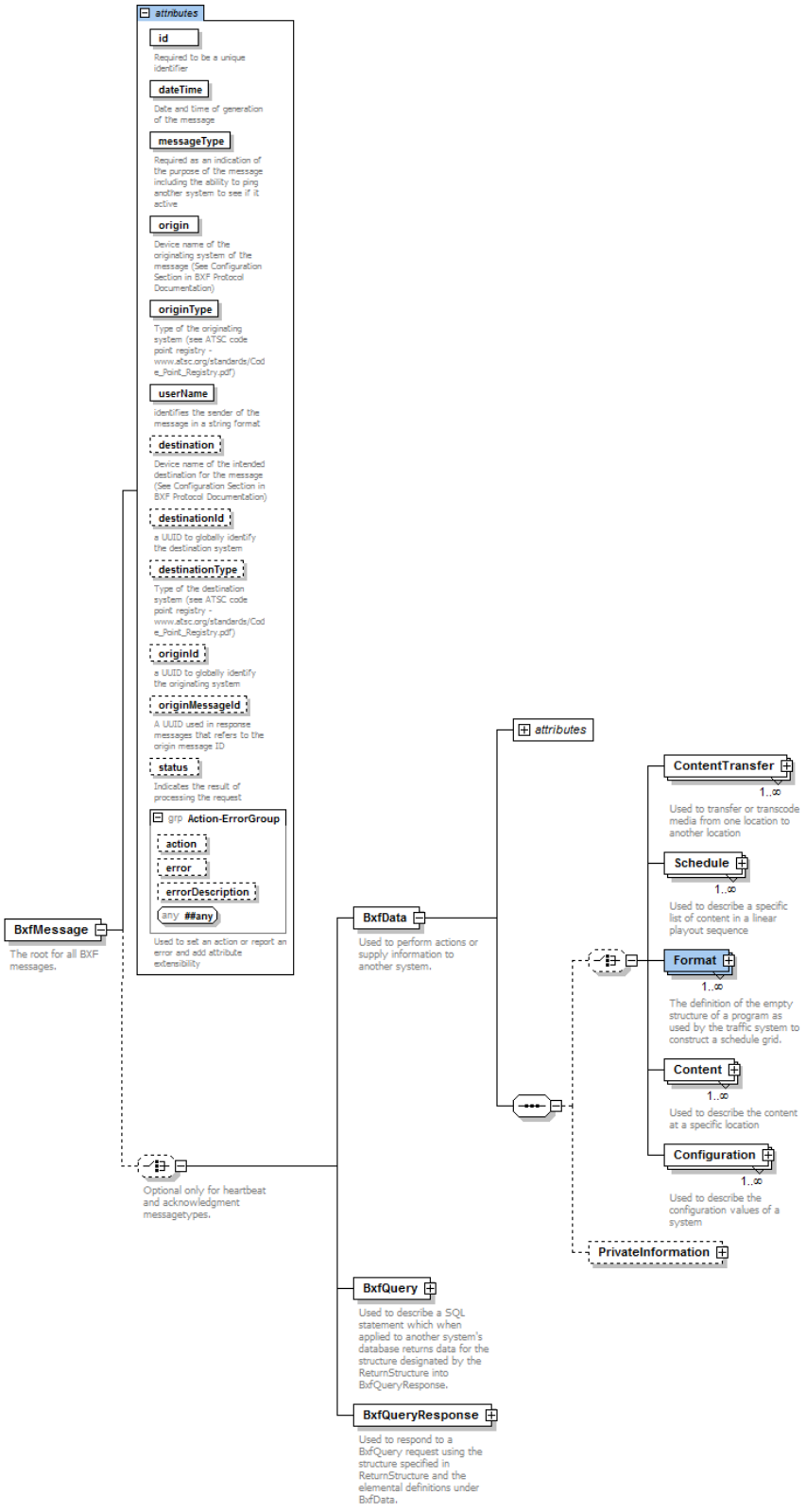


Figure 4 – Message Attributes from Schema

10.3 Overview of Transactions

The BXF schema supports request/response transactions and asynchronous notifications. The transaction type is communicated in the message itself, by means of the message type field attribute of BxfMessage - the root structure in the XML message.

The following view shows the top-level elements of BXF.

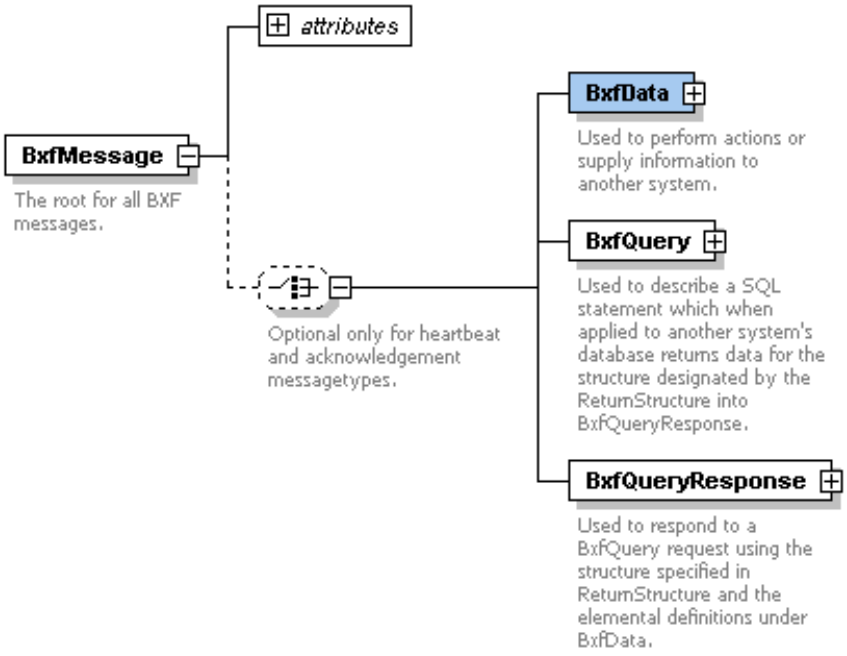


Figure 5 – Top Level BXF Elements

BXF messages can be considered to be in one of the four categories listed below, between party **A** and party **B**. In these examples, **A** originates the transactions, but in an actual system, either party can initiate the transaction. BXF messages are point to point (not broadcast), however, BXF is not limited to two participants: messages can be sent to multiple parties using individual point to point messaging (e.g. {**A** to **B**} {**A** to **C**} {**B** to **D**}).

A requesting **B** to take some action. **A** provides the necessary data; **B** responds with a status and returned data as appropriate to the request. **A** requesting **B** for a set of entities – similar to query (II), but generally more strictly defined.

I. **A** queries **B** for information. **A** provides parameters for the query; **B** provides a response

II. **A** notifies **B** of an addition/ change/ removal of entity, or entity status/ value change.

III. **A** makes a utility request to **B**, such as status of a previous request. A Heartbeat message sent **A** -> **B**, and **B** -> **A** is another example of a utility request.

The following table shows various example transactions, grouped according to the four transaction categories above. The table also outlines the use of the messageType and other fields in the BXF message. For reference, the BxfMessage attribute **messageType** values are: Request, Reply, Acknowledgement, Information, Heartbeat, and Message Status Request.

The following table illustrates the relationship between category of message, purpose of message (via use case examples), applicable messageType values, and the expected reply (message lifecycle). The details of the message lifecycle is explored in more detail later in this section.

Table 3 – Example Transactions by Category

Category	Use Case	messageType	Notes
I	Schedule Download *Sending of a day schedule to Automation from Traffic.* Invoke Schedule *Request to Automation to make a schedule active* Dub Order *Request for material to be dubbed from specified source to specified destination.*	Request (sender) Acknowledgement (receiver) Reply (receiver) Acknowledgement*	
II	Request Inventory *Request list of current clips available on video servers, with metadata for each.* Request Schedule *Request the schedule data for an entire day, or for specific time period.*	Request (sender) Acknowledgement (receiver) Reply (receiver) Acknowledgement* (sender)	BxfQuery node is populated in the request. BxfData.action = queryresult
III	Content Notify *Notification of newly-arrived content (to a video server, for example)* Content Transfer *Notification of newly-completed transfer of material.*	Information (sender) Acknowledgement (receiver)	
IV	Heartbeat	Heartbeat (sender) Acknowledgement (receiver)	
IV	Message Status Request	Request (sender) Acknowledgment (receiver) Reply (receiver) Acknowledgement*	BxfMessage.error indicates status of request.

* An acknowledgement of the reply is optional

10.4 Message Lifecycles

Each BXF transaction has an “initiating” message that starts it. The following message types are transaction initiators:

- Request
- Information
- Heartbeat
- Message Status Request

The Reply and Acknowledgement message types participate in the transaction and provide useful information regarding its current state. The following sections provide a conceptual overview of a message’s lifecycle, based on its initiating message type.

10.4.1 BXF Request Message Lifecycle

A request message represents a request to do something that warrants a reply. This may involve returning desired information or completing a task. If an error is detected by the destination, then an acknowledgement or reply shall be sent to the originator depending upon where the error was detected -- This is the last message that is sent by the destination. In other words, no reply is generated by the destination if the acknowledgement contains an “invalid” or “error” status.

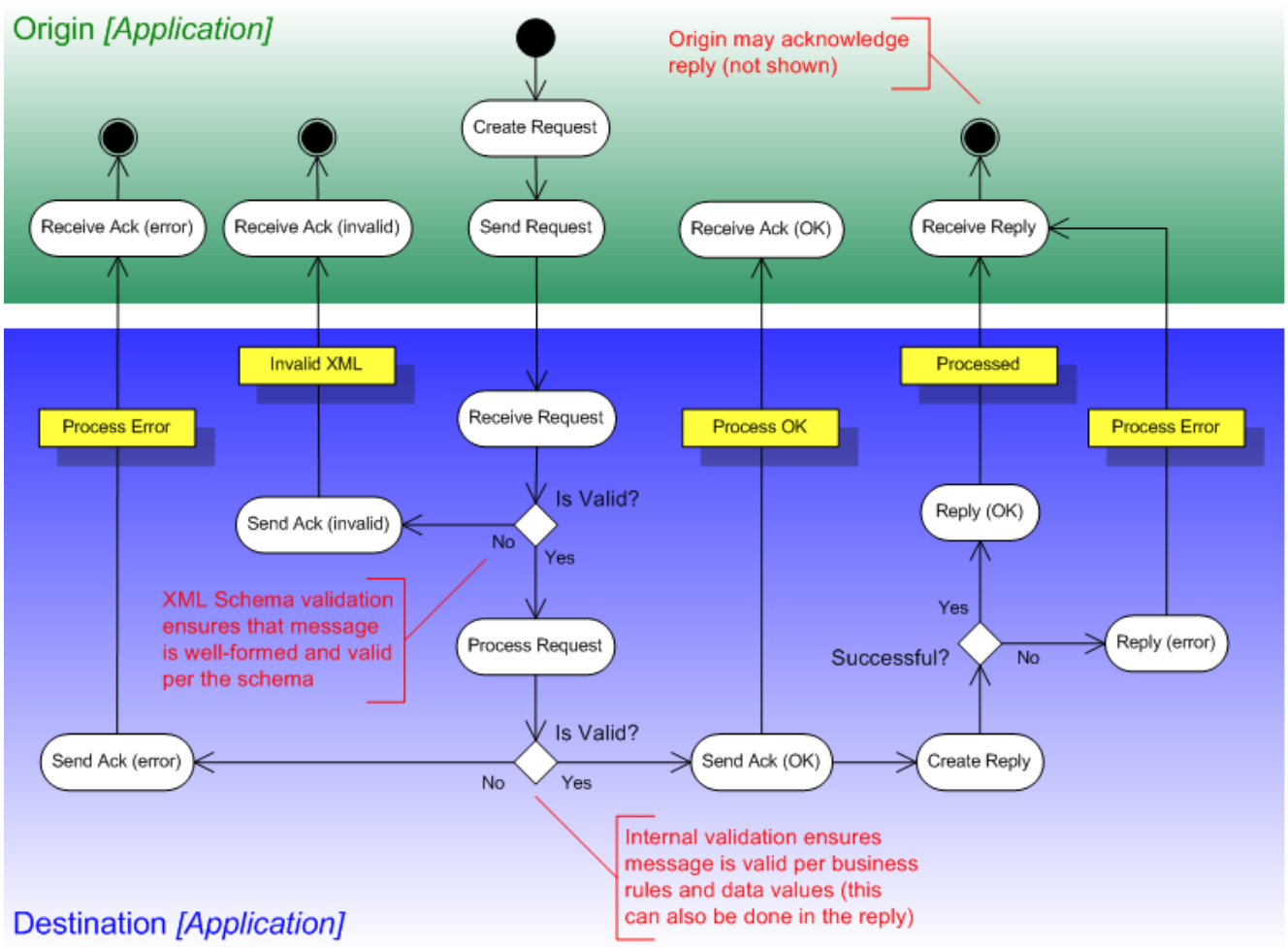


Figure 6 – "Request" Message Lifecycle

10.4.2 Information Message

An Information message represents a notification or request to do something that does not warrant a reply. This may involve modifying information (e.g., content) or completing a task. If an error is detected by the destination, then an acknowledgement shall be sent to the originator -- This is the last message that is sent by the destination. In other words, the destination does not generate additional messages (or attempt to process the message) if the initial acknowledgement is rejected with a status of “invalid”.

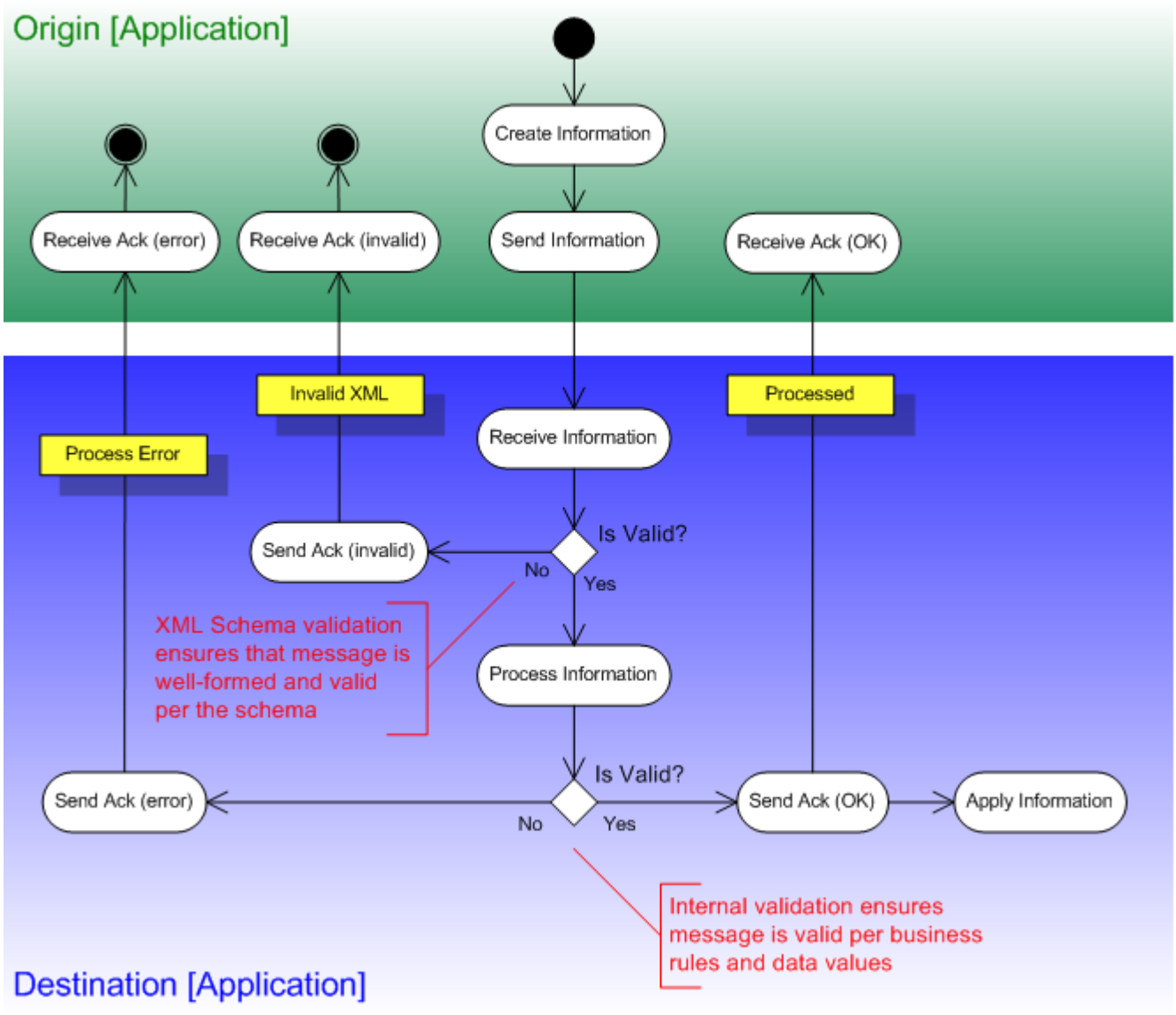


Figure 7 – "Information" Message Lifecycle

10.4.3 Heartbeat Message

A Heartbeat message represents a connectivity check. Details regarding heartbeat processing are discussed later in this document.

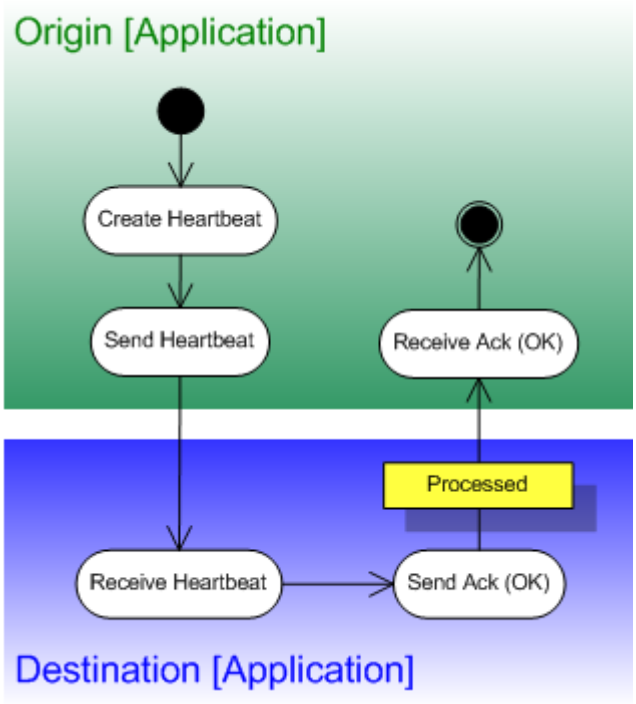


Figure 8 – "Heartbeat" Message Lifecycle

10.4.4 Message Status Request

A Message Status Request message represents a request for the status of a message that was previously sent. Details regarding message status processing are discussed later in this document.

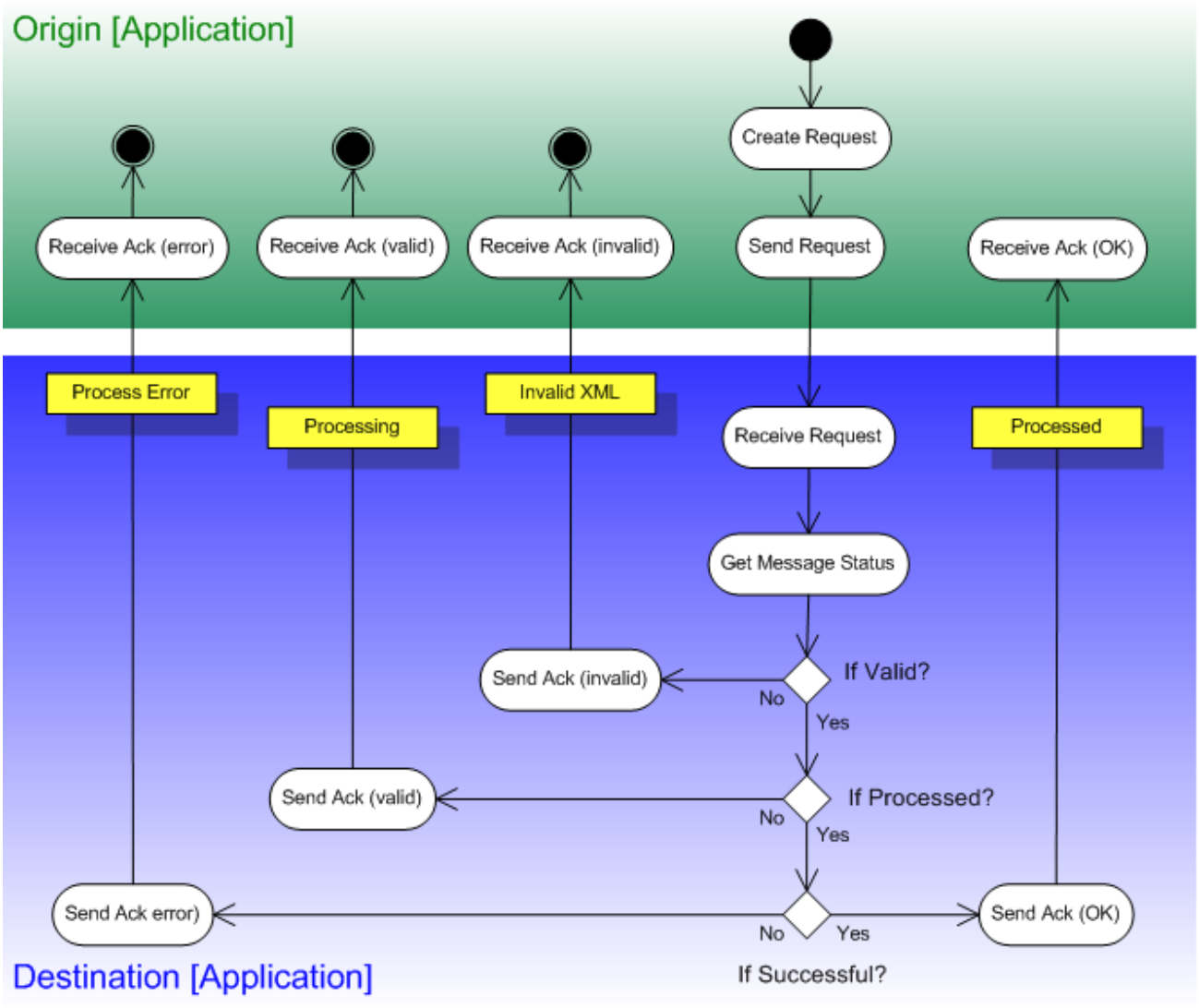


Figure 9 – "Message Status" Message Lifecycle

10.5 Message Processing

The following subsections describe the role of certain common messages exchanged in BXF transactions

10.5.1 Acknowledgement Messages

As indicated in the message lifecycle diagrams, the receiving device shall generate at least one acknowledgement for each message that is received. Multiple acknowledgements may be generated when a message is received, but not processed until a later date. For example, if the receiving device intends the queue the message for future processing, then it should send an acknowledgement indicating that the message is valid per the schemas (i.e., status “valid”). At a later date, it should send another acknowledgement indicating the message is valid for processing (i.e., status “OK”).

The sending device has the option of requesting the status of the receiving device, via a heartbeat message, or the status of a particular message, via a message status request. The receiving device shall generate an

acknowledgment to satisfy either of these requests. The acknowledgement must contain the status information that the sending device desires.

10.5.2 Heartbeat Messages

In connection-based implementations, client devices should monitor the server devices that they are connected to by using heartbeat messages. Client devices should support heartbeat message creation, as well as consumption of heartbeat acknowledgement messages. Server devices should support heartbeat message consumption, as well as create heartbeat acknowledgements. Additionally, each server device should monitor its clients by tracking their heartbeat message requests. Refer to the use case section for heartbeat message examples.

10.5.2.1 Heartbeat Message Timing and Timeout

For client devices, the interval between heartbeat request messages shall be configurable per connection. The acceptable range for the heartbeat interval is as follows:

1 second <= acceptable range <= 10 minutes

Heartbeats are not required if other messages are being sent between devices. Heartbeat message are intended to detect connectivity issues during idle or slow periods, when no messages are being sent. The timeout period for client devices should be configurable per connection. The client shall ensure that the heartbeat request interval is always greater than this timeout.

The server heartbeat timeout is defined as the maximum duration that the server waits between heartbeat request messages before deciding that a client device is no longer connected. It should be configurable per connection. Acceptable values are at the discretion of the implementers. It is recommended that the server should not simply take a single lost heartbeat request to indicate a failed device or connection, but instead wait for subsequent messages to verify if they are also lost. The number of heartbeat periods to wait before considering the connection lost should be configurable per connection. The server time out period and number of periods to wait should be adjusted during system set up based on the heartbeat message period set for the client in order to prevent false communication failure alarms and detect communication failures in a timely fashion.

10.6 Primary Message Attributes

Certain key attributes convey the meaning of the message and provide a means to communicate results. These attributes are:

BxfMessage.messageType

BxfMessage.status

Action-Error Group (consists of action, error, errorDescription)

The *messageType* and *status* attributes are only at the top-level node (BxfMessage). The Action-Error Group is the set of attributes as shown above. This set is available at the top-level BxfMessage node, and at other levels of the BXF Message. The role and function of each attribute is described below.

10.6.1 messageType Attribute

The “messageType” attribute provides crucial contextual information regarding the expected message contents. The existence of the action, status, error, errorDescription and originMessageId attributes is largely based off of the message type. Message types fall into two general categories: Initiating types and responding types.

10.6.1.1 Initiating

Initiating message types include “request”, “information”, “heartbeat” and “message status request”. The creator, or initiator, of a message shall specify one of these values for the message type. Initiating message types may

include action attributes, but they should not include status, error, errorDescription and originMessageId attributes, which are associated with response message types.

10.6.1.2 Responding

Response message types include “reply” and “acknowledgement”. The consumer of a message shall specify one of these values in the message it generates in response to a received message. Response message types must include values for the status and originMessageId attributes. If the message is rejected, the response should also contain error and errorDescription attributes.

10.6.1.3 Relationship to Message Body

The following table summarizes the relationship between the different BXF messages body types (i.e., sub-elements) and the message types that indicate how the body should be processed:

Table 4 – Message Body and Type Relationships

Message Body Type	Message Type
BxfData	Information
BxfQuery	Request, MessageStatusRequest
BxfQueryResponse	Reply
(none)	Acknowledgement, Heartbeat

Successful acknowledgement and heartbeat messages should not contain a message body type, since these messages consist of a BxfMessage element with no sub-elements. Error handling creates an exception to the above rules, however, since the original message body must be included in the response with errors indicating why the original message was rejected. For example:

An acknowledgment error message must include the original BxfData or BxfQuery element with the appropriate error attributes set, indicating why the message is being rejected.

A reply error message must include the original BxfQuery element with the appropriate error attributes set, indicating why the message is being rejected.

10.6.2 *status* Attribute

The status attribute is utilized on reply and acknowledgement messages to indicate the status of a received message: request, information, heartbeat or message status request message. The valid status values are described below.

10.6.2.1 valid

Acknowledgement messages with a status “valid” shall be generated when a message has been received and is valid per the BXF schema, but it has not been processed yet.

10.6.2.2 invalid

Acknowledgement messages with a status “invalid” shall be generated when a message has been received but does not comply with the schema described in this document and cannot be processed.

10.6.2.3 OK

Acknowledgement and reply messages with a status of “OK” shall be generated when a message was successfully processed by the receiving device.

10.6.2.4 error

Acknowledgement and reply messages with a status of “error” shall be generated when a message was rejected by the receiving device during processing. The message should include detailed information about the error, as described in the error handling section.

10.6.3 *action* Attribute

The action attribute is inherited from PMCP3.0 and describes what ‘action’ is requested of the receiver of the message. The action attribute is made available at many nodes in the hierarchy in order to support precise meaning in messages that may be large and complex. The ability to create precise messages is a language feature designed to enable participating systems to communicate effectively with minimum ambiguity in the messages.

The *action* values are **add**, **update**, **remove**, **query result**, and **information**. Section 4: *Actions in Messages* provides more detail and examples.

10.6.4 *error* Attribute

The error attribute implementation within the BXF protocol is an extension of PMCP3.0 and performs the same functionality with several new options as detailed below in table 5. It allows one or more errors to be identified within a single message. Each error attribute provides both the location of the error and the type(s) of errors that occurred. Note that the offending element or attribute name is added to certain error types to clearly identify fault.

Table 5 – Error Types

Error Types in BXF Schema			
Type	**Description**	**Examples**	**Origin**
**_change_denied*	Use when an element or attribute value cannot be changed	HouseNumber_change_denied	PMCP
**_missing*	Use when an element or attribute value is missing	ContentId_missing	PMCP
**_out_of_range*	Use when an element or attribute value is out of range	SmpteTimeCode_out_of_range	PMCP
**_does_not_exi*st	Use when an element or attribute value does not exist	Destination_does_not_exist	PMCP
duplicate_message	Use when a duplicate message is received		BXF*
processing_exception	Use when the destination system throws an unexpected exception during message processing; or when no other applicable error type exists -- this is the overarching error type		BXF*
Not_supported	Use when a message’s contents are not	An AsRun message is sent to a digital asset management system	BXF*

Error Types in BXF Schema			
Type	**Description**	**Examples**	**Origin**
	supported by the destination system		
system_unavailable	Use when the destination system is reachable, but internally not available (the message is considered to be valid)		BXF*
*Note that on some inherited elements from PMCP the error attribute may not support the BXF options. All of the items can be extended using the format "XmlTime_missing:system_ref_aaaaa". The use of the colon and the following text is allowed on any of the items.			

Every complex type within the BXF protocol has an error attribute. Refer to the Error Handling section for additional information on errors.

10.6.5 errorDescription Attribute

The errorDescription attribute implementation within the BXF protocol allows an error message to be passed that provides a detailed description of the error, or errors, that were identified by the receiving device. This information is provided for information purposes and can be used to expedite resolution to this error. Note that the errorDescription value is intended for the support staff, not the end user -- This value is not localizable so each error description should be specified in English.

Error descriptions should be provided, where appropriate, to clarify error types. However, it is acceptable to have error types without error descriptions (and vice-versa).

Every complex type within the BXF protocol has an errorDescription attribute. Refer to the Error Handling section for additional information on errors.

10.7 Actions in Messages

The action attribute implementation within the BXF protocol is very flexible, allowing actions to be defined at multiple levels within a message. It is the responsibility of the message creator to ensure that the appropriate action value is specified at the appropriate levels within a message. The valid values for action are "add", "update", "information" and "remove".

Every "actionable" element shall have an action attribute. If and element does not have an action, then it inherits the action of its parent.

The following rules apply for processing actions:

A parent element action provides the default action for the child elements.

The "information" action indicates that the value is being provided for contextual purposes. For example, a required element in the schema that represents a key value. The value has not changed, but it must be included to process the message.

The default action on request, information, heartbeat or message status request messages is "information". If an element does not contain an action, nor do any of its parents, then the implied action is "information".

A query request (i.e., BxfQuery) shall not contain any action attributes with a value other than "information".

A reply message shall not contain any action attributes.

An acknowledgement message shall not contain any action attributes.

Messages shall be rejected whenever conflicting actions are encountered (e.g., parent element action is "add", child element action is "delete")

An information or request data message (i.e., BxfData) that does not contain an action at any level, or only contains “information” actions, is ambiguous and shall be rejected.

If an element has an action attribute with the value “add”, the whole element, including its children, should be added by the receiving device. If a child of such an element has an action attribute, its value shall also be “add”. If the consuming application already had an element with the same identifier, then it and its children should be replaced.

If there is an action attribute with the value “update”, then the values should be updated within the receiving device. Each child element may have its own independent action attribute. For example, a child element may have an action of “add”, “remove” or “update” whenever the parent element action is “update”.

If an element has an action attribute with the value “remove”, then the referenced element should be removed, marked as removed, or disabled within the receiving device. Only the attributes required for unique identification should be interpreted by the receiver. All children elements and all other attributes should be ignored.

10.7.1 Action Examples (Valid)

The following sections contain examples of the use of the action attribute. This should be considered a representation of the common scenarios, but not an all-inclusive list.

10.7.1.1 Add All Elements

In the following example, all of the elements should be added:

```
 <SomeElement1 action="add">
     <SomeElement2>newValue</ SomeElement2>
     <SomeElement3>
          <SomeElement4>Newvalue</SomeElement4>
     </SomeElement3>
     <SomeElement5>
          <SomeElement6>Newvalue</SomeElement6>
     </SomeElement5>
</SomeElement1>
```

10.7.1.2 Update All Elements

In the following example, all of the elements should be updated:

```
    <SomeElement1 action="update">
        <SomeElement2> updatedValue</ SomeElement2>
        <SomeElement3>
             <SomeElement4>updateValue</SomeElement4>
        </SomeElement3>
        <SomeElement5>
             <SomeElement6>updatedValue</SomeElement6>
        </SomeElement5>
    </SomeElement1>
```

10.7.1.3 Remove All Elements

In the following example, all of the elements should be removed:

```
    <SomeElement1 action="remove"/>
```

10.7.1.4 Add Sub-Element to an Existing Element

In the following example, SomeElement7 is added to SomeElement3 (which is updated):

```
    <SomeElement1>
        <SomeElement3 action="update">
```

```
            <SomeElement7 action="add">newValue</SomeElement7>
        </SomeElement3>
    </SomeElement1>
```

10.7.1.5 Add and Update Sub-Elements Concurrently

In the following example, SomeElement2 contains an update (i.e., action is derived from parent) and SomeElement3, SomeElement5 and SomeElement6 are being added:

```
    <SomeElement1 action="update">
        <SomeElement2>updatedValue</ SomeElement2>
        <SomeElement3 action="add">
            <SomeElement4>newValue</SomeElement4>
        </SomeElement3>
        <SomeElement5 action="add">
            <SomeElement6>newValue</SomeElement6>
        </SomeElement5>
    </SomeElement1>
```

10.7.1.6 Add and Remove Sub-Elements Concurrently

In the following example, SomeElement3 is being added and SomeElement5 is being removed:

```
    <SomeElement1 action="update">
        <SomeElement3 action="add">
            <SomeElement4>newValue</SomeElement4>
        </SomeElement3>
        <SomeElement5 action="remove"/>
    </SomeElement1>
```

10.7.1.7 Update and Remove Sub-Elements Concurrently

In the following example, SomeElement3 is being updated and SomeElement5 is being removed:

```
    <SomeElement1 action="update">
        <SomeElement3>
            <SomeElement4>updatedValue</SomeElement4>
        </SomeElement3>
        <SomeElement5 action="remove"/>
    </SomeElement1>
```

10.7.1.8 Updating Nested Child Element

In the following example, SomeElement4 is being updated. No action is required on SomeElement1, since it is not changing:

```
    <SomeElement1>
        <SomeElement3 action="update">
            <SomeElement4>updatedValue</SomeElement4>
        </SomeElement3>
    </SomeElement1>
```

10.7.1.9 Action Examples (Not Valid)

The following examples of *invalid* action attribute usage are provided as an aid to understanding the functioning of the action attribute in a BXF message.

10.7.1.10 Delete Parent Element and Add Child

In the following example, SomeElement1 is deleted and child SomeElement3 is added. This message should be rejected by the consumer.

```
<SomeElement1>
    <SomeElement3 action="delete">
        <SomeElement7 action="add">newValue</SomeElement7>
    </SomeElement3>
</SomeElement1>
```

10.7.1.11 Add Parent Element and Remove Child

In the following example, SomeElement1 is added and child SomeElement3 is removed. This message should be rejected by the consumer.

```
<SomeElement1>
    <SomeElement3 action="add">
        <SomeElement7 action="remove">removeValue</SomeElement7>
    </SomeElement3>
</SomeElement1>
```

10.7.1.12 Add Parent Element and Update Child

In the following example, SomeElement1 is added and child SomeElement3 is updated. This message should be rejected by the consumer.

```
<SomeElement1>
    <SomeElement3 action="add">
        <SomeElement7 action="update">updateValue</SomeElement7>
    </SomeElement3>
</SomeElement1>
```

10.8 Error Handling

The previous sections described a variety of BXF attributes, including those that are involved in identifying errors. This section summarizes how errors should be represented using these attributes and provides error handling examples.

10.8.1 Error Handling Responsibilities

Error handling applies to acknowledgement and reply messages that are generated by a receiving device, after receiving message and determining that it is not valid. The acknowledgement or reply must contain the following information to clearly indicate that the message is not considered valid:

- Set status attribute on BxfMessage to “error” or “invalid”

- Set one or more error types and/or error descriptions within the message body

10.8.2 Error Handling Examples

The following sections contain representative sample of common error scenarios, but not an all inclusive list.

10.8.2.1 Acknowledgement Error Example

The following acknowledgement message contains two errors were identified in the received message: The userName attribute is missing from the BxfMessage element and no ContentId element was specified under ContentInformation. An error description was provided on the second error to aid in understanding.

```
<?xml version="1.0" encoding="UTF-8"?>
<BxfMessage id="urn:uuid:34567890-2345-2345-1234-345678901234" userName="Joe" messageType="Acknowledgement"
dateTime="2006-08-16T20:44:43.16" origin="Automation System" originType="Automation" destination="Traffic" status="error"
error="userName_missing" xmlns="http://smpte-ra.org/schemas/2021/2012/BXF" originMessageId="urn:uuid:12345678-1234-1234-
1234-123456789012" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://smpte-
ra.org/schemas/2021/2012/BXF BxfSchema.xsd">
  <BxfData action="add">
    <ContentTransfer transferId="urn:uuid:ABCCDDDD-1111-22E3-9AFF-0038338391E1" transferType="Purge" priority="Normal">
      <Content user="ProTrack User">
        <ProgramContent>
          <ContentMetaData error="ContentId_missing" errorDescription="The element 'ContentMetaData' in namespace
'http://smpte-ra.org/schemas/2021/2012/BXF' has invalid child element 'Name' in namespace 'http://smpte-
ra.org/schemas/2021/2012/BXF'. List of possible elements expected: 'ContentId' in namespace 'http://smpte-
ra.org/schemas/2021/2012/BXF'">
                                    <ContentId>
                                        <Isan root="0000-0000-0000"></Isan>
                                    </ContentId>
            <Name>A really, really, really long name</Name>
          </ContentMetaData>
        </ProgramContent>
      </Content>
    </ContentTransfer>
  </BxfData>
</BxfMessage>
```

10.8.2.2 Reply Error Example

The following reply message indicates that the receiving device experienced an internal error when processing the received message. The received message may, in fact, be valid, but the receiver cannot process it at this time.

```
<?xml version="1.0" encoding="UTF-8"?>
<BxfMessage id="urn:uuid:24567890-2345-2345-1234-345678901234" messageType="Reply" dateTime="2006-08-16T20:44:43.16"
origin="Automation System" originType="Automation" destination="Traffic" userName="jdoe" status="error"
error="processing_exception" errorDescription="Automation error occurred when processing message. Error occurred in
Automation.MessageHandler.ProcessMessage(). Error: NullPointerException at line 1173 of AutomationHandler.cs. Contact Automation
support."  xmlns="http://smpte-ra.org/schemas/2021/2012/BXF" originMessageId="urn:uuid:12345678-1234-1234-1234-123456789012"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://smpte-ra.org/schemas/2021/2012/BXF
BxfSchema.xsd">
      <BxfQuery>
          <WhereClause>BxfMessage/BxfData/Content/ProgramType/ContentMetaData/ContentId/HouseNumber="AS01002"</Whe
          reClause>
        <ReturnStructure>Content/*</ReturnStructure>
      </BxfQuery>
</BxfMessage>
```

10.9 Query Syntax

BXF Query has adopted a subset of XPATH 2.0 (http://www.w3.org/TR/xpath20/) to implement the <WhereClause> and <ReturnStructure> nodes syntax.

The following is the grammar used to define the syntax for both nodes in BNF notation. This grammar is implemented as XML regular expressions (see schema for implementation)

10.9.1 Syntax:

BxfPathExpression :: = NodeName | BxfPathExpression/NodeName

BxfVariable ::= BxfPathExpression/TerminalNodeName | BxfPathExpression/@AttributeName

BxfReturnStructure ::= BxfPathExpression | BxfPathExpression/*

BxfWhereClause ::= BxfSimpleWhereClause | BxfSimpleWhereClause BooleanOperator BxfWhereClause

BxfSimpleWhereClause ::= BxfVariable ComparisonOperator QuotedString

ComparisonOperator ::= = | != | < | > | <= | >= | <>

BooleanOperator := AND | OR

QuotedString ::= " unquoted string "

TerminalNodeName ::= node with no descendents.

AttributeName :: = any attribute associated with the last node on the path

NodeName ::= any node defined in the BXF schema.

10.9.2 Symbol Definition and Semantics:

- The "/" symbol is used to define absolute path from parent node to descendent.
- The "@" is used to make the distinction between whether the value is specifying that of a node or an attribute.
- The "*" symbol is used to specify "all" from that point on.

10.9.3 Reference Examples

10.9.3.1 All active Channels for a given date range

```
<BxfMessage ….>
<BxfQuery>
   <WhereClause> Schedule/@scheduleStart>= "2006-08-16T05:00:00.00"    </WhereClause>
   <ReturnStructure>Schedule/Channel/*</ReturnStructure>
  </BxfQuery>
</BxfMessage>
```

The above query would return all channels for which a schedule >=2006-08-16T05:00:00:00.000 is defined. Returned record set would be something like:

```
<BxfMessage ….>
<BxfAction ….>
```

```
    <Schedules>
        <Schedule scheduleId=”1234” …..>
            <Channel …..>
            </Channel>
        </Schedule>
        <Schedule…>
            …
        </Schedule>
  </BxfAction>
</BxfMessage>
```

The “*” indicates all sub-nodes at that level and below, if the “*” is omitted only that node and its attributes would returned.

10.9.3.2 Schedule identification for a given date range

```
<BxfMessage>
<BxfQuery>
   <WhereClause>Schedule/@scheduleStart>= '2006-08-16T05:00:00.00'
   </WhereClause>
   <ReturnStructure>Schedule</ReturnStructure>
  </BxfQuery>
</BxfMessage>
```

Must return

```
<BxfMessage ….>
<BxfAction ….>
   <Schedules>
        <Schedule scheduleId=”1234” …../>
        <Schedule…/>
            …
  </BxfAction>
</BxfMessage>
```

10.9.3.3 Schedules for a specific Channel for a given date range

```
<BxfMessage>
<BxfQuery>
   <WhereClause>Schedule/Channel/Name=”WXYZ” and  Schedule/@scheduleStart>= '2006-08-16T05:00:00.00'
   </WhereClause>
   <ReturnStructure>Schedule/*</ReturnStructure>
  </BxfQuery>
</BxfMessage>
```

This query would return, Schedules including all events for Channel Named “WXYZ” with a scheduleStart>='2006-08-16T05:00:00.00'

```
<BxfMessage ….>
<BxfAction ….>
   <Schedules>
        <Schedule scheduleId=”1234” …..>
          <Channel …>
            <Name>WXYZ</Name>
```

```
                ….
            </Channel>
            <Scheduled…>
                …
            </Scheduled>
            …
        <Schedule…/>
                …
  </BxfAction>
</BxfMessage>
```

10.9.3.4 Only “as Run” schedules on Channel WXYZ

```
<BxfMessage>
<BxfQuery>
   <WhereClause>Schedule/Channel/Name=”WXYZ” and  Schedule/@scheduleStart>= '2006-08-16T05:00:00.00'
   </WhereClause>
   <ReturnStructure>Schedule/AsRun/*</ReturnStructure>
  </BxfQuery>
</BxfMessage>
```

10.9.3.5 Content Transfer Query

This query returns the status of all content transfers on the device specified as the message destination.

```
<BxfMessage …>
<BxfQuery>
   <WhereClause/>
   <ReturnStructure>ContentTransfer<ReturnStructure>
  </BxfQuery>
</BxfMessage>
```

Returns

```
<BxfMessage ….>
<BxfAction ….>
   <ContentTranfers transferId=.. transferType=… status=../>
   <ContentTranfers transferId=.. transferType=… status=../>
      …
  </BxfAction>
/BxfMessage>
```