An Open Interface to a DTV PSIP Generator

Mark Corl, Katja Sakiewicz, Gomer Thomas
Triveni Digital, Inc.
Princeton Junction, NJ

Abstract

Digital television (DTV) standards include protocols for carrying "program and system information" in a DTV broadcast stream. On the receiver side, this data supports both channel tuning and an on-screen interactive program guide. In an ATSC environment this data is called PSIP data, and it is typically inserted into the broadcast stream by a system called a PSIP generator. The role of a PSIP generator is to gather the program- and system information from various sources, put it into the proper format, and insert it into the DTV broadcast stream at the proper times. This paper describes what information is needed and where it comes from, and it defines an XML-based open interface for communicating the information to the PSIP generator from the various sources of the information.

Program and System Information

The two primary digital television (DTV) standards in the world are the Advanced Television Systems Committee (ATSC) standard [1], developed in the United States, and the Digital Video Broadcast (DVB) standard [2], developed in Europe. Both of these use the MPEG-2 transport stream, defined in the MPEG-2 Systems standard [3], as the basis for the digital bitstream in a DTV broadcast. In both systems this transport stream may carry multiple "virtual channels". Consequently, a receiver needs "roadmap" data in the transport stream to identify the virtual channels and tune to them.

Both standards define a protocol for carrying this additional data to support channel tuning, as well as data to support an on-screen interactive program guide. This is called the PSIP standard [4] by ATSC and the SI standard [5] by DVB. In this paper we will often use the term "PSIP/SI" to refer to this data in a generic sense, regardless of environment.

The ATSC PSIP data consists of the following:

- Virtual Channel Table (VCT)
  Lists the virtual channels in the transport stream and gives information about each one, including channel name, logical channel number, and a list of its program elements (video, audio, and/or data streams making up the channel).
- Event Information Tables
  Lists the "events" (TV programs) in each virtual channel, for up to 16 days in the future, giving information such as title, start time, duration, and content advisory rating for each.
- Extended Text Tables
  Provides additional text descriptions of events and channels.
- System Time Table
  Gives the current UTC time.
- Rating Region Table
  Describes content advisory rating system in use.
- Directed Channel Change Table
  Instructs receiver to change channels at specified times, based on viewer's location, program preferences, etc.
- Directed Channel Change Selection Code Table
  Describes program preference code system.

Presented at NAB2001 Broadcast Engineering Conference, April 21-26, 2001

- Master Guide Table
  Provides a road map to the PSIP tables.

The DVB SI system is similar. The main functional differences are that it has a mechanism for associating multiple virtual channels into a "bouquet", and it does not have the directed channel change mechanism.

The carriage of PSIP/SI data is typically slightly different in cable or satellite systems than in terrestrial systems.

In a cable or satellite environment the system operator knows all the channels that the receiver can receive over the cable feed and has a vested interest in giving the viewer full information about all of them. Thus, the cable or satellite operator will typically set aside a separate communications channel that carries the PSIP/SI data for all of the TV channels in the system. This makes it convenient for a program guide application in the receiver to get at the data as needed.

In a terrestrial broadcast environment the set of stations the receiver can receive is unpredictable, stations typically have no incentive to help a viewer receive other stations, and there is no common channel that can be used exclusively for PSIP data. Therefore each station's broadcast stream typically carries PSIP/SI data for its own channels and no others. This forces the TV receiver to scan through all the physical channels it can receive to pick up this data.

The DVB SI standard explicitly supports having SI data in one transport stream that describes virtual channels in other transport streams. The ATSC PSIP standard can be interpreted as supporting this, but it is not clear that it is intended to be used that way. The Society of Cable Television Engineers (SCTE) has defined a standard specifically intended for cable environments that supports aggregating PSIP/SI data for multiple channels in a cable multiplex into a single transport stream [6].

PSIP/SI Generator

The role of a PSIP/SI generator is to collect the required information from various sources, put it into the proper format, and insert it into the transport stream at the proper intervals.

For terrestrial DTV broadcasting there is typically one PSIP/SI generator per station, inserting the PSIP/SI data into the station's one transport stream (or perhaps one per network if a network of stations are broadcasting identical content). For cable DTV broadcasting there would typically be one PSIP/SI generator per cable head end, inserting data for all the transport streams into the single channel set aside for such data.

The update requirements for the various PSIP/SI data items are widely variable.

The content advisory rating system and the directed channel change selection code system are essentially constant. Thus the PSIP generator never needs to acquire new information for the RRT and DCCSCT. It can simply send out the same information at the required intervals.

The event information in the EITs and ETTs is typically known well ahead of time, so the PSIP generator can usually acquire that information just once a day. However, there may be last minute changes in the timing, for example when a sports event runs overtime.

In an ATSC environment it is important that these last minute timing changes be reflected immediately in the PSIP data, because the content advisory blocking system depends on it. The only way for an ATSC DTV receiver to know the content advisory rating of the current program is to look in the EIT to see what event is scheduled at the current time and then look at that event's content advisory rating. If the start or finish time of the event is wrong in the EIT, the DTV receiver may use the wrong content advisory rating and fail to block properly. For the same reason, it is important that the UTC time in the STT be accurate. (The allowed tolerance in the STT time is +/- 4 seconds, which is hopefully adequate.)

Whenever there is any change in the channel lineup, say going from one high definition (HD) channel to multiple standard definition (SD) channels, or a change in the program elements, say going from a program or commercial with sound tracks in both English and Spanish to one having only an English sound track, it is important that this change be reflected in the VCT at exactly the point that the change occurs. Otherwise the receiver may fail to adjust to the change in location of the data to be fed into its audio and/or

video decoders, with resultant loss of audio and/or video.

Thus, even though the PSIP generator may know well ahead of time when such a change is going to occur, it will usually be necessary for it to get a signal at the exact time the change actually occurs, to take account of even very small timing deviations.

In theory the PSIP generator can get the information it needs about directed channel changes well ahead of time. However, in practice it may need to get a signal at the exact instant it is supposed to insert a DCCT. For example, suppose a station switches from broadcasting multiple SD channels to broadcasting a single HD channel, and it wants the viewers on all the SD channels to be switched to the HD channel at the time the change occurs. If the DCCT is inserted ahead of time, giving the time of the change, receivers may actually be off by several seconds in the time they make the switch, because of the allowed variability in the STT time. The safest thing to do is wait to insert the DCCT until the exact point at which the switch is to take place, with the time window for the change starting several seconds before this. Then all receivers will switch at the right time.

Sources of Information

A key question in designing input interfaces for a PSIP generator is where the information needed by the PSIP generator comes from.

Some of the information may be entered via a manual user interface. It is certainly possible to enter all the event information manually, although this is often not very cost effective. It is more feasible to enter last minute programming changes manually, such as time overruns or more detailed description of the contents of news programs.

A manual user interface may be local or remote. For example, defining the virtual channels may be best done locally by the engineering staff, but entering last minute details about the contents of the evening news program may be best done remotely by the news room staff, and entering last minute time changes may be best done remotely by the master control operator.

For many stations the most cost effective way to enter the program information is to import it directly in electronic form from a listing service, such as TMS (Tribune Media Services). This can typically be done once a day or so.

For some stations the program information can come from the traffic system, although this often needs to be augmented with descriptive information from elsewhere. (The traffic system is usually the definitive source for what is going to be aired when, but it usually contains only very sketchy descriptive information about the programs.)

In theory, the station's automation system is usually the best source for the split second information about changes in channel lineup, although most automation systems are not yet prepared to provide that information to a PSIP generator.

An ATSC "Implementation Subcommittee" (IS) is charged with analyzing the infrastructure needed to support DTV broadcasting. This group has identified the need for a "multiplex manager" that would manage bandwidth allocations and program element ("PID") assignments among the various virtual channels, and send the necessary information to the encoders, multiplexers, data servers, and PSIP generator. Such a multiplex manager could be a component of an automation system, or could be a stand-alone system communicating with both automation and traffic systems. It would obviously be the source for some of the key information needed by the PSIP generator.

Interface considerations

It should be clear from all this that the interface approach for a PSIP generator needs to be very flexible. It needs to be able to take different types of input from a number of different systems supplied by a number of different vendors. Some of the input must be in real time, while other input can be off-line. In many cases it needs to be able to merge input from multiple sources. Given the rapidly evolving nature of the broadcasting infrastructure, different combinations of input may be coming from different sources as time goes on.

The best way to approach this situation is to have a standard, open interface that can be used by a wide variety of sources to deliver the required

information. The remainder of this paper is devoted to describing such an interface.

Open Interface

Introduction

Triveni Digital's PSIP generator, the GuideBuilder Pro AE-10 (or GuideBuilder for short) supports a number of external interfaces to third party devices and systems that provide information intended to be encoded into the resultant PSIP output stream. Examples of these systems include traffic, automation, and listing services. These interfaces are usually proprietary to the particular vendor. The integration of these interfaces into GuideBuilder has been decided in the past based on prevalence of the interface in the market place or because the particular vendor has paid for it to be developed. The External Data Source Interface Protocol (EDSIP) described in the following is proposed as an open interface between a PSIP generator and a variety of other devices and systems, eliminating the need for custom pair wise interfaces.

Overview

The EDSIP is based on TCP/IP Sockets and XML. All information is passed in the form of requests from the third party device to the PSIP generator. In addition, the PSIP generator can respond to requests by returning information. All interactions are initiated by the third party system. The interface allows all aspects of a PSIP generator to be configured and allows a means of requesting the current configuration of the PSIP generator.

The protocol is message based. Each connection establishes a communication path between the PSIP generator and the third party system. The PSIP generator supports multiple EDSIP connections. To allow only authorized clients to access the PSIP generator, a mechanism for establishing a connection by name is provided. Authorization for named connections would then have to be defined in the setup of the PSIP generator. (At some point a more secure mechanism will be needed, perhaps using SSL.) Communication is accomplished by sending messages in either direction over the connection.

Messages are defined as complete XML pages. A single DTD describes all pages that are allowed on the link. Pages received by the PSIP generator that cannot be parsed are ignored but parsing errors are logged by the system. The PSIP generator gives feedback (also in form of a short XML page) to the client, if the sent message could successfully be executed or not.

Each XML page can be as complete or sparse as desired. The interface allows the entire system to be defined with a single message or portions of the system to be updated as necessary. There are mechanisms that add information as well as remove information to allow complete redefinition of the data set used by the PSIP generator.

Data Type Definition

The Data Type Definition (DTD) defines the syntax of EDSIP XML messages. The DTD uses a modified BNF syntax to describe the specification of the EDSIP messages. Refer to Annex A for a short introduction to XML, or see [7] for a complete specification of XML and DTD syntax.

In general, the values are all strings encoded based on the specified encoding in the header (presently, the encodings UTF-8 and UTF-16 are supported). Each message is a tree structure with more complexity added as branches are added. Each node of the tree (defined as an ELEMENT) may contain sub nodes, and a set of attributes described in an attribute list (ATTLIST). Values in the attribute list describe the basic settings of the system.

Text values are used as is. Number fields are converted from the textual representation. Numbers may be represented in either decimal or hexadecimal form. Hexadecimal numbers are specified by the prefix "0x" before the number and may contain only digits "0" through "9" and the letters "A" through "F".

Please refer to Annex B for an excerpt from the DTD that defines the syntax of an EDSIP message.

General Structure of Messages

Each message sent over the EDSIP is defined as an EDSIP_Message. The EDSIP_Message contains various attributes, but must have at least the Command attribute that defines how the PSIP generator should handle the remainder of the message.

In addition to the attributes, the EDSIP_Message can also contain zero or more Channel elements.

Channels are equivalent semantically to a single delivery frequency. Each channel represents a single transport stream.

Within each Channel element can be multiple Service elements. Services are equivalent to MPEG2 Programs and ATSC Virtual Channels. A Service defines how to extract a set of audio, video and data streams from a Channel multiplex.

Each Service element contains a collection of Program elements. Programs are thought of in the broadcaster's sense and represent a television program. Programs are encoded as ATSC Events.

Figure 1. provides an example of the EDSIP_Message structure. Note that this is only an example, the number of Channels, Services or Programs that can be defined is not restricted. The protocol allows any of these elements to be added, modified, deleted or queried by setting the Command field in the message attributes. These elements are described in detail below, in the next section.

All of the fields in the attribute lists are marked as "IMPLIED". This means that the fields may be included or not. A not specified attribute for a field in an already existing, corresponding data structure in the PSIP generator simply means that the old value of the data structure will remain the same. But also, this way the protocol allows addressing a group of elements at once. For example, if the third party system would like to define a specific AC-3 encoding setting for all programs in a given service, the message would describe the specific channel and service along with a non specified program containing the AC-3 element. Moreover, empty attributes within a new element that does not yet have a corresponding data structure in the PSIP generator implies that default values are to be used for the fields in the data structure.

EDSIP Element Descriptions

EDSIP_Message Element

Optionally, the EDSIP_Message can contain a Cable attribute that specifies whether the PSIP generator is running in a cable or terrestrial environment. Also, the EDSIP_Message can contain a ClientID, which specifies the sender of the message. This attribute is used for named connections.

The Command attribute is a text field that contains a single command word. The following are valid commands:

update

The update command is typically the most used command since it allows third party systems to send data to the PSIP generator. Information in the

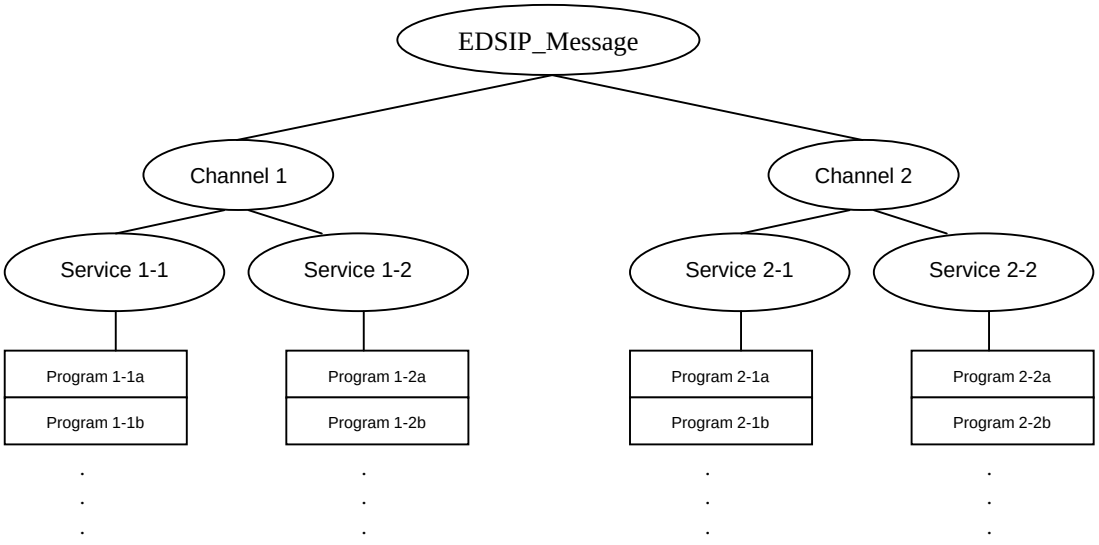

Figure 1. Logical Structure of Messages

remaining fields of the message is added to the PSIP generator's data set. In most cases, the third party system will supply updates for program information. Only the information necessary to complete the request is necessary. For example, if the third party system would like to modify Program 2-1c it can send a message with the update command and include the Channel 2 definition, the Service 2-1 definition and the definition of the particular program – in this case, Program 2-1c.

delete

The delete command allows any portion of the data set to be deleted. The third party system should use this command with care since it is easy to delete entire groups of services and programs by simply specifying an empty channel for deletion. (Typically each remote system would have a "role" assigned to it that would limit the scope of its operations.)

query

The query command requests data from the PSIP generator. The third party system provides a skeleton data structure as part of the query that the PSIP generator will fill out and return. Lack of particular attributes, for example the name in a particular element, is interpreted as a wild card causing the PSIP generator to return all Elements matching the specific attributes defined. For example, if all services of a particular channel are desired, the third party system will supply the exact description of the channel (the name) and a service entry with no attributes. The PSIP generator will return all of services (without Programs) defined for the channel.

ack

The ack command is sent by the PSIP generator back to the client in order to acknowledge that the last message from the client was received and processed. If the last message sent to the PSIP generator was a query command, the acknowledgement message will also contain the requested information. In all other cases, the returned ack message contains only the command attribute.

nak

The nak command is sent by the PSIP generator back to the client in order to report that the last message was received but could not be processed. The nak message will only contain the command attribute.

Furthermore, the EDSIP_Message can contain any number of Channel elements (possibly none).

Channel Element

The Channel Element has these attributes that are used to define a particular channel: a name, short name, transport stream id, modulation mode, major channel number and carrier frequency. Also, the Channel element can contain any number of Service elements.

Service Element

The Service element has these attributes that are used to define a particular virtual channel: a short name, extended name, minor channel number, program number, service type, source id, a flag indicating if this service is hidden, and a flag indicating if this service is access controlled.

The Service element also has various sub elements, the most important being Programs and ServiceLocations, which describe the MPEG2 program elements within this virtual channel.

Program Element

Program elements have these attributes: start time and duration. Sub elements are Title, Description, ContentAdvisory, AC3Definition, CaptionService.

Please refer to Annex C for an example message a client could send to the PSIP generator.

The EDSIP in Practice

There are many use cases for the EDSIP. For example, the news room could add more details to a news program by sending an EDSIP_Message to the PSIP generator with an update command. That message would contain a Channel element possibly with a major channel number attribute. The Channel element would contain a Service element possibly with a minor channel number attribute. The Service element would contain a Program element with an identifier of the Program to be changed, and a new Description element.

A similar example would be the master control room indicating changes to the time information of a specific program event, for example when a program runs over its previously assigned time. The master control can simply send a message to update the times of the affected Programs.

An automation system can notify the PSIP generator of PID changes by sending an EDSIP update message to update a ServiceLocation element of the corresponding Service.

Summary

It is important for DTV broadcasters to put complete and correct PSIP/SI data in their broadcast stream. It allows DTV receivers to tune to their channel(s) correctly, and it allows on-screen interactive EPGs to display information about their programs, thereby attracting viewers.

Many DTV encoders and multiplexers put skeletal PSIP/SI data in the broadcast stream, but it typically requires a separate PSIP/SI generator to do the job well. The generator collects the needed information from multiple sources and inserts it into the broadcast stream with the correct format and timing.

Sources for PSIP/SI data include local and remote user interfaces, traffic and automation systems, data broadcast servers, and program listing services. Rather than have many pair wise interfaces between different PSIP/SI generators and different sources of PSIP/SI data, it is far better to have a single open interface specification.

This paper proposes an XML-based interface for this purpose. This interface supports adding, deleting, updating, and also retrieving all or any portion of the PSIP/SI data, with great flexibility. The use of XML makes it readily extensible to meet future needs.

References

[1] *ATSC Digital Television Standard with Amendment No. 1*, ATSC Doc. A/53, March 2000.

[2] *Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for digital terrestrial broadcasts*, ETSI Doc. En 300 744, v1.1.2, August 1997.

[3] *Information technology – Generic coding of moving pictures and associated audio information: Systems*, ISO/IEC International Standard 13818-1, First edition, April 1996.

[4] *Program and System Information Protocol for Terrestrial Broadcast and Cable* (Revision A) with Amendment No. 1, ATSC Doc A/65A, May 2000.

[5] *Digital Video Broadcasting (DVB): Specification for Service Information (SI) in DVB Systems,* ETSI Doc. EN 300 468, v1.3.1, February 1998.

[6] *Service Information Delivered Out-of-Band for Digital Cable Television,* SCTE Doc. DVS 234, March 2000.

[7] *Extensible Markup Language (XML) 1.0* (Second Edition), W3C Recommendation 6, October 2000 (http://www.w3.org/TR/REC-xml).

Annex A. Introduction to XML

XML is a text-based markup language that is widely being used for data interchange between applications. The World Wide Web Consortium (W3C) is responsible for standardizing XML. As with HTML, data is identified using tags (identifiers enclosed in angle brackets, like this: <...>). Collectively, the tags are known as "markup".

But unlike HTML, XML tags tell the meaning of data, rather than how to display it. For instance, while an HTML tag says something like "display this data in bold font" (`<b>...</b>`), an XML tag puts a label on a piece of data that identifies it (for example: `<message>...</message>`).

Tags can also have attributes that give additional information about the tag. Since tags can be nested arbitrarily, XML is ideally suited to represent data hierarchically.

An example XML structure would be:

```
<person ssn=”123-45-6789” >
    <name>
        <first>John </first>
        <last> Smith</last>
    </name>
</person>
```

Here there are four tags (“person”, “name”, “first”, “last”). “ssn” is an attribute of the tag “person”. The tags “first” and “last” are nested within the “name” tag.

Although one is free to invent tags freely for structuring data, when different applications plan to interface using XML, it becomes convenient to use a Document Type Declaration (DTD), which

specifies the kind of tags and attributes, and their arrangement

Annex B: Abbreviated DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Other encodings are allowed -->
<!ELEMENT EDSIP_Message (Channel* , RatingRegion*)>
    <!ATTLIST EDSIP_Message
        Command CDATA #REQUIRED
        ClientID CDATA #IMPLIED
        Cable (0|1) #IMPLIED
    >
    <!--END ATTLIST EDSIP_Message -->

    <!ELEMENT Channel (Service*)>
        <!ATTLIST Channel
            Name CDATA #IMPLIED
            ShortName CDATA #IMPLIED
            TransportStreamID CDATA #IMPLIED
            ModulationMode (Analog|SCTE_Mode_1|SCTE_Mode_2|
                            ATSC_8VSB|ATSC_16VSB) 'ATSC_8VSB'
            MajorChannelNumber CDATA #IMPLIED
            CarrierFrequency CDATA #IMPLIED
        >

        <!ELEMENT Service (ServiceDescription* , Program* ,
                           TimeShiftedService* ,
                           ServiceLocation*, AC3Definition?,
                           CaptionService*)>
            <!ATTLIST Service
                ShortName CDATA #IMPLIED
                ExtendedName CDATA #IMPLIED
                MinorChannelNumber CDATA #IMPLIED
                ProgramNumber CDATA #IMPLIED
                ServiceType (ATSC_Digital_Television|Analog_Television|
                             ATSC_Audio_Only|ATSC_Data_Broadcast_Service)
                             'ATSC_Digital_Television'
                SourceID CDATA #IMPLIED
                IsAccessControlledFlag (1|0) #IMPLIED
                IsHiddenFlag (1|0) #IMPLIED>


            <!ELEMENT ServiceDescription (#PCDATA)*>
            <!--Enter the description itself within the tags of this element-->
                <!ATTLIST ServiceDescription
                    Language (Chinese|Danish|Dutch|English|Finnish|
                              French|German|Italian|Japanese|Korean|
                              Konkani|Spanish|Swedish) 'English'
                    Compression (0|1|2) #IMPLIED
                >
                <!--END ATTLIST ServiceDescription-->
            <!--END ELEMENT ServiceDescription -->

            <!ELEMENT Program (Title* , ProgramDescription* ,
```

```
                            ContentAdvisoryElement*,
                            AC3Definition? , CaptionService*,
                            ServiceLocation*)>
            <!-- Zero or more programs per service. -->
                <!ATTLIST Program
                    ProgramID CDATA #IMPLIED
                    StartTime CDATA #IMPLIED
                    Duration CDATA #IMPLIED
                >
                <!--END ATTLIST Program -->

                <!ELEMENT Title (#PCDATA)*>
                <!--Enter the title itself within the tags of this element-->
                    <!ATTLIST Title
                        Language (Chinese|Danish|Dutch|English|
                                  Finnish|French|German|Italian|
                                  Japanese|Korean|Konkani|Spanish|
                                  Swedish) 'English'
                        Compression (0|1|2) #IMPLIED
                    >
                    <!--END ATTLIST Title-->
                <!--END ELEMENT Title -->
              <!--END ELEMENT ProgramDescription -->
            <!--END ELEMENT Program -->
        <!--END ELEMENT Service-->
    <!--END ELEMENT Channel-->
<!--END ELEMENT EDSIP_Message -->
```

Annex C: Sample message

EDSIP message to add (or update) two virtual channels within one Channel:

```
<?xml version="1.0" encoding="UTF-16"?>
<!DOCTYPE EDSIP_Message SYSTEM "file:./EDSIPMessage.dtd">
<EDSIP_Message
 Command="update"
 Cable="0"
>
    <Channel
     Name="Triveni Digital TestChannel"
     ShortName="TD - TC"
     TransportStreamID="5"
     MajorChannelNumber="6"
     CarrierFrequency="542310000"
    >
        <Service
         ShortName="TD-1"
         ExtendedName="First Triveni Digital Virtual Test Channel"
         MinorChannelNumber="1"
         ProgramNumber="1"
         SourceID="23"
         IsAccessControlledFlag="0"
         IsHiddenFlag="0"
        >
            <ServiceLocation
             StreamType="ATSC_Video"
```

```
             PCR="1"
             PID="17"
             Language="English"
            >
            </ServiceLocation>
            <ServiceLocation
             StreamType="ATSC_AC-3_Audio"
             PCR="0"
             PID="20"
             Language="English"
            >
            </ServiceLocation>
        </Service>
        <Service
         ShortName="TD-2"
         ExtendedName="Second Triveni Digital Virtual Test Channel"
         MinorChannelNumber="2"
         ProgramNumber="2"
         SourceID="31"
         IsAccessControlledFlag="0"
         IsHiddenFlag="0"
        >
            <ServiceLocation
             StreamType="ATSC_Video"
             PCR="1"
             PID="33"
             Language="English"
            >
            </ServiceLocation>
            <ServiceLocation
             StreamType="ATSC_AC-3_Audio"
             PCR="0"
             PID="36"
             Language="English"
            >
            </ServiceLocation>
        </Service>
    </Channel>
</EDSIP_Message>
```