



XML Playlist Specification



Pebble Beach Documentation

Document Number :	PBS-1-00120
Document Version:	2.0
Document Revision:	D
Revision Date:	15/04/09
Author(s):	Peter Brown

© 2009 Pebble Beach Systems
All rights reserved

Disclaimer

This document has been checked and is believed to be reliable. However Pebble Beach Systems assumes no responsibility for any errors or omissions. Pebble Beach Systems reserves the right to amend its products in the future, and this document, without notice or obligation.

REVISION HISTORY

Version	Changes
Draft 0.1	First Draft for internal review
Draft 0.2	Final draft
Issue 1.0	1 st release
Issue 1.1	Updated inline with Neptune 2.4.0.0
Issue 1.2	Updated inline with Neptune 3.2.x.x destpath definition changed added transition effects and rates added reference to cg documentation
Issue 1.3	Added Reconcile key. Both Reconcile key and Houseld can now be used by traffic systems to carry external data. Corrected onairdate dateformat to match implementation. Corrected event type=FIX for fixed events to match implementation. Removed <device> element
Issue 1.4	Fixed typo regarding onairdate. Modified the example to make it clearer. Changed title field to 32 chars. Added <category> field for event types.
Issue 1.5	Added subtitler information
Issue 1.6	Updated to match new secondary event layout. Renamed <logo> to <gpimacro>.
Issue 1.7	Neptune 3.6.0.3 supports up to 64 character mediaids. <subtitle> schema changed Various omissions corrected.
Issue 1.8	Added examples for secondary GPI events
Issue 1.9	Changes for Neptune 3.6.1.x. Use of <onairdate> deprecated.
Issue 1.9.1	Added Marker Event Type and example of SecXPoint.Combined Secondary example for different device types.
Issue 1.10	Added support for Secondary Templates
Issue 2.0	General document overhaul. Updated Primary Subtitles section , Custom Data sections (primary and secondary), added VPS section, Implemented xsd, validated example xml files. Removed rogue 'bookmarks, and also reformatted MS Words 'smart-quotes'.
Issue 2.0B	Added additional WSS Aspect Ratios for VPS section
Issue 2.0C	Updated Secondary Rate and Tran elements to fix inaccuracy in the specification.
Issue 2.0D	Added information about the ordering of the subtitles section to make it compatible with the .xsd validation.

CONTENTS

Revision History	2
Contents	3
1 Introduction	6
1.1 Supporting Documentation	6
1.2 Nomenclature	7
1.3 Playlist	8
1.3.1 History	8
1.3.2 List	8
1.3.3 Eventlist	8
1.3.4 SecondaryEventlist	9
1.3.5 SecondaryTemplateList	9
2 XML Elements	10
2.1 XML Headers	10
2.1.1 Character Encoding	10
2.2 <playlist> Element	10
2.3 <history> Element	11
2.3.1 <created> Element	11
2.3.2 <modified> Element	11
2.4 <list> Element	11
2.4.1 <txdate> Element	12
2.4.2 <channel> Element	13
2.4.3 <name> Element	13
2.5 <eventlist> Element	13
2.6 <event> Element	13
2.6.1 <event type="MEDIA">	14
2.6.2 <event type="COMMENT">	14
2.6.3 <event type="LIVE">	14
2.6.4 <event type="MARKER">	15
2.6.5 <action> Element	15
2.6.6 <eventnote> Element	15
2.6.7 <onairtime> Element	16

2.6.8	<onairdate> Element - deprecated	17
2.6.9	<mediaid> Element	17
2.6.10	<houseid> Element	17
2.6.11	<reconcilekey> Element	18
2.6.12	<title> Element	18
2.6.13	<category> Element	18
2.6.14	<duration> Element	19
2.6.15	<som> Element	19
2.6.16	<starttype> Element	19
2.6.17	<endtype> Element	20
2.6.18	<stream> Element	20
2.6.19	<source> Element	20
2.6.20	<audiosource> Element	21
2.6.21	<destpath> Element	21
2.6.22	<gpimacro> Element	21
2.6.23	<effect> Element	22
2.6.24	<rate> Element	22
2.6.25	<cg> Element	22
2.6.26	<subtitles> Element	23
2.6.27	<segment> Element	25
2.6.28	<timer> Element	25
2.6.29	<vps> Element	26
2.6.30	Primary Event <customdata> Element	27
2.7	<secondaryeventlist> Element	29
2.7.1	<secondaryevent type="...">	29
2.7.2	<secondaryevent type="CG">	32
2.7.3	<secondaryevent type="LOGO">	32
2.7.4	<secondaryevent type="KEY">	33
2.7.5	<secondaryevent type="MEDIA">	33
2.7.6	<secondaryevent type="GPI1 ON">	33
2.7.7	<secondaryevent type="GPI1 OFF">	34
2.7.8	<secondaryevent type="GPI2 Pulse">	34
2.7.9	<secondaryevent type="SecXPoint">	34
2.7.10	Combined Custom Data for different list devices.	35
2.8	<secondarytemplatelist> Element	36

2.8.1	<secondarytemplate> Element	36
2.8.2	<startoffset> Element	36
3	Element Summary	37
3.1	Primary Events	37
3.2	Primary Event Custom Data	39
3.3	Secondary Events	41
3.4	Secondary Templates	43
4	Example Playlists	44
4.1	Example 1 - Simple Playlist	44
4.2	Example2: Playlist with Secondary Events	46
4.3	Example 3: Playlist with Secondary Templates	49

1 INTRODUCTION

This document describes the playlist file format for the Pebble Beach Systems Neptune automation system. This format is intended to be compatible with all future versions of the Neptune product although this specification will change as new functionality is introduced into the Neptune system.

The file format uses the XML (eXtensible Markup Language) language which, like HTML, is a descendent of the SGML (Standard Generalised Markup Language). XML is an application of SGML that has been defined to be simpler and easier to use. XML is rapidly becoming the de facto standard for the interchange of information between systems.

The actual amount data required by the automation system to transmit a transmission schedule will depend on the customer's particular requirements and whether information about the media to be broadcast is already in Neptune's SQL database. As such, a playlist could simply provide the automation system with a list of media IDs to be played in sequence, or the playlist could contain all of the information required by the automation to cue and play the media.

The Neptune automation system relies on playlist files to control the playout of material from various devices. Playlist files can be created within the Neptune system or can be imported from other systems such as traffic and scheduling systems.

The exact contents of the playlist file may vary from one customer to another depending on the customer's configuration and requirements and also depending on the capabilities of the traffic and scheduling systems. As a consequence of this, certain fields within the playlist structure are defined as mandatory, others are optional.


Playlist files should be given the standard file extension ".xml"

1.1 SUPPORTING DOCUMENTATION

This specification is provided along with the following additional files:

\PlaylistSpecification.pdf	This document
\Schema\NeptunePlaylistvX.X.xsd	XSD Schema file to help validate playlists during development
\samples\example1_simplePlaylist.xml	Example used in section 4.1
\samples\example2_playlistWithSecondaryEvents.xml	Example used in section 4.1
\samples\example3_playlistWithSecondaryTemplates.xml	Example used in section 4.1
\samples\example4_largePlaylist.xml	Long playlist example
\samples\example5_vpsExample.xml	Playlist with VPS support example

1.2 NOMENCLATURE

Term	Meaning
Playlist	<p>For the purposes of this document, the word "playlist" refers to a file containing playlist information.</p> <p>Examples of XML syntax are shown in the <code>courier</code> font. For example:-</p> <pre data-bbox="479 604 1144 703"><playlist> <history> <created name="Joe Bloggs" datetime="2001-03-02T12:00:02"/></pre> <div data-bbox="495 751 620 871">  </div> <p data-bbox="673 751 1226 856">Note: XML is a case-sensitive description language. <PLAYLIST> is not the same as <playlist></p>
Document Object Model	<p>The playlist itself can be thought of as being composed of several distinct sections, each section being a branch or node of a tree structure, In summary:</p> <pre data-bbox="479 1150 998 1554"> Playlist ├── History ├── List └── Eventlist ├── Event ├── SecondaryEventList │ └── SecondaryEvent └── SecondaryTemplateList └── SecondaryTemplate </pre>

1.3 PLAYLIST

The playlist element indicates the start and end of the playlist information. This is the 'root node' of the XML document. An XML document may contain only one root node.

1.3.1 History

The History element contains a creation and modification history of the playlist file. When the file is created the created

element should be set and when any subsequent edits are made on the file, the modified elements are set.

1.3.2 List

The List field contains general information about the list, including a long name for the list, the broadcast channel that the list is to be broadcast on and the date o

f the eventual transmission of the schedule. This information allows the user to be sure that they have selected the correct file before loading the file into a transmission channel.

1.3.3 Eventlist

The Eventlist tag indicates the start of the actual playlist data. Each item in the playlist schedule is defined by an <event> element. The XML document must contain only one Eventlist section.

1.3.3.1 Event

Each item in the transmission schedule is defined by an <event> element. The order of the <event> elements in the XML file is the order in which the individual items will be broadcast. The information contained within the <event> element indicates all (or some - depending on system configuration) of the information required to broadcast the playlist.

Events can be of different type. Currently 3 kinds of event are defined; more may be added in future. These are:-

- Media event. Used for playing and recording all types of physical media
- Comment event. An event that only serves as a comment in the list for the user.
- Live event. An event where the transmission channels broadcasts a live video source.
- Marker event. An event with zero duration which may be used to carry Secondary Events in a Secondary Event List.

The events themselves contain child elements that contain data about the event. This information typically includes the following information:-

- the type of action the automation system will perform on the event
- the start type of the event that indicates how the automation should start the event.
- the on-air time and date (approximate or exact) when the event will start
- media ID of the item that will be played

- the House or Local ID of the item to be played
- the Media Title of the item that will be played
- the Duration of the event
- the Start Of Message (SOM) timecode value of the material that will be played
- the destination video path for the transmission channel.
- the transition that should be used to start the event
- whether a channel logo should be displayed
- any additional note information about the event
- the end-type of the event that indicates how the automation should end the event.
- optional custom data for the event
- A list of any secondary or child events.

1.3.4 *SecondaryEventList*

The <secondaryeventlist> tag indicates the start of the secondary event data. A secondary event is an action that starts and ends relative to the primary event. A secondary event may be, for example, a CG event, a GPI closure, an additional router switch or a DVE effect.

An <event> element must contain only one <secondaryeventlist> section.

1.3.4.1 *SecondaryEvent*

One or more secondary events may be defined in the <secondaryeventlist>. The secondary event elements themselves contain child elements that contain data about the secondary event. This information typically includes the following information:-

- when the secondary event should start, relative to the start or end of the primary event
- when the secondary event should end, relative to the start or end of the primary event
- data specific to secondary device being controlled.

1.3.5 *SecondaryTemplateList*

The <secondarytemplatelist> tag indicates the start of a list of templates, which contain secondary events to be applied to the primary event: the templates are defined in the registry and are referred to by name from this section. When the transmission list is imported, all references to secondary templates are replaced by their corresponding definitions from the registry and this is not reversible, i.e. if you then export the playlist, all references to templates will have been replaced by their component secondary events as defined in the registry.

2 XML ELEMENTS

2.1 XML HEADERS

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="http://Neptune/playlist.xsl" ?>
```

The header information provides the recipient with important information about the XML file and its structure. The current version of the XML language is 1.0. More information about XML can be found at <http://www.xml.org>.

The XSL stylesheet reference allows the XML playlist file to be readily displayed in a standard Web browser by the use of an XSLT stylesheet. Pebble Beach can supply customers with custom stylesheets depending on their formatting requirements. This feature allows customers to share and print playlist files without directly using an automation client.

[Note: whilst developing XML playlists it may be useful to comment out or remove the XML-stylesheet reference as certain Web browsers (e.g. Microsoft Internet Explorer 5.x+) allow raw XML files to be displayed in a primitive tree-like view]

2.1.1 Character Encoding

Neptune allows different character encoding schemes to be used within the playlist XML files. By default, UTF-8 is used. Where different language character sets must be supported, Neptune has usually configured to use ISO-8859 encoding. This is specified by the XML header:-

```
<?xml version="1.0" encoding="ISO-8859-1" ?> (Western European)
```

Other ISO-8859 encoding schemes may be used such as ISO-8859-2 (Eastern European) and ISO-8859-7 (Greek). Non-ISO code pages such as "windows-1256" (Arabic) are also supported.

Neptune 3.6.0.3 and above has full unicode support for <title>, <eventnote> and secondary event cg text.

2.2 <playlist> ELEMENT

```
<playlist>
```

The playlist element is the root element of the XML document. It signifies that everything else in the document describes a playlist.

The playlist element can also reference a xml schema file (.xsd) which can be useful to help validate the xml playlist. This document is provided along with a sample .xsd and some sample .xml files.

To add a reference to the .xsd you can add the following to this element:

```
<playlist xmlns="http://www.pebble.tv/playlist"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.pebble.tv/playlist
../NeptunePlaylistv2.0.xsd">
```

2.3 <history> ELEMENT

```
<playlist>
```

```
└─ <history>
```

The history element contains information about who initially created the playlist document and also the person who last edited the playlist document. It acts as the parent node for <created> and <modified> elements.

2.3.1 <created> Element

```
<playlist>
```

```
└─ <history>
```

```
    └─ <created name="Joe Bloggs" datetime="2001-03-02T12:03:00">
```

```
        </created>
```

This field indicates the person, system or username that first saved the playlist field. The created field must contain 2 attributes, namely:- 'name' and 'datetime'.

The date and time fields shall be specified in ISO8601 format (XML Schema dateTime format):

i.e. "YYYY-MM-DDThh:mm:ss" where 'T' is used as a separator between date and time

2.3.2 <modified> Element

```
<playlist>
```

```
└─ <history>
```

```
    └─ <modified name="Fred Smith" datetime="2001-03-02T12:03:00">
```

```
        </modified>
```

This field indicates the person, system or username that last modified the playlist field. The modified field must contain 2 attributes, namely:- 'name' and 'datetime'.

The date and time fields shall be specified in ISO8601 format (XML Schema dateTime format):

i.e.: "YYYY-MM-DDThh:mm:ss" where 'T' is used as a separator between date and time

2.4 <list> ELEMENT

```
<playlist>
```

```
└─ .....
    └─ <list>
```

The list element contains useful information about the playlist as a whole. Typically this includes the date on which the transmission schedule is expected to be broadcast and also the name of the transmission of channel that the playlist is intended for. It acts as the parent node for <channel> , <txdate> and <name> elements.

2.4.1 <txdate> Element

```
<playlist>
  |
  └─ <list>
      |
      └─ <txdate>2001-03-02</txdate>
```

The <txdate> tag indicates the expected date on which the schedule will be broadcast. The date is represented in ISO8601 format. (XML Schema date format). i.e. "YYYY-MM-DD"

2.4.2 <channel> Element

```

<playlist>
  |
  |— <list>
      |
      |— <channel>SportsNet</channel>
  
```

The <channel> tag indicates the channel on which the playlist is intended to be broadcast. Whilst this is primarily intended to be informational, the automation system can use this to provide a check when the list is loaded as to whether the list is permitted to be broadcast on a particular transmission channel. This may prevent, for example, an 'adult themed' playlist being loaded into the Children's Channel transmission list.

2.4.3 <name> Element

```

<playlist>
  |
  |— <list>
      |
      |— <name>SportsNet Master playlist </name>
  
```

The <name> element is used to provide the user with a meaningful name for the playlist. This can be useful for systems that are unable to use long filenames for the playlist file itself.

2.5 <eventlist> ELEMENT

```

<playlist>
  |
  |— <eventlist>
  
```

This element indicates the start of the real playlist event information. It acts as the parent node for <event> elements.

2.6 <event> ELEMENT

There are several types of media event that can be specified in the playlist:-

- Media (the default type)
- Comment
- Live
- Marker

The different types are qualified by an element attribute named "type". An event element may contain a <secondaryeventlist> element

2.6.1 <event type="MEDIA">

```

<playlist>
  |
  |-----<eventlist>
  |         |
  |         |-----<event type="MEDIA">
  
```

<event> elements form the bulk of the playlist document. One <event> element is required for each piece of media or content to be played on the channel. An event could therefore relate to a video tape, server clip or router source.

Note that <event> elements within the playlist document occur in their intended broadcast order. Each event element has an optional attribute called "type". The default type is "MEDIA" event.

Note: - In the case of media events the attribute *type="MEDIA"* can be omitted.

In other words: - <event type="MEDIA"> is the same as <event>

2.6.2 <event type="COMMENT">

```

<playlist>
  |
  |-----.....
  |
  |-----<eventlist>
  |         |
  |         |-----<event type="COMMENT">
  |                 |
  |                 |-----<eventnote>This is a long note in the list</eventnote>
  |                 |
  |                 |-----</event>
  
```

An event element of *type="COMMENT"* is used to signify within the list, that the event is a comment and not intended for broadcast. This means that child elements within the event are informational and are not intended for broadcast. The automation shall typically only display the contents of one child <note> element for the event.

2.6.3 <event type="LIVE">

```

<playlist>
  |
  |-----.....
  |
  |-----<eventlist>
  |         |
  |         |-----<event type="LIVE">
  |         |
  |         |-----
  
```

Live events are specified when the automation must do nothing but switch a live video and audio source on-air. During a live event none of the automation devices are actually playing. Live events are typically used for studio programs, sports events and satellite feeds.

2.6.4 <event type="MARKER">

```

<playlist>
├── .....
├── <eventlist>
│   └── <event type="MARKER">
│       |

```

Marker events are zero duration Events that may be used to carry Secondary Events in a Secondary Event List. Typically they are specified when the automation must do something at the end of a LIVE Event with an undefined end type. For example turn off a logo or graphic.

2.6.5 <action> Element

```

<playlist>
└── <eventlist>
    └── <event type="MEDIA">
        └── <action>PLAY</action>

```

This field tells the automation system in which context to process the event. At present the automation differentiates only between 'PLAY' and 'RECORD' actions.

2.6.6 <eventnote> Element

```

<playlist>
└── <eventlist>
    └── <event type="MEDIA">
        └── <eventnote>This is a note</eventnote>

```

```

<playlist>
└── <eventlist>
    └── <event type="COMMENT">
        └── <eventnote>This is a long comment</eventnote>

```

An eventnote element may be used to describe additional information to the user about a particular event.

An eventnote element may also be used to show a comment or to indicate or delimit a certain position in the list. These notes greatly help the readability of the playlist for the user.

Examples of when comments might be used might include:-

- before and after commercial breaks in order to separate commercials from programs.
- to indicate places in the list that may require user intervention close to on-air time to resolve timing problems or alternate program choices.

Eventnotes greater than 120 characters may be truncated in the automation client due to the available screen width in the user-interface.

2.6.7 <onairtime> Element

```
<playlist>
  |
  |   <eventlist>
  |   |
  |   |   <event type="MEDIA">
  |   |   |
  |   |   |   <onairtime>2008-02-04T12:00:00:00</onairtime>
```

This field is mandatory only for fixed start time events. ie. <starttype>FIX</starttype> However, it is usual practice for playlists to include the estimated starting time for all events so that before the list is loaded into the automation, it is possible to see the approximate start times for every event in the list. Once the automation begins playing a list, operator intervention and live program events frequently change the eventual start times of events in the list. For fixed start time events however, the <onairtime> value is taken to be the absolute time at which the automation should start the event.

Note:- in Neptune 3.6.1.x this element is also used to specify the date part of the on-air time. Previously this was specified in the <onairdate> which is now deprecated.

The time is formatted as a timecode string or ISO-8601-style dateTime string:-

Eg. "hh:mm:ss:ff".
 "yyyy-mm-ddTHH:MM:SS" (fully ISO-compliant)
 "yyyy-mm-ddTHH:MM:SS:FF"

Drop frame notation (hh:mm:ss;ff) is also supported.

2.6.8 <onairdate> Element - deprecated

```
<playlist>
  |
  | <eventlist>
  | |
  | | <event type="MEDIA">
  | | |
  | | | <onairdate>21 03 04</onairdate>
```

Note: Use of <onairdate> is deprecated in Neptune 3.6.1.x, 3.8.x and later versions.

This field is mandatory only for fixed start time events ie. <starttype>FIX</starttype>. It is used to ensure that the fixed start time event plays on the correct day. This is important when there are many fixed events throughout the day and the playlist is longer than 24 hours. The date is represented in "dd mm yy". Note this field was originally specified as being in an ISO-8601 date format - the actual implementation is as above.

2.6.9 <mediaid> Element

```
<playlist>
  |
  | <eventlist>
  | |
  | | <event type="MEDIA">
  | | |
  | | | <mediaid>ABCDEFG1DSADDS</mediaid>
```

This field indicates the identifier of the media that will be played. Depending on the customer configuration and hardware being controlled in the system, this field may be limited to 8 characters.

2.6.10 <houseid> Element

```
<playlist>
  |
  | <eventlist>
  | |
  | | <event type="MEDIA">
  | | |
  | | | <houseid>BA323218</houseid>
```

This field is included to support customers that require their own alternate id or house number for a piece of media. The automation simply carries this field though the system and does not need to be able to decode this data.

2.6.11 **<reconcilekey> Element**

```

<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <reconcilekey>45CB7F88</reconcilekey>

```

This field is included to support customers that require their own unique data or key to be associated with a playlist event. The automation simply carries this field through the system and does not need to be able to decode this data. Typically this field is used by customers to reconcile the as-run list with the original transmission schedule.

2.6.12 **<title> Element**

```

<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <title>the title</title>

```

It is usual for playlists to include a user-friendly name for the event that is to be broadcast.

2.6.13 **<category> Element**

```

<playlist>
  └─ <eventlist>
      └─ <event>
          └─ <category>PROMO</category>

```

The category element is used to enable a scheduling system to classify the kind of video material represented by the playlist event. The defined event types in the automation system are user configurable on a system and the user can configure different colours for each type. This enables COMMERCIALS or ADVERTS to be shown in a different colour from PROGRAMS in the playlist. This feature may be useful as the automation database may not have this type of classification available. If the imported playlist does not contain a <category> element, or if the category specified has not been configured in the automation system this field will be displayed as [NONE] in the playlist.

2.6.14 <duration> Element

```

<playlist>
  |
  | <eventlist>
  | |
  | | <event type="MEDIA">
  | | |
  | | | <duration>00:00:20:00</duration>

```

In almost all cases the duration of an event should always be included in the playlist. The automation will generally check this duration against the Neptune SQL database and/or the device that will play the media event and so if durations are not supplied as soon as the list is loaded, the user will be able to see the durations of the events. The time is formatted as a timecode string:- "hh:mm:ss:ff".

Note that in order for Neptune to play media that is not in the Neptune database this field is mandatory.

2.6.15 <som> Element

```

<playlist>
  |
  | <eventlist>
  | |
  | | <event type="MEDIA">
  | | |
  | | | <som>10:00:00:00</som>

```

Usually the Start Of Message (SOM) value for a media event is looked up in the automation's SQL database when the transmission list is loaded into the system. If the creator of the playlist knows the exact SOM and Duration of the media then these can be included in the list. The time is formatted as a timecode string:- "hh:mm:ss:ff".

Note that in order for Neptune to play media that is not in the Neptune database this field is mandatory.

2.6.16 <starttype> Element

```

<playlist>
  |
  | <eventlist>
  | |
  | | <event type="MEDIA">
  | | |
  | | | <starttype>SEQ</starttype>

```

This field controls how the automation will start the event. **SEQ**uential indicates that the event will follow-on sequentially from the preceding event. **FIX**ed means that the event will start at a supplied fixed start time (entered in the <airtime> element). **MAN**ual indicates that the event will be manually started by the operator.

2.6.17 **<endtype> Element**

```

<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <endtype>NORM</endtype>
  
```

This element determines how a LIVE event will behave once its duration has elapsed. If this element is not supplied, the automation shall assume that the type is '**NORM**'al. Normal means that the event shall end after its scheduled duration. '**UNDEF**'ined means that the event shall not end until explicitly ended by the user. This field currently only affects LIVE events ie. <event type="LIVE">.

2.6.18 **<stream> Element**

```

<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <stream>0</stream>
  
```

The <stream> element is used in special configurations where the playlist must explicitly specify the Neptune device and stream that shall play the event (Usually for A/B operation). The default stream in the list is '0'. The allowable values are 0..9.

2.6.19 **<source> Element**

```

<playlist>
  └─ <eventlist>
      └─ <event type="LIVE">
          └─ <source>SAT-1</source>
  
```

The <source> element is used to provide the source path for video and audio into the automation system. This typically is only used for live events where the playlist must specify the router source to be used on-air. The <source> used must match a source name in the Neptune router configuration.

2.6.20 <audiosource> Element

```

<playlist>
  └─ <eventlist>
      └─ <event type="LIVE">
          └─ <audiosource>CD PLAYER</audiosource >

```

The <audiosource> element is used to provide an alternate audio source path into the automation system. This typically is only used for live events where the playlist must specify the router source to be used on-air. The <audiosource> used must match a source name in the Neptune router configuration. If this field is not specified, it is assumed that the audio follows the video source.

2.6.21 <destpath> Element

```

<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <destpath>TX-A</destpath>

```

The <destpath> element is used to provide an alternate destination path for video and audio through the automation system. This only needs to be used if the playlist must select different destinations for video and audio for the transmission channel on an event by event basis. Usually this is not required and the destination is configured by the automation system. The <destpath> used must match a destination name in the Neptune router configuration.

2.6.22 <gpimacro> Element

```

<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <gpimacro>LOGO 1 ON</gpimacro>

```

The <gpimacro> element indicates that GPI macro is to be associated with the playlist event. This could be used, for example, to indicate that a logo should be triggered during the event or that a GPI should be pulsed. The names used must match the names used in the Neptune BOM Macro/GPI configuration table, which is customer specific.

2.6.23 <effect> Element

```
<playlist>
├
  <eventlist>
    └ <event type="MEDIA">
      └ <effect>FadeFade</effect>
```

The <effect> element is used to specify the 2.6.23 transition effect that is used at the START of the associated playlist event. The possible values include, but are not limited to:

- Take (the default if nothing specified)
- FadeFade
- TakeFade
- Mix
- FadeTake
- Wipe

2.6.24 <rate> Element

```
<playlist>
├
  <eventlist>
    └ <event type="MEDIA">
      └ <rate>Medium</rate>
```

The <rate> element is used to specify the transition rate that is used for the transition effect at the START of the associated playlist event. The possible values include, but are not limited to:

- Fast (default if nothing specified)
- Medium
- Slow

2.6.25 <cg> Element

No longer used.

2.6.26 <subtitles> Element

```

<playlist>
├── <eventlist>
│   ├── <event type="MEDIA">
│   │   └── <subtitles>
│   │       ├── <file>GB12345678</file>
│   │       ├── <cap>Open</cap>
│   │       ├── <lang>ENG</lang>
│   │       ├── <lang>SWE</lang>
│   │       └── <cust>font1</cust>
│   └── </subtitles>
└── </eventlist>
</playlist>

```

The <subtitles> element is used to indicate that an event should be played out together with subtitles. The contents of the node specifies the filename, the captioning mode, the language and any other custom settings the subtitle system might have. The subtitle filename is usually the same, or can be derived from, the <mediaid> of the event. The usage of the other subtitle fields is dependent on the system being controlled. Multiple <lang> elements are allowed in order to specify multiple languages.

Captioning Mode <cap>	Language <lang> (ISO639-3)
Undefined	Undefined
Open	ENG
Closed	SWE
Mixed	NOR
	DAN
	FIN
	ARA
	FRE
	GER



In order for the <subtitles> element to meet the .xsd XML validation, the ordering of the child elements is critical.

The XML specification states that if there are optional child elements and the possibility of multiple identical child elements (in this case <lang>) then the model used is Sequence. This therefore means that the ordering of the elements is mandatory.

As such, the ordering MUST be:

<file>	MANDATORY
<cap>	optional
<lang>	optional
<lang>	optional multiple lang elements
<cust>	optional

2.6.26.1 Subtitles Custom Data Field (<cust>)

The <cust> element is site/equipment specific and allows certain functions to be selected such as subtitle font, aspect ratio etc. These are device specific and are not needed for every subtitle device.

Please check with Pebble Beach Systems regarding this field in relation to your chosen Subtitle device.

The <cust> element is optional.

If required, the <cust> data string can be concatenated by the semicolon (;) character. Thus multiple custom fields can be defined for the same subtitle event. (Note that this means the semicolon is not permitted within any custom string data as it is a reserved character.)

The ordering of the parameters is non critical within the <cust> element.

Only ONE <cust> element should exist within the <subtitles> element.

Example of Custom Data Parameters:

ASPECT

This has valid options of:

- 16:9
- 4:3
- HD

MODE

This has valid options of:

- File
- File + Live
- Live
- Newsroom

AUXID

The Aux ID is an arbitrary string which can be used to identify the subtitle file, and this is device specific. One example may be that the subtitle device needs more than just a filename to reference the specific subtitles to be used, and this AuxID field allows such information to be passed from the traffic system through to the subtitle device.

Some Examples of the Custom Data string inside subtitles section:

```
<subtitles>
  <file>STFile001</file>
  <cap>Open</cap>
  <lang>ENG</lang>
  <cust>ASPECT=16:9;AUXID=3455;MODE=File;</cust>
</subtitles>
```

```
<subtitles>
  <file>STFile001</file>
  <cap>Open</cap>
```



```

    <lang>ENG</lang>
    <cust>ASPECT=HD;MODE=File + Live;AUXID=abc;</cust>
  </subtitles>

```

```

<subtitles>
  <file>STFile001</file>
  <cap>Mixed</cap>
  <lang>ENG</lang>
  <cust>ASPECT=4:3;MODE= Newsroom;</cust>
</subtitles>

```

2.6.27 <segment> Element

```

<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <segment>2</segment>

```

The <segment> element is used to indicate the presence of multi-segment media. Multi-segment media has been separated into different segments or parts. Typically, this feature might be used to divide a long program into several parts which will be played out separated by commercial breaks. Each segment is defined by its own SOM and DURATION. Segments are allowed to overlap if required.

This field may be used to simply indicate to the user that the media id is a segment or part of a longer piece of media. The automation database may have no knowledge of the actual segment SOM and DUR other than what is specified in the playlist event. In this case, the field is simply used for display purposes. In this example, the scheduling or traffic system is responsible for managing the segment data.

Alternatively, the segment field can be used by the automation in order to lookup the segment details from the automation database. The SOM and DUR in the playlist xml event will then be overwritten by the segment data in the automation database. This depends on the configuration settings of the automation playlist.

2.6.28 <timer> Element

```

<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <timer>1</timer>

```

The timer elements adds a marker to an event so that the playlist's time to 'next timer' function will display a countdown for the user. This feature may be used to allow the scheduling system to control what is displayed in the countdown timer,

2.6.29 <vps> Element

The VPS section permits home VCR's in German speaking countries to automatically record programs and account for schedule slippages.

```

<playlist>
├── <eventlist>
│   └── <event type="MEDIA">
│       └── <vps>....</vps>

```

Only ONE <vps> element is permitted per event.

The element <vps> contains child elements which are all mandatory:

Element	Description/Values
<WSS>	Aspect Ratio: 4 = 4:3 9 = 16:9 6800 = 4:3 Full 6100 = 14:9 Letterbox Centered 6200 = 14:9 Letterbox Top 6400 = 16:9 Letterbox Top 6B00 = 16:9 Letterbox Centered 6D00 = >16:9 Letterbox deeper than 16:9 6700 = 16:9 Full-height Anamorphic 6E00 = 14:9 Full-Height framed to be "14:9 safe"
<CH>	This is the channel Ident. 4 ASCII characters, padded with spaces
<Child>	Children Friendly J = Ja (Yes) N = Nein (No)
<Time>	Time HH:MM format LC:LC UC:UC SC:SC
<Date>	Date dd.mm format LC.LC UC.UC SC.SC
<Aud>	S = Stereo M = Mono 2 = 2 channel

VPS Element Example:

```
<vps>
  <WSS>6B00</WSS>
  <CH>TV1 </CH>
  <Child>J</Child>
  <Time>LC:LC</Time>
  <Date>LC.LC</Date>
  <Aud>M</Aud>
</vps>
```

2.6.30 Primary Event <customdata> Element

```
<playlist>
  └─ <eventlist>
      └─ <event type="MEDIA">
          └─ <customdata>.....</customdata>
```

The custom data element can contain data for one or more custom primary events. This section is optional.

The following items are supported within custom data:

- Abel
- Video Shuffle
- Voice Over
- Up Converter
- Aspect Ratio
- Dolby
- Slave Channels

Abel customdata Example:

```
<customdata>
  <abelcmt>Local Commercial, DLC</abelcmt>
  <abelswt>1</abelswt>
  <abelsrvcmd>0</abelsrvcmd>
  <abel>1</abel>
  <abelsrv>123456</abelsrv>
</customdata>
```

Visual Shuffle customdata Example:

```
<customdata>
  <shuffle>1</shuffle>
  <shmacro>CUSTOM</shmacro>
  <sh0>0</sh0>
  <sh1>1</sh1>
  <sh2>0</sh2>
  <sh3>1</sh3>
</customdata>
```

Voice Over customdata Example:

```
<customdata>
  <vopst>4</vopst>
  <vopstEnable>1</vopstEnable>
</customdata>
```

Upconverter customdata Example:

```
<customdata>
  <upconv>4</upconv>
  <upconvEnable>1</upconvEnable>
</customdata>
```

Aspect Ratio customdata example:

```
<customdata>
  <aspect>16:9_0</aspect>
  <aspectEnable>1</aspectEnable>
</customdata>
```

WSS:

```
<customdata>
  <aspect>4_vid</aspect>
  <aspectEnable>1</aspectEnable>
</customdata>
```

Dolby customdata example:

```
<customdata>
  <ddpst>B</ddpst>
  <ddpstEnable>1</ddpstEnable>
</customdata>
```

Slave Channels customdata example:

```
<customdata>
  <ch1_Name>TV1</ch1_Name>
  <ch1_State>0</ch1_State>

  <ch2_Name>TV2</ch2_Name>
  <ch2_State>0</ch2_State>

  <ch3_Name>TV2 Backup</ch3_Name>
  <ch3_State>0</ch3_State>

  <ch4_Name> Euro</ch4_Name>
  <ch4_State>1</ch4_State>

  <ch5_Name> Euro Backup</ch5_Name>
  <ch5_State>2</ch5_State>

  <ch6_Name>Child</ch6_Name>
  <ch6_State>1</ch6_State>
</customdata>
```

Refer to section 3.2 for more details.

2.7 <secondaryeventlist> ELEMENT

An <event> element may have 0 or 1 <secondaryeventlist> elements

```

<event>
  | .....
  |
  | <secondaryeventlist>
  |   |
  |   | <secondaryevent type="LOGO">
  |   |
  |
</event>

```

2.7.1 <secondaryevent type="....">

A primary event may contain a number of secondary events. The actual number of secondary events allowed, is dependent on the software version of Neptune being used, but is a maximum of 16.

The number of different types of secondary events is not limited by the system and new types may defined as required.

All secondary events have a defined <starttype> and/or <endtype>. The <starttype> specifies that a start action should occur for the secondary event and specifies the offset when the secondary should start relative to the primary event. The offset can be specified relative to the start of the primary event or relative to the end of the primary event and may be positive or negative. The <endtype> specifies that an end or stop action should occur for the secondary event and this has an offset like <starttype>.

A secondary event must have either a <starttype> or <endtype> specified and can have both if required. Not all secondary devices require both a start and end trigger: some simply require a start trigger or end trigger and may have no concept of a duration.

The number of types of secondary events is not limited. The type names are configurable in the system but typically include names like, "LOGO", "CG", "DVE" etc. The secondary type name may be the name of a GPI contact in the case where secondary GPIs are required.

Different devices require different parameters and so a facility is provided to allow device-specific data for a secondary event to be incorporated. This data is enclosed within the <customdata> element. As there are similarities between standard types of equipment, several schemas for the element have been defined. These are detailed below.

There may be cases where it is required to produce playlists with secondary event types where the target lists into which they are loaded have different devices although they may carry out the same function. For example one list may have a keyer type device to display a logo while another list may have a Cg or Logo Type device. In these cases it the type name is configured the same in all the lists and then the custom data for both the different device types included. When a list is loaded only the relevant custom data will be loaded by the list.

When creating playlists containing secondary events it is advisable to agree the exact configuration with Pebble Beach Systems beforehand.

2.7.1.1 <starttype origin="+Start" offset="00:00:00:00">

```

<event>
  | .....
  |
  | <secondaryeventlist>
  |   |
  |   | <secondaryevent type="LOGO">
  |       |
  |       | <starttype origin="+Start" offset="00:00:02:00"/>
  |       | <endtype origin="-End" offset="00:00:02:00"/>
  |
</event>

```

The starttype element indicates that a start action should occur for the secondary event. The action occurs at a time relative to either the start time or end time of the primary event.

It has 2 mandatory attributes:- origin and offset. The possible values for origin are:-

Origin	Description
+Start	Offset is added to the actual onair time of the primary event
-Start	Offset is subtracted from the onair time of the primary event
+End	Offset is added to the actual offair time of the primary event
-End	Offset is subtracted from the offair time of the primary event

The offset attribute is formatted as a timecode string:- "hh:mm:ss:ff".

Note:- Where a secondary offset is required to start more than a few seconds before the start time of the primary event or end more than a few seconds after the end of the primary event, (i.e. the secondary event overhangs the primary event) special consideration and handling of this case may be required depending on the device being controlled. Contact Pebble Beach for more guidance in this case.

2.7.1.2 <endtime origin="+Start" offset="00:00:00:00">

The endtype element indicates that an end action should occur for the secondary events. This may be required, for example, to stop a secondary media video clip or to clear a CG page. In some cases a stop action is not required. The action occurs relative to either the start or end time of the primary event, or it can be relative to the start time of the secondary event. It has 2 mandatory attributes:- origin and offset.

Origin	Description
+Start	Offset is added to the actual onair time of the primary event
-Start	Offset is subtracted from the onair time of the primary event
+End	Offset is added to the actual offair time of the primary event
-End	Offset is subtracted from the offair time of the primary event
+Sec Start	Offset is relative to the start time of the secondary event (ie. this is the effective duration of the secondary event)

2.7.1.3 <secondarynote>

The `<secondarynote>` element allows a textual description to be added to the secondary event that describes to the operator what the secondary event does.

```
<event>  
|  
├── .....  
└── <secondaryeventlist>  
    |  
    └── <secondaryevent type="LOGO">  
        |  
        ├── <starttype origin="+Start" offset="00:00:02:00"/>  
        ├── <endtype origin="-End" offset="00:00:02:00"/>  
        └── <secondarynote>16:9 Logo RHS</secondarynote>  
  
</event>
```

2.7.2 <secondaryevent type= "CG">

Character Generator (CG) examples:-

```
<secondaryeventlist>
  <secondaryevent type="CG">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <secondarynote>Age warning 18+</secondarynote>
    <customdata>
      <page>page200</page>           // page to be recalled
    </customdata>
  </secondaryevent>
</secondaryeventlist>

<secondaryeventlist>
  <secondaryevent type="CG">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <secondarynote>Coming Next Strap</secondarynote>
    <customdata>
      <temp>page202</temp>           // template page to be populated
      <f0>coming up next...</f0>    // up to 20 text fields
      <f1>08:00 Battlestar Gallactica</f1>
      <f2>08:30 Coronation Street</f2>
      <f3>09:00 X-Factor</f3>
    </customdata>
  </secondaryevent>
</secondaryeventlist>

<secondaryeventlist>
  <secondaryevent type="CG">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <secondarynote>16:9 Logo RHS</secondarynote>
    <customdata>
      <seq>page500</seq>           // sequence to be recalled
    </customdata>
  </secondaryevent>
</secondaryeventlist>
```

2.7.3 <secondaryevent type= "LOGO">

Logo Generator example:-

```
<secondaryeventlist>
  <secondaryevent type="LOGO">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <secondarynote>16:9 Logo RHS</secondarynote>
    <customdata>
      <page>100</page>           // page number to be recalled
      <lyr>1</lyr>              // layer to be used
      <f0>Halloween Party</f0>  // text for strap or crawl.
    </customdata>
  </secondaryevent>
</secondaryeventlist>
```


2.7.4 <secondaryevent type= "KEY">

Keyer / Voiceover example:

```
<secondaryeventlist>
  <secondaryevent type="KEY">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <secondarynote>Logo Keyer</secondarynote>
    <customdata>
      <fxno>2</fxno>           // the fx port to be used
      <tran>1</tran>           // the transition type (1 = Take) see section 3.3
      <rate>2</rate>           // the transition rate (2 = Medium) see section 3.3
    </customdata>
  </secondaryevent>
</secondaryeventlist>
```

2.7.5 <secondaryevent type= "MEDIA">

Secondary media event example:

```
<secondaryeventlist>
  <secondaryevent type="MEDIA">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <secondarynote>DVE Sec A/V source</secondarynote>
    <customdata>
      <som>10:00:00:00</som>    // the som of the clip
      <trk>3</trk>             // track number (CD and DVD only)
    </customdata>
  </secondaryevent>
</secondaryeventlist>
```

2.7.6 <secondaryevent type= "GPI1 ON">

Secondary GPI example. This turns a GPI ON after 5s from the start of the event. The GPI remains on until turned off by another event:

```
<secondaryeventlist>
  <secondaryevent type="GPI1 ON">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <secondarynote>Turn a GPI on</secondarynote>
  </secondaryevent>
</secondaryeventlist>
```

2.7.7 <secondaryevent type= "GPI1 OFF">

Secondary GPI example. This turns a GPI OFF, 5s before the end of the event. The GPI remains off until turned on by another event:

```
<secondaryeventlist>
  <secondaryevent type="GPI1 OFF">
    <starttype origin="-End" offset="00:00:05:00"/>
    <secondarynote>Turn a GPI off </secondarynote>
  </secondaryevent>
</secondaryeventlist>
```

2.7.8 <secondaryevent type= "GPI2 Pulse">

Secondary GPI example. Turns the GPI contact on and off within the same event:

```
<secondaryeventlist>
  <secondaryevent type="GPI1 ON">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <endtype origin="-End" offset="00:00:05:00"/>
    <secondarynote>Turn a GPI on </secondarynote>
  </secondaryevent>
</secondaryeventlist>
```

2.7.9 <secondaryevent type= "SecXPoint">

Secondary SecXPoint example, Switches a Router Crosspoint during a primary event:

```
<secondaryeventlist>
  <secondaryevent type="SecXPoint">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <customdata>
      <src>input8</src>      // source name in PBS config
      <dst>output10</dst>   // destination name in PBS config
      <rtr>3</rtr>           // Router name to switch to PBS config
    </customdata>
    <secondarynote>squeezeback source </secondarynote>
  </secondaryevent>
</secondaryeventlist>
```

2.7.10 Combined Custom Data for different list devices.

```
<secondaryeventlist>
  <secondaryevent type="Logo">
    <starttype origin="+Start" offset="00:00:05:00"/>
    <secondarynote>Logo from Keyer or CG </secondarynote>
    <customdata>
      <logo>3</logo>           // the logo on the cg
      <fxno>2</fxno>          // the fx port to be used
      <tran>16</tran>         // the transition type (16=FadeTake) see section 3.3
      <rate>4</rate>          // the transition rate (4=Fast) see section 3.3
    </customdata>
  </secondaryevent>
</secondaryeventlist>
```

2.8 <secondarytemplatelist> Element



NOTE: Secondary Templates can only be scheduled from the playlist XML in Neptune 3.8.1 and later versions.

An <event> element may have an optional <secondarytemplatelist> element containing one or more templates that are attached to the event. These templates can have optional start offsets from the beginning of the primary event. See <startoffset> below for further details.

```
<event>
| .....
|
| <secondarytemplatelist>
|   |
|   | <secondarytemplate name="firstTemplate">
|   |   |
|   |   | <startoffset>00:00:05:00</startoffset>
|   |   | </secondarytemplate>
|   |   |
|   |   | <secondarytemplate name="secondTemplate"/>
|   |   | ...
|   | </secondarytemplatelist>
|
| </event>
```

2.8.1 <secondarytemplate> Element

One or more <secondarytemplate> elements specify names of templates that have been defined and stored in the registry. The templates contain predefined arrangements of secondary events that can be referenced in playlists as required. An optional start offset can also be specified from the beginning of the primary event. See <startoffset> for details.

Example:

```
<secondarytemplate name="firstTemplate"/>
```

2.8.2 <startoffset> Element

The <startoffset> element is an optional parameter in the secondary template block that allows you to control when the secondary events start relative to the start of the primary event. Only those secondary events whose start times are referenced relative to the primary event's start time are affected. Any secondary events that are specified relative to the end of the primary event remain in the same place relative to that endpoint. Therefore, you can control the synchronisation of secondary events that occur, for example, at the beginning of a program's end credits, without being concerned with any events that need to stay where they are at the end of those credits.

The offset value is specified in the format hh:mm:ss:ff.

Example:

```
<secondarytemplate name="firstTemplate">
  <startoffset>00:28:00:00</startoffset>
</secondarytemplate>
```

3 ELEMENT SUMMARY

3.1 PRIMARY EVENTS

Element	Mandatory	Default (if unspecified)	Attributes	Allowable values
<playlist>	Yes			None (parent element)
<history>				None (parent element)
<created>			name, datetime	
<modified>			name, datetime	
<list>				None (parent element)
<txdate>				ISO 8601 date
<channel>				Up to 64 chars
<name>				Up to 64 chars
<eventlist>	Yes			None (parent element)
<event>	Yes		type	None (parent element)
<action>		'PLAY'		'PLAY' 'RECORD'
<starttype>		'SEQ'		'SEQ' 'FIX' 'MAN'
<onairtime>				Timecode or ISO datetime string
<onairdate>	yes ²			dd mm yy (deprecated)
<mediaid>	Yes			Up to 64 chars
<houseid>				Customer specific. Up to 32 chars
<reconcilekey>				Customer specific. Up to 32 chars
<title>	yes ¹			Up to 64 chars
<category>				Customer Site specific
<duration>	yes ¹			timecode string
<som>	yes ¹			timecode string
<stream>		0		0..9
<source>				Customer Site specific
<destpath>				Customer Site specific
<audio>				Customer Site specific
<gpimacro>				Customer Site specific

<eventnote>				Up to 256 chars
<endtype>		'NORM'		'NORM' 'UNDEF'
<effect>		Take		See section 2.6.23
<rate>		Fast		See section 2.6.24
<eg>				No longer used.
<customdata>				See section 3.2 . or PBS-1-00115 for more details.
<vps>				None (parent element)
<WSS>	Yes			See section 2.6.29
<CH>	Yes			See section 2.6.29
<Child>	Yes			See section 2.6.29
<Time>	Yes			See section 2.6.29
<Date>	Yes			See section 2.6.29
<Aud>	Yes			See section 2.6.29
<subtitles>				None (parent element)
<file>				Subtitle filename up to 364chars
<cap>				'open' 'closed' 'mixed'
<lang>				Customer site specific. Usually one or more ISO country code
<cust>				Custom subtitle settings
<secondaryeventlist>				None (parent element)
<secondarytemplatelist>				None (parent element)
<secondaryevent>				None (parent element)
<starttype> secondary event			origin, offset	None
<endtype> secondary event			origin, offset	None
<secondarynote>				up to 32chars

Note 1:- these fields are mandatory if Neptune is to play material that is not in its media database

Note 2:- mandatory only for fixed start events

3.2 PRIMARY EVENT CUSTOM DATA

Element	Mandatory	Allowable values
<customdata>		None (parent element)
Abel		
<abel>	YES (unless disabled)	0 = Disabled 1 = Enabled
<abelswt>	Required.	Transmitter Switch Mode; 0 = National Mode 1 = Local Mode
<abelsrv>	Required.	Able Server Block Id (0 = No Server Event).
<abelsrvcmd>	Always. Required if abelsrv is not 0.	The type of server event; 0 = Start / Stop 1 = Start 2 = Stop
<abelcmt>	Optional.	Comment for event.
Audio Shuffle		
<shuffle>	Always. If missing Disabled is assumed.	0 = Disabled 1 = Enabled
<shmacro>	Required.	Shuffle macro; * = A macro name that maps to a shuffle arrangement configure within the target list. CUSTOM = Stipulates that a shuffle arrangement in the form of <sh?> nodes, has been defined within this events <customdata>.
<sh?>	Only valid when the shuffle macro is set to CUSTOM.	Used to set a custom shuffle. '?' should be replaced with the 0 based output channel number. * = Input number.
Voice Over		
<vopstEnable>	Always. If missing Disabled is assumed.	0 = Disabled 1 = Enabled
<vopst>	Required.	Voice over preset.
Upconverter		
<upconvEnable>	Always. If missing Disabled is assumed.	0 = Disabled 1 = Enabled
<upconv>	Required.	Upconverter preset.

Aspect Ratio		
<aspectEnable>	Always. If missing Disabled is assumed.	0 = Disabled 1 = Enabled
<aspect>	Required.	Aspect Ratio. Can be the video index or WSS e.g."1_vid" or "4:3_0" The mode is configurable via the registry and defaults to using video index.
Dolby		
<ddpstEnable>	Always. If missing Disabled is assumed.	0 = Disabled 1 = Enabled
<ddpst>	Required.	Dolby preset.
Slave Channels		
<ch?_Name>	Required if the channel ? is to be slaved from the master.	Channel Name '?' should be replaced with the 0 based output channel number.
<ch?_State>	Required.	0 = START 1 = STOP 2 = START/STOP '?' should be replaced with the 0 based output channel number.

3.3 SECONDARY EVENTS

See PBS-1-00115 for more details and examples.

Element	Mandatory	Allowable values
<customdata>		None (parent element)
CG and Logo generators		
<page>		the page number or name
<temp>		the template number or name
<seq>		the sequence number or name
<lyr>		layer number
<f0> ..<f1>.. <fr19>		the CG template fields. Maximum 20 fields.
General Purpose Strings macros		
<str0> <str1>....<str19>		Variable number of string fields used by CG macro processor. Maximum 20 Strings
Secondary Switcher Effect (Transition Effect)		
<fxno>	Required.	The 1 based fx port number. Top range is dependant on system config.
<rate>	Optional.	0 = None 1 = Slow 2 = Medium 4 = Fast
<tran>	Optional.	0 = None 1 = Take 2 = FadeFade 4 = TakeFade 8 = Mix 16 = FadeTake 32 = Wipe NOTE: More options are available with some hardware.
<logo>	Optional [Keyer]	Numeric logo Id for MCS's that support it.
<volvl>	Optional [AudioOver]	Voice Over Level as a Percentage. 100% = Full VO Audio
<vofile>	Optional [AudioOver]	Voice Over Filename
<phase>	Optional [AudioMode]	0 = Normal, 1 = Reverse Phase
<mode>	Optional [AudioMode]	0 = Stereo, 1 = Left Mono, 2 = Right Mono
<bus>	Optional [AudioMode]	1 = PGM, 2 = PST, 3 = Both

Secondary Xpoint Event		
<rtr>	Required	The name of the XPoint device to switch.
<src>	Required	The name of the source/input port to switch.
<dst>	Required	The name of the destination/output port to switch.
<type>	Optional.	The name of the transition type.
<rate>	Optional.	The name of the transition rate.
<lyrx>	Optional.	Where x represents the layer index, specifies the numeric value of the layer to switch. Multiple layers are specified using <lyr0>, <lyr1> etc.
Strings		
<strx>	Required	Where x represents the string index, specifies the given string. Multiple strings are specified using <lyr0>, <lyr1> etc.
Voice Over		
<fxno>	Required.	Voice over preset.
<volvl>	Optional [AudioOver]	Voice over level as a Percentage. 100% = Full VO Audio
Secondary Media Event		
<type>	Always (if the tag is missing Play is assumed).	1 = Play Media, 2 = Record Media
<id>	Only one of these is valid and required.	The MediaId of item to Play/Record.
<trk>		The track number to play.
<som>	Optional.	Set the medias SOM.
Secondary GPI Event		
Note: Secondary GPI Events have no <customdata>. The secondary type name is the name of the GPI macro as defined by the configuration.		

3.4 SECONDARY TEMPLATES

Element	Mandatory	Default (if unspecified)	Allowable values
<secondarytemplatelist>	YES (if template required)		None (parent element)
<secondarytemplate>	YES (if template required)		Name of secondary template to schedule.
<startoffset>	No	00:00:00:00	Timecode string

4 EXAMPLE PLAYLISTS

4.1 EXAMPLE 1 - SIMPLE PLAYLIST

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<playlist>
  <history>
    <created name="Fred Bloggs" datetime="2009-02-03T12:03:00"/>
    <modified name="Joe Smith" datetime="2009-02-03T12:03:00"/>
  </history>
  <list>
    <name>This is the playlist name and other info</name>
    <txdate>2009-02-03</txdate>
    <channel>TV5 Slough</channel>
  </list>
  <eventlist>
    <event type="COMMENT">
      <eventnote>This is commercial break 43234</eventnote>
    </event>
    <event>
      <action>PLAY</action>
      <starttype>SEQ</starttype>
      <onairtime>12:01:30:00</onairtime>
      <mediaid>COMM1</mediaid>
      <houseid>COMM1</houseid>
      <reconcilekey>3F6B4A35</reconcilekey>
      <title>Puma ad</title>
      <category>COMMERCIAL</category>
      <duration>00:00:20:00</duration>
      <som>00:02:00:00</som>
      <destpath>TX-A</destpath>
      <gpimacro>LOGO OFF</gpimacro>
      <eventnote>Good commercial</eventnote>
    </event>
    <event>
      <action>PLAY</action>
      <starttype>SEQ</starttype>
      <onairtime>12:01:50:00</onairtime>
      <mediaid>COMM2</mediaid>
      <houseid>COMM2</houseid>
      <reconcilekey>3F6B4A36</reconcilekey>
      <title>Heineken ad</title>
      <category>COMMERCIAL</category>
      <duration>00:00:20:00</duration>
      <som>00:02:00:00</som>
      <destpath>TX-A</destpath>
      <gpimacro>LOGO OFF</gpimacro>
      <eventnote>Another commercial</eventnote>
    </event>
    <event type="COMMENT">
      <eventnote>End of commercial break 43234</eventnote>
    </event>
    <event>
      <action>PLAY</action>
      <starttype>SEQ</starttype>
      <onairtime>12:02:10:00</onairtime>
      <mediaid>PROG1</mediaid>
      <houseid>1423142343</houseid>
      <reconcilekey>3F6B4A38</reconcilekey>
      <title>Next week on Zee TV</title>
      <category>PROGRAM</category>
      <duration>00:00:30:00</duration>
      <som>00:02:00:00</som>
      <destpath>TX-A</destpath>
      <gpimacro>LOGO ON</gpimacro>
      <eventnote>Another commercial</eventnote>
    </event>
  </eventlist>
</playlist>

```

```
</event>
<event type="LIVE">
  <action>PLAY</action>
  <starttype>SEQ</starttype>
  <endtype>UNDEF</endtype>
  <onairtime>12:02:40:00</onairtime>
  <title>Live from studio</title>
  <category>STUDIO</category>
  <duration>00:30:00:00</duration>
  <destpath>TX-A</destpath>
  <gpimacro>LOGO ON</gpimacro>
  <eventnote>Need to cue studio</eventnote>
</event>
</eventlist>
</playlist>
```

4.2 EXAMPLE2: PLAYLIST WITH SECONDARY EVENTS

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<history>
    <created name="Fred Bloggs" datetime="2009-02-03T12:03:00"/>
    <modified name="Joe Bloggs" datetime="2009-02-03T12:03:00"/>
</history>
<list>
    <name>This is the playlist name and other useful information</name>
    <txdate>2009-02-03</txdate>
    <channel>TV5 Slough</channel>
</list>
<eventlist>
    <event type="MEDIA">
        <action>PLAY</action>
        <starttype>FIX</starttype>
        <onairtime>2004-02-03T12:00:00:00</onairtime>
        <mediaid>12345678</mediaid>
        <houseid>12345678-ABJ2001</houseid>
        <reconcilekey>3F6B4A33</reconcilekey>
        <title>the title</title>
        <category>PROGRAM</category>
        <duration>00:22:30:00</duration>
        <som>10:00:00:00</som>
        <destpath>TX-A</destpath>
        <gpimacro>16:9 ON</gpimacro>
        <secondaryeventlist>
            <secondaryevent type="LOGO">
                <starttype origin="+Start" offset="00:00:02:00"/>
                <endtype origin="-End" offset="00:00:02:00"/>
                <customdata>
                    <logo>100</logo>
                    <page>1</page>
                </customdata>
            </secondaryevent>
        </secondaryeventlist>
        <eventnote>This is an event note</eventnote>
    </event>
    <event type="MEDIA">
        <action>PLAY</action>
        <starttype>SEQ</starttype>
        <onairtime>12:00:30:00</onairtime>
        <mediaid>12345677</mediaid>
        <houseid>12345677-ABJ2001</houseid>
        <reconcilekey>3F6B4A34</reconcilekey>
        <title>another title</title>
        <category>TRAILER</category>
        <duration>00:01:00:00</duration>
        <som>10:00:00:00</som>
        <destpath>TX-AA</destpath>
        <gpimacro>LOGO OFF</gpimacro>
        <secondaryeventlist>
            <secondaryevent type="CG">
                <starttype origin="+Start" offset="00:00:02:00"/>
                <endtype origin="+Start" offset="00:00:12:00"/>
                <customdata>
                    <temp>2002</temp>
                    <f0>Next week</f0>
                    <f1>08:00 Coronation Street</f1>
                    <f2>08:30 Emerdate Farm</f2>
                </customdata>
            </secondaryevent>
            <secondaryevent type="CG">
                <starttype origin="-End" offset="00:01:00:00"/>
                <endtype origin="-End" offset="00:00:50:00"/>
                <customdata>
                    <temp>2002</temp>
                    <f0>coming next..</f0>
                </customdata>
            </secondaryevent>
        </secondaryeventlist>
    </event>
</eventlist>
```

```

                                <f1>08:00 Coronation Street</f1>
                                <f2>08:30 Emerdate Farm</f2>
                                </customdata>
          </secondaryevent>
          <secondaryevent type="LOGO">
            <starttype origin="+Start" offset="00:00:00:00"/>
            <customdata>
              <logo>200</logo>
              <page>1</page>
            </customdata>
          </secondaryevent>
        </secondaryeventlist>
        <eventnote>This is an event note</eventnote>
      </event>
      <event type="COMMENT">
        <eventnote>This is commercial break No 43234</eventnote>
      </event>
      <event>
        <action>PLAY</action>
        <starttype>SEQ</starttype>
        <onairtime>12:01:30:00</onairtime>
        <mediaid>COMM1</mediaid>
        <houseid>COMM1</houseid>
        <reconcilekey>3F6B4A35</reconcilekey>
        <title>Puma ad</title>
        <category>COMMERCIAL</category>
        <duration>00:00:20:00</duration>
        <som>00:02:00:00</som>
        <destpath>TX-A</destpath>
        <gpimacro>LOGO OFF</gpimacro>
        <eventnote>Good commercial</eventnote>
      </event>
      <event>
        <action>PLAY</action>
        <starttype>SEQ</starttype>
        <onairtime>12:01:50:00</onairtime>
        <mediaid>COMM2</mediaid>
        <houseid>COMM2</houseid>
        <reconcilekey>3F6B4A36</reconcilekey>
        <title>Heineken ad</title>
        <category>COMMERCIAL</category>
        <duration>00:00:20:00</duration>
        <som>00:02:00:00</som>
        <destpath>TX-A</destpath>
        <gpimacro>LOGO OFF</gpimacro>
        <eventnote>Another commercial</eventnote>
      </event>
      <event type="COMMENT">
        <eventnote>End of commercial break No 43234</eventnote>
      </event>
      <event>
        <action>PLAY</action>
        <starttype>SEQ</starttype>
        <onairtime>12:02:10:00</onairtime>
        <mediaid>PROG1</mediaid>
        <houseid>1423142343</houseid>
        <reconcilekey>3F6B4A38</reconcilekey>
        <title>Next week on Zee TV</title>
        <category>PROGRAM</category>
        <duration>00:00:30:00</duration>
        <som>00:02:00:00</som>
        <destpath>TX-A</destpath>
        <gpimacro>LOGO ON</gpimacro>
        <secondaryeventlist>
          <secondaryevent type="DVE">
            <starttype origin="-End" offset="00:01:30:00"/>
            <endtype origin="-End" offset="00:00:30:00"/>
            <secondarynote>Squeezeback</secondarynote>
            <customdata>
              <temp>2002</temp>
            </customdata>
          </secondaryevent>
        </secondaryeventlist>
      </event>
    </eventlist>
  </playlist>

```

```

                                <f0>Coming Up</f0>
                                <f1>08:00 Coronation Street</f1>
                                <f2>08:30 Emerdate Farm</f2>
                                </customdata>
          </secondaryevent>
          <secondaryevent type="MEDIA">
            <starttype origin="-End" offset="00:01:30:00"/>
            <endtype origin="-End" offset="00:00:30:00"/>
            <secondarynote>Promo for squeezeback</secondarynote>
            <customdata>
              <id>PR1234</id>
              <som>10:00:00:00</som>
            </customdata>
          </secondaryevent>
          <secondaryevent type="KEY">
            <starttype origin="-End" offset="00:01:20:00"/>
            <endtype origin="-End" offset="00:01:00:00"/>
            <secondarynote>VoiceOver</secondarynote>
            <customdata>
              <fxno>3</fxno>
              <tran>1</tran>
              <rate>4</rate>
            </customdata>
          </secondaryevent>
          <secondaryevent type="LOGO">
            <starttype origin="+End" offset="00:00:00:00"/>
            <customdata>
              <logo>200</logo>
              <page>1</page>
            </customdata>
          </secondaryevent>
        </secondaryeventlist>
        <eventnote>Another commercial</eventnote>
        <subtitles>
          <file>PROG1</file>
        </subtitles>
      </event>
      <event type="LIVE">
        <action>PLAY</action>
        <starttype>SEQ</starttype>
        <endtype>UNDEF</endtype>
        <onairtime>12:02:40:00</onairtime>
        <title>Live from studio</title>
        <category>STUDIO</category>
        <duration>00:30:00:00</duration>
        <source>IRD23</source>
        <destpath>TX-A</destpath>
        <gpimacro>LOGO ON</gpimacro>
        <eventnote>Need to cue studio</eventnote>
      </event>
    </eventlist>
  </playlist>

```


4.3 EXAMPLE 3: PLAYLIST WITH SECONDARY TEMPLATES

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<playlist>
  <history>
    <created name="Fred Bloggs" datetime="2009-02-03T12:03:00"/>
    <modified name="Joe Bloggs" datetime="2009-02-03T12:03:00"/>
  </history>
  <list>
    <name>This is the playlist name and other useful information</name>
    <txdate>2009-02-03</txdate>
    <channel>TV5 Slough</channel>
  </list>
  <eventlist>
    <event type="MEDIA">
      <action>PLAY</action>
      <starttype>FIX</starttype>
      <onairtime>2004-02-03T12:00:00</onairtime>
      <mediaid>12345678</mediaid>
      <houseid>12345678-ABJ2001</houseid>
      <reconcilekey>3F6B4A33</reconcilekey>
      <title>the title</title>
      <category>PROGRAM</category>
      <duration>00:22:30:00</duration>
      <som>10:00:00:00</som>
      <destpath>TX-A</destpath>
      <gpimacro>16:9 ON</gpimacro>
      <secondaryeventlist>
        <secondaryevent type="LOGO">
          <starttype origin="+Start" offset="00:00:02:00"/>
          <endtype origin="-End" offset="00:00:02:00"/>
          <customdata>
            <logo>100</logo>
            <page>1</page>
          </customdata>
        </secondaryevent>
      </secondaryeventlist>
      <secondarytemplatelist>
        <secondarytemplate name="firstTemplate"/>
      </secondarytemplatelist>
      <eventnote>This is an event note</eventnote>
    </event>
    <event type="MEDIA">
      <action>PLAY</action>
      <starttype>SEQ</starttype>
      <onairtime>12:00:30:00</onairtime>
      <mediaid>12345677</mediaid>
      <houseid>12345677-ABJ2001</houseid>
      <reconcilekey>3F6B4A34</reconcilekey>
      <title>another title</title>
      <category>TRAILER</category>
      <duration>00:01:00:00</duration>
      <som>10:00:00:00</som>
      <destpath>TX-AA</destpath>
      <gpimacro>LOGO OFF</gpimacro>
      <secondaryeventlist>
        <secondaryevent type="CG">
          <starttype origin="+Start" offset="00:00:02:00"/>
          <endtype origin="+Start" offset="00:00:12:00"/>
          <customdata>
            <temp>2002</temp>
            <f0>Next week</f0>
            <f1>08:00 Coronation Street</f1>
            <f2>08:30 Emerdate Farm</f2>
          </customdata>
        </secondaryevent>
      </secondaryeventlist>
    </event>
  </eventlist>
</playlist>

```

```

        </customdata>
      </secondaryevent>
      <secondaryevent type="CG">
        <starttype origin="-End" offset="00:01:00:00"/>
        <endtype origin="-End" offset="00:00:50:00"/>
        <customdata>
          <temp>2002</temp>
          <f0>coming next..

```